Computers, Cut-ups and Combinatory Volvelles:

An Archaeology of Text-Generating Mechanisms

By

Whitney Anne Trettien

B.A. English, Philosophy
Hood College, 2007

SUBMITTED TO THE PROGRAM IN COMPARATIVE MEDIA STUDIES IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE IN COMPARATIVE MEDIA STUDIES
AT THE
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

JUNE 2009

Signature of Author:_____
Program in Comparative Media Studies
May 6, 2009

Certified by:_____
William Charles Uricchio
Professor of Comparative Media Studies
Co-Director, Comparative Media Studies
Thesis Supervisor

Accepted by:_____
Henry Jenkins III
Peter de Florez Professor of Humanities
Professor of Comparative Media Studies and Literature
Co-Director, Comparative Media Studies

Computers, Cut-ups and Combinatory Volvelles:

An Archaeology of Text-Generating Mechanisms

by

Whitney Anne Trettien

Submitted to the Program in Comparative Media Studies
on May 6, 2009, in Partial Fulfillment of the
Requirements for the Degree of Master of Science in
Comparative Media Studies

ABSTRACT

Through an archaeology of text-generating mechanisms, the present work excavates the deep history of reading and writing as material, combinatory practices. On the one hand, by positing the physical manipulation of language as a form of reading and writing, this archaeology answers Roger Chartier's call for book historians to "take on the task of retracing forgotten gestures and habits" that do not fit "the genealogy of our own contemporary manner of reading," a call echoed in much recent work on the "use" of early modern books. It thus challenges our assumptions about how readers and writers of the past made meaning from printed texts and, more broadly, the expressive potentials of the printed book itself. Yet this archaeology of *ars combinatoria*, the art of combination, also presents a imaginative challenge to historians of the book. For if we accept physically cutting paper or spinning a volvelle as a readerly *and* writerly act, then we must also erase the boundaries we have drawn between "the book" as a material form and "the digital" as an epistemology, reconsidering the various literacies each facilitates or forecloses.

In keeping with the spirit of media archaeology, which seeks to defamiliarize the past, the present work *on* text-generating mechanisms exists *as* a web-based text-generating mechanism. On the one hand, this medium allows me to present a comparative history without compromising specificity or reducing the complexity of one moment to a mere reflection of another; yet it still strives for thematic cohesion by using our digital present quite literally as a map for exploring programmatic epistemologies in our past. It lives on the web at:
http://www.whitneyannetrettien.com/thesis/

Since MIT Libraries requires a paper copy of a thesis, all HTML pages and code used to produce this thesis are copied in the space below.

Thesis Supervisor: William Charles Uricchio
Title: Professor of Comparative Media Studies

ACKNOWLEDGMENTS

I would like to express my deep gratitude toward my committee members William Uricchio, Kurt Fendt and Ed Barrett for their support and guidance throughout the process of writing (and reading) this work. I would also like to thank Nick Montfort and Chris Funkhouser for their invaluable input on early drafts of my project.

I am also grateful to Madeleine Clare Elish, Lana Swartz and Jason Rockwood for reading and commenting on a draft of my introduction, and to study-buddies Xiaochang Li and Lauren Silberman for keeping me at the library when I'd rather be out riding my bike. Thanks also to the CMS community at large for not only supporting my crazy ideas, but encouraging them.

Finally, "thanks" is too feeble a word for my gratitude to Phillip Torres, without whom none of this would have been possible.

TABLE OF CONTENTS

```
<!--
___index.html___
by Whitney Anne Trettien

This is the main page.
-->

<!doctype html><head><meta http-equiv="Content-Type" content="text/html;
charset=utf-8" />
<title>COMPUTERS, CUT-UPS AND COMBINATORY VOLVELLES, by whitney anne
trettien</title>
<link rel="stylesheet" type="text/css" href="style.css" />
<script src="scripts/jquery-1.3.1.min.js" type="text/javascript"></script>
<script src="scripts/effects.core.js" type="text/javascript"></script>
<script src="scripts/effects.slide.js" type="text/javascript"></script>
<script src="scripts/thesis.js" type="text/javascript"></script>
<script src="scripts/thickbox-compressed.js" type="text/javascript"></script>
<link rel="stylesheet" href="scripts/thickbox.css" type="text/css"
media="screen" />
</head>
<body onLoad="loadContentTopleftvert(1);">
<div id="titlebox">
<p class="titlebox">
<span style="color:#9FEE00;font-size:1.65em"><b>computers,<br/></span>
<span style="color:#9FEE00;font-size:2.3em;">cut-ups,<br/></span>
<span style="color:#9FEE00;font-size:1.3em;">&combinatory<br/></span>
<span style="color:#9FEE00;font-size:1.8em;">volvelles:<br/></span>
<span style="font-size:1.7em">an archae-</span><BR/>
<span style="font-size:2.2em;">ology of</span><BR/>
<span style="font-size:3.7em;">text-</span><BR/>
<span style="font-size:1.7em;">generating</span><BR/>
<span style="font-size:1.7em;">mechanisms</span><BR/>
<span style="color:#E6399B;font-size:1.15em;">------------------------------
<br/></span>
<span style="color:#888888;font-size:.9em;">whitney trettien <span
style="color:#E6399B">|</span> <a href="#thesis">you</a><br/></span>
<span style="color:#E6399B;font-size:1.15em;">------------------------------
<br/></span>
<span style="font-size:1.2em"><a
href="intro1.html?KeepThis=true&TB_iframe=true&height=480&width=450"
class="thickbox"
title="I&middot;N&middot;T&middot;R&middot;O&middot;D&middot;U&middot;C&middo
t;T&middot;I&middot;O&middot;N">about this work</span></a></b><br/></span>
</p>
</div>
<div id="cubebox">
<span style="cursor: pointer" onclick="hideAll(14833);"><div class="cube14833
permutation"></div></span>
<span style="cursor: pointer" onclick="hideAll(161);"><div class="cube161
permutation"></div></span>
<span style="cursor: pointer" onclick="hideAll(162);"><div class="cube162
permutation"></div></span>
<span style="cursor: pointer" onclick="hideAll(163);"><div class="cube163
permutation"></div></span>
<span style="cursor: pointer" onclick="hideAll(164);"><div class="cube164
permutation"></div></span>
<span style="cursor: pointer" onclick="hideAll(100);"><div class="cube100
cutups"></div></span>
```

```
<span style="cursor: pointer" onclick="hideAll(101);"><div class="cube101
cutups"></div></span>
<span style="cursor: pointer" onclick="hideAll(102);"><div class="cube102
cutups"></div></span>
<span style="cursor: pointer" onclick="hideAll(60);"><div class="cube60
thirtyyearswar"></div></span>
<span style="cursor: pointer" onclick="hideAll(61);"><div class="cube61
thirtyyearswar"></div></span>
<span style="cursor: pointer" onclick="hideAll(62);"><div class="cube62
thirtyyearswar"></div></span>
<span style="cursor: pointer" onclick="hideAll(63);"><div class="cube63
thirtyyearswar"></div></span>
<span style="cursor: pointer" onclick="hideAll(200);"><div class="cube200
leibniz"></div></span>
<span style="cursor: pointer" onclick="hideAll(201);"><div class="cube201
leibniz"></div></span>
<span style="cursor: pointer" onclick="hideAll(254);"><div class="cube254
permutation"></div></span>
<span style="cursor: pointer" onclick="hideAll(253);"><div class="cube253
permutation"></div></span>
<span style="cursor: pointer" onclick="hideAll(252);"><div class="cube252
permutation"></div></span>
<span style="cursor: pointer" onclick="hideAll(251);"><div class="cube251
permutation"></div></span>
<span style="cursor: pointer" onclick="hideAll(250);"><div class="cube250
permutation"></div></span>
<span style="cursor: pointer" onclick="hideAll(105);"><div class="cube105
cutups"></div></span>
<span style="cursor: pointer" onclick="hideAll(104);"><div class="cube104
cutups"></div></span>
<span style="cursor: pointer" onclick="hideAll(103);"><div class="cube103
cutups"></div></span>
<span style="cursor: pointer" onclick="hideAll(67);"><div class="cube67
thirtyyearswar"></div></span>
<span style="cursor: pointer" onclick="hideAll(66);"><div class="cube66
thirtyyearswar"></div></span>
<span style="cursor: pointer" onclick="hideAll(65);"><div class="cube65
thirtyyearswar"></div></span>
<span style="cursor: pointer" onclick="hideAll(64);"><div class="cube64
thirtyyearswar"></div></span>
<span style="cursor: pointer" onclick="hideAll(203);"><div class="cube203
leibniz"></div></span>
<span style="cursor: pointer" onclick="hideAll(202);"><div class="cube202
leibniz"></div></span>
<span style="cursor: pointer" onclick="hideAll(255);"><div class="cube255
permutation"></div></span>
<span style="cursor: pointer" onclick="hideAll(256);"><div class="cube256
permutation"></div></span>
<span style="cursor: pointer" onclick="hideAll(257);"><div class="cube257
permutation"></div></span>
<span style="cursor: pointer" onclick="hideAll(258);"><div class="cube258
permutation"></div></span>
<span style="cursor: pointer" onclick="hideAll(259);"><div class="cube259
permutation"></div></span>
<span style="cursor: pointer" onclick="hideAll(106);"><div class="cube106
cutups"></div></span>
<span style="cursor: pointer" onclick="hideAll(107);"><div class="cube107
cutups"></div></span>
```

```
<span style="cursor: pointer" onclick="hideAll(108);"><div class="cube108
cutups"></div></span>
<span style="cursor: pointer" onclick="hideAll(109);"><div class="cube109
cutups"></div></span>
<span style="cursor: pointer" onclick="hideAll(110);"><div class="cube110
cutups"></div></span>
<span style="cursor: pointer" onclick="hideAll(68);"><div class="cube68
thirtyyearswar"></div></span>
<span style="cursor: pointer" onclick="hideAll(1003);"><div class="cube1003
thirtyyearswar"></div></span>
<span style="cursor: pointer" onclick="hideAll(204);"><div class="cube204
leibniz"></div></span>
<span style="cursor: pointer" onclick="hideAll(205);"><div class="cube205
leibniz"></div></span>
<span style="cursor: pointer" onclick="hideAll(263);"><div class="cube263
permutation"></div></span>
<span style="cursor: pointer" onclick="hideAll(262);"><div class="cube262
permutation"></div></span>
<span style="cursor: pointer" onclick="hideAll(261);"><div class="cube261
permutation"></div></span>
<span style="cursor: pointer" onclick="hideAll(260);"><div class="cube260
permutation"></div></span>
<span style="cursor: pointer" onclick="hideAll(116);"><div class="cube116
cutups"></div></span>
<span style="cursor: pointer" onclick="hideAll(115);"><div class="cube115
cutups"></div></span>
<span style="cursor: pointer" onclick="hideAll(114);"><div class="cube114
cutups"></div></span>
<span style="cursor: pointer" onclick="hideAll(113);"><div class="cube113
cutups"></div></span>
<span style="cursor: pointer" onclick="hideAll(112);"><div class="cube112
cutups"></div></span>
<span style="cursor: pointer" onclick="hideAll(111);"><div class="cube111
cutups"></div></span>
<span style="cursor: pointer" onclick="hideAll(1005);"><div class="cube1005
thirtyyearswar"></div></span>
<span style="cursor: pointer" onclick="hideAll(1004);"><div class="cube1004
thirtyyearswar"></div></span>
<span style="cursor: pointer" onclick="hideAll(207);"><div class="cube207
leibniz"></div></span>
<span style="cursor: pointer" onclick="hideAll(206);"><div class="cube206
leibniz"></div></span>
<span style="cursor: pointer" onclick="hideAll(264);"><div class="cube264
permutation"></div></span>
<span style="cursor: pointer" onclick="hideAll(265);"><div class="cube265
permutation"></div></span>
<span style="cursor: pointer" onclick="hideAll(266);"><div class="cube266
permutation"></div></span>
<span style="cursor: pointer" onclick="hideAll(267);"><div class="cube267
permutation"></div></span>
<span style="cursor: pointer" onclick="hideAll(117);"><div class="cube117
cutups"></div></span>
<span style="cursor: pointer" onclick="hideAll(118);"><div class="cube118
cutups"></div></span>
<span style="cursor: pointer" onclick="hideAll(119);"><div class="cube119
cutups"></div></span>
<span style="cursor: pointer" onclick="hideAll(11911);"><div class="cube11911
cutups"></div></span>
```

```
<span style="cursor: pointer" onclick="hideAll(120);"><div class="cube120
cutups"></div></span>
<span style="cursor: pointer" onclick="hideAll(121);"><div class="cube121
cutups"></div></span>
<span style="cursor: pointer" onclick="hideAll(1006);"><div class="cube1006
thirtyyearswar"></div></span>
<span style="cursor: pointer" onclick="hideAll(1016);"><div class="cube1016
thirtyyearswar"></div></span>
<span style="cursor: pointer" onclick="hideAll(208);"><div class="cube208
leibniz"></div></span>
<span style="cursor: pointer" onclick="hideAll(209);"><div class="cube209
leibniz"></div></span>
<span style="cursor: pointer" onclick="hideAll(1020);"><div class="cube1020
permutation"></div></span>
<span style="cursor: pointer" onclick="hideAll(1034);"><div class="cube1034
permutation"></div></span>
<span style="cursor: pointer" onclick="hideAll(128);"><div class="cube128
cutups"></div></span>
<span style="cursor: pointer" onclick="hideAll(127);"><div class="cube127
cutups"></div></span>
<span style="cursor: pointer" onclick="hideAll(149);"><div class="cube149
cutups"></div></span>
<span style="cursor: pointer" onclick="hideAll(126);"><div class="cube126
cutups"></div></span>
<span style="cursor: pointer" onclick="hideAll(1011);"><div class="cube1011
cutups"></div></span>
<span style="cursor: pointer" onclick="hideAll(124);"><div class="cube124
cutups"></div></span>
<span style="cursor: pointer" onclick="hideAll(123);"><div class="cube123
cutups"></div></span>
<span style="cursor: pointer" onclick="hideAll(122);"><div class="cube122
cutups"></div></span>
<span style="cursor: pointer" onclick="hideAll(213);"><div class="cube213
leibniz"></div></span>
<span style="cursor: pointer" onclick="hideAll(212);"><div class="cube212
leibniz"></div></span>
<span style="cursor: pointer" onclick="hideAll(211);"><div class="cube211
leibniz"></div></span>
<span style="cursor: pointer" onclick="hideAll(210);"><div class="cube210
leibniz"></div></span>
<span style="cursor: pointer" onclick="hideAll(129);"><div class="cube129
computers"></div></span>
<span style="cursor: pointer" onclick="hideAll(143);"><div class="cube143
computers"></div></span>
<span style="cursor: pointer" onclick="hideAll(12911);"><div class="cube12911
cutups"></div></span>
<span style="cursor: pointer" onclick="hideAll(130);"><div class="cube130
cutups"></div></span>
<span style="cursor: pointer" onclick="hideAll(131);"><div class="cube131
cutups"></div></span>
<span style="cursor: pointer" onclick="hideAll(13111);"><div class="cube13111
cutups"></div></span>
<span style="cursor: pointer" onclick="hideAll(13122);"><div class="cube13122
cutups"></div></span>
<span style="cursor: pointer" onclick="hideAll(132);"><div class="cube132
cutups"></div></span>
<span style="cursor: pointer" onclick="hideAll(133);"><div class="cube133
cutups"></div></span>
```

```
<span style="cursor: pointer" onclick="hideAll(134);"><div class="cube134
cutups"></div></span>
<span style="cursor: pointer" onclick="hideAll(214);"><div class="cube214
leibniz"></div></span>
<span style="cursor: pointer" onclick="hideAll(215);"><div class="cube215
leibniz"></div></span>
<span style="cursor: pointer" onclick="hideAll(216);"><div class="cube216
leibniz"></div></span>
<span style="cursor: pointer" onclick="hideAll(219);"><div class="cube219
leibniz"></div></span>
<span style="cursor: pointer" onclick="hideAll(146);"><div class="cube146
computers"></div></span>
<span style="cursor: pointer" onclick="hideAll(144);"><div class="cube144
computers"></div></span>
<span style="cursor: pointer" onclick="hideAll(141);"><div class="cube141
cutups"></div></span>
<span style="cursor: pointer" onclick="hideAll(140);"><div class="cube140
cutups"></div></span>
<span style="cursor: pointer" onclick="hideAll(139);"><div class="cube139
cutups"></div></span>
<span style="cursor: pointer" onclick="hideAll(138);"><div class="cube138
cutups"></div></span>
<span style="cursor: pointer" onclick="hideAll(137);"><div class="cube137
cutups"></div></span>
<span style="cursor: pointer" onclick="hideAll(136);"><div class="cube136
cutups"></div></span>
<span style="cursor: pointer" onclick="hideAll(135);"><div class="cube135
cutups"></div></span>
<span style="cursor: pointer" onclick="hideAll(224);"><div class="cube224
leibniz"></div></span>
<span style="cursor: pointer" onclick="hideAll(223);"><div class="cube223
leibniz"></div></span>
<span style="cursor: pointer" onclick="hideAll(222);"><div class="cube222
leibniz"></div></span>
<span style="cursor: pointer" onclick="hideAll(221);"><div class="cube221
leibniz"></div></span>
<span style="cursor: pointer" onclick="hideAll(220);"><div class="cube220
leibniz"></div></span>
<span style="cursor: pointer" onclick="hideAll(150);"><div class="cube150
computers"></div></span>
<span style="cursor: pointer" onclick="hideAll(151);"><div class="cube151
computers"></div></span>
<span style="cursor: pointer" onclick="hideAll(142);"><div class="cube142
cutups"></div></span>
<span style="cursor: pointer" onclick="hideAll(147);"><div class="cube147
cutups"></div></span>
<span style="cursor: pointer" onclick="hideAll(14711);"><div class="cube14711
cutups"></div></span>
<span style="cursor: pointer" onclick="hideAll(14722);"><div class="cube14722
cutups"></div></span>
<span style="cursor: pointer" onclick="hideAll(148);"><div class="cube148
cutups"></div></span>
<span style="cursor: pointer" onclick="hideAll(14811);"><div class="cube14811
cutups"></div></span>
<span style="cursor: pointer" onclick="hideAll(14822);"><div class="cube14822
cutups"></div></span>
<span style="cursor: pointer" onclick="hideAll(225);"><div class="cube225
leibniz"></div></span>
```

```
<span style="cursor: pointer" onclick="hideAll(226);"><div class="cube226
leibniz"></div></span>
<span style="cursor: pointer" onclick="hideAll(227);"><div class="cube227
leibniz"></div></span>
<span style="cursor: pointer" onclick="hideAll(228);"><div class="cube228
leibniz"></div></span>
<span style="cursor: pointer" onclick="hideAll(229);"><div class="cube229
leibniz"></div></span>
<span style="cursor: pointer" onclick="hideAll(152);"><div class="cube152
computers"></div></span>
<span style="cursor: pointer" onclick="hideAll(153);"><div class="cube153
computers"></div></span>
<span style="cursor: pointer" onclick="hideAll(154);"><div class="cube154
computers"></div></span>
<span style="cursor: pointer" onclick="hideAll(155);"><div class="cube155
computers"></div></span>
<span style="cursor: pointer" onclick="hideAll(156);"><div class="cube156
computers"></div></span>
<span style="cursor: pointer" onclick="hideAll(43);"><div class="cube43
towerofbabel"></div></span>
<span style="cursor: pointer" onclick="hideAll(44);"><div class="cube44
towerofbabel"></div></span>
<span style="cursor: pointer" onclick="hideAll(14);"><div class="cube14
stammworter"></div></span>
<span style="cursor: pointer" onclick="hideAll(15);"><div class="cube15
stammworter"></div></span>
<span style="cursor: pointer" onclick="hideAll(16);"><div class="cube16
stammworter"></div></span>
<span style="cursor: pointer" onclick="hideAll(233);"><div class="cube233
leibniz"></div></span>
<span style="cursor: pointer" onclick="hideAll(232);"><div class="cube232
leibniz"></div></span>
<span style="cursor: pointer" onclick="hideAll(231);"><div class="cube231
leibniz"></div></span>
<span style="cursor: pointer" onclick="hideAll(230);"><div class="cube230
leibniz"></div></span>
<span style="cursor: pointer" onclick="hideAll(171);"><div class="cube171
computers"></div></span>
<span style="cursor: pointer" onclick="hideAll(170);"><div class="cube170
computers"></div></span>
<span style="cursor: pointer" onclick="hideAll(159);"><div class="cube159
computers"></div></span>
<span style="cursor: pointer" onclick="hideAll(158);"><div class="cube158
computers"></div></span>
<span style="cursor: pointer" onclick="hideAll(157);"><div class="cube157
computers"></div></span>
<span style="cursor: pointer" onclick="hideAll(45);"><div class="cube45
towerofbabel"></div></span>
<span style="cursor: pointer" onclick="hideAll(1008);"><div class="cube1008
towerofbabel"></div></span>
<span style="cursor: pointer" onclick="hideAll(19);"><div class="cube19
stammworter"></div></span>
<span style="cursor: pointer" onclick="hideAll(18);"><div class="cube18
stammworter"></div></span>
<span style="cursor: pointer" onclick="hideAll(17);"><div class="cube17
stammworter"></div></span>
<span style="cursor: pointer" onclick="hideAll(234);"><div class="cube234
leibniz"></div></span>
```

```
<span style="cursor: pointer" onclick="hideAll(235);"><div class="cube235
leibniz"></div></span>
<span style="cursor: pointer" onclick="hideAll(1029);"><div class="cube1029
volvelles"></div></span>
<span style="cursor: pointer" onclick="hideAll(1030);"><div class="cube1030
volvelles"></div></span>
<span style="cursor: pointer" onclick="hideAll(46);"><div class="cube46
materiality"></div></span>
<span style="cursor: pointer" onclick="hideAll(4611);"><div class="cube4611
materiality"></div></span>
<span style="cursor: pointer" onclick="hideAll(4622);"><div class="cube4622
materiality"></div></span>
<span style="cursor: pointer" onclick="hideAll(172);"><div class="cube172
computers"></div></span>
<span style="cursor: pointer" onclick="hideAll(173);"><div class="cube173
computers"></div></span>
<span style="cursor: pointer" onclick="hideAll(174);"><div class="cube174
computers"></div></span>
<span style="cursor: pointer" onclick="hideAll(20);"><div class="cube20
stammworter"></div></span>
<span style="cursor: pointer" onclick="hideAll(21);"><div class="cube21
stammworter"></div></span>
<span style="cursor: pointer" onclick="hideAll(22);"><div class="cube22
stammworter"></div></span>
<span style="cursor: pointer" onclick="hideAll(23);"><div class="cube23
stammworter"></div></span>
<span style="cursor: pointer" onclick="hideAll(1017);"><div class="cube1017
leibniz"></div></span>
<span style="cursor: pointer" onclick="hideAll(1019);"><div class="cube1019
leibniz"></div></span>
<span style="cursor: pointer" onclick="hideAll(1);"><div class="cube1
volvelles"></div></span>
<span style="cursor: pointer" onclick="hideAll(2);"><div class="cube2
volvelles"></div></span>
<span style="cursor: pointer" onclick="hideAll(4655);"><div class="cube4655
materiality"></div></span>
<span style="cursor: pointer" onclick="hideAll(4644);"><div class="cube4644
materiality"></div></span>
<span style="cursor: pointer" onclick="hideAll(4633);"><div class="cube4633
materiality"></div></span>
<span style="cursor: pointer" onclick="hideAll(177);"><div class="cube177
computers"></div></span>
<span style="cursor: pointer" onclick="hideAll(176);"><div class="cube176
computers"></div></span>
<span style="cursor: pointer" onclick="hideAll(175);"><div class="cube175
computers"></div></span>
<span style="cursor: pointer" onclick="hideAll(28);"><div class="cube28
stammworter"></div></span>
<span style="cursor: pointer" onclick="hideAll(27);"><div class="cube27
stammworter"></div></span>
<span style="cursor: pointer" onclick="hideAll(26);"><div class="cube26
stammworter"></div></span>
<span style="cursor: pointer" onclick="hideAll(25);"><div class="cube25
stammworter"></div></span>
<span style="cursor: pointer" onclick="hideAll(24);"><div class="cube24
stammworter"></div></span>
<span style="cursor: pointer" onclick="hideAll(5);"><div class="cube5
volvelles"></div></span>
```

```
<span style="cursor: pointer" onclick="hideAll(4);"><div class="cube4
volvelles"></div></span>
<span style="cursor: pointer" onclick="hideAll(3);"><div class="cube3
volvelles"></div></span>
<span style="cursor: pointer" onclick="hideAll(47);"><div class="cube47
materiality"></div></span>
<span style="cursor: pointer" onclick="hideAll(48);"><div class="cube48
materiality"></div></span>
<span style="cursor: pointer" onclick="hideAll(49);"><div class="cube49
materiality"></div></span>
<span style="cursor: pointer" onclick="hideAll(178);"><div class="cube178
computers"></div></span>
<span style="cursor: pointer" onclick="hideAll(179);"><div class="cube179
computers"></div></span>
<span style="cursor: pointer" onclick="hideAll(180);"><div class="cube180
computers"></div></span>
<span style="cursor: pointer" onclick="hideAll(29);"><div class="cube29
stammworter"></div></span>
<span style="cursor: pointer" onclick="hideAll(30);"><div class="cube30
stammworter"></div></span>
<span style="cursor: pointer" onclick="hideAll(31);"><div class="cube31
stammworter"></div></span>
<span style="cursor: pointer" onclick="hideAll(1007);"><div class="cube1007
stammworter"></div></span>
<span style="cursor: pointer" onclick="hideAll(6);"><div class="cube6
volvelles"></div></span>
<span style="cursor: pointer" onclick="hideAll(7);"><div class="cube7
volvelles"></div></span>
<span style="cursor: pointer" onclick="hideAll(8);"><div class="cube8
volvelles"></div></span>
<span style="cursor: pointer" onclick="hideAll(9);"><div class="cube9
volvelles"></div></span>
<span style="cursor: pointer" onclick="hideAll(412);"><div class="cube412
materiality"></div></span>
<span style="cursor: pointer" onclick="hideAll(411);"><div class="cube411
materiality"></div></span>
<span style="cursor: pointer" onclick="hideAll(12011);"><div class="cube12011
materiality"></div></span>
<span style="cursor: pointer" onclick="hideAll(184);"><div class="cube184
computers"></div></span>
<span style="cursor: pointer" onclick="hideAll(183);"><div class="cube183
computers"></div></span>
<span style="cursor: pointer" onclick="hideAll(182);"><div class="cube182
computers"></div></span>
<span style="cursor: pointer" onclick="hideAll(181);"><div class="cube181
computers"></div></span>
<span style="cursor: pointer" onclick="hideAll(400);"><div class="cube400
volvelles"></div></span>
<span style="cursor: pointer" onclick="hideAll(57);"><div class="cube57
volvelles"></div></span>
<span style="cursor: pointer" onclick="hideAll(1000);"><div class="cube1000
volvelles"></div></span>
<span style="cursor: pointer" onclick="hideAll(13);"><div class="cube13
volvelles"></div></span>
<span style="cursor: pointer" onclick="hideAll(12);"><div class="cube12
volvelles"></div></span>
<span style="cursor: pointer" onclick="hideAll(11);"><div class="cube11
volvelles"></div></span>
```

```
<span style="cursor: pointer" onclick="hideAll(10);"><div class="cube10
volvelles"></div></span>
<span style="cursor: pointer" onclick="hideAll(413);"><div class="cube413
materiality"></div></span>
<span style="cursor: pointer" onclick="hideAll(414);"><div class="cube414
materiality"></div></span>
<span style="cursor: pointer" onclick="hideAll(415);"><div class="cube415
materiality"></div></span>
<span style="cursor: pointer" onclick="hideAll(32);"><div class="cube32
kabbalah"></div></span>
<span style="cursor: pointer" onclick="hideAll(33);"><div class="cube33
kabbalah"></div></span>
<span style="cursor: pointer" onclick="hideAll(185);"><div class="cube185
computers"></div></span>
<span style="cursor: pointer" onclick="hideAll(1032);"><div class="cube1032
computers"></div></span>
<span style="cursor: pointer" onclick="hideAll(401);"><div class="cube401
volvelles"></div></span>
<span style="cursor: pointer" onclick="hideAll(402);"><div class="cube402
volvelles"></div></span>
<span style="cursor: pointer" onclick="hideAll(403);"><div class="cube403
volvelles"></div></span>
<span style="cursor: pointer" onclick="hideAll(404);"><div class="cube404
volvelles"></div></span>
<span style="cursor: pointer" onclick="hideAll(405);"><div class="cube405
volvelles"></div></span>
<span style="cursor: pointer" onclick="hideAll(406);"><div class="cube406
volvelles"></div></span>
<span style="cursor: pointer" onclick="hideAll(407);"><div class="cube407
volvelles"></div></span>
<span style="cursor: pointer" onclick="hideAll(418);"><div class="cube418
materiality"></div></span>
<span style="cursor: pointer" onclick="hideAll(417);"><div class="cube417
materiality"></div></span>
<span style="cursor: pointer" onclick="hideAll(416);"><div class="cube416
materiality"></div></span>
<span style="cursor: pointer" onclick="hideAll(34);"><div class="cube34
kabbalah"></div></span>
<span style="cursor: pointer" onclick="hideAll(35);"><div class="cube35
kabbalah"></div></span>
<span style="cursor: pointer" onclick="hideAll(36);"><div class="cube36
kabbalah"></div></span>
<span style="cursor: pointer" onclick="hideAll(37);"><div class="cube37
kabbalah"></div></span>
<span style="cursor: pointer" onclick="hideAll(303);"><div class="cube303
volvelles"></div></span>
<span style="cursor: pointer" onclick="hideAll(302);"><div class="cube302
volvelles"></div></span>
<span style="cursor: pointer" onclick="hideAll(301);"><div class="cube301
volvelles"></div></span>
<span style="cursor: pointer" onclick="hideAll(300);"><div class="cube300
volvelles"></div></span>
<span style="cursor: pointer" onclick="hideAll(410);"><div class="cube410
volvelles"></div></span>
<span style="cursor: pointer" onclick="hideAll(409);"><div class="cube409
volvelles"></div></span>
<span style="cursor: pointer" onclick="hideAll(408);"><div class="cube408
volvelles"></div></span>
```

```
<span style="cursor: pointer" onclick="hideAll(419);"><div class="cube419
materiality"></div></span>
<span style="cursor: pointer" onclick="hideAll(1009);"><div class="cube1009
materiality"></div></span>
<span style="cursor: pointer" onclick="hideAll(1010);"><div class="cube1010
materiality"></div></span>
<span style="cursor: pointer" onclick="hideAll(41);"><div class="cube41
kabbalah"></div></span>
<span style="cursor: pointer" onclick="hideAll(40);"><div class="cube40
kabbalah"></div></span>
<span style="cursor: pointer" onclick="hideAll(39);"><div class="cube39
kabbalah"></div></span>
<span style="cursor: pointer" onclick="hideAll(38);"><div class="cube38
kabbalah"></div></span>
<span style="cursor: pointer" onclick="hideAll(304);"><div class="cube304
volvelles"></div></span>
<span style="cursor: pointer" onclick="hideAll(305);"><div class="cube305
volvelles"></div></span>
<span style="cursor: pointer" onclick="hideAll(306);"><div class="cube306
volvelles"></div></span>
<span style="cursor: pointer" onclick="hideAll(307);"><div class="cube307
volvelles"></div></span>
<span style="cursor: pointer" onclick="hideAll(308);"><div class="cube308
volvelles"></div></span>
<span style="cursor: pointer" onclick="hideAll(309);"><div class="cube309
volvelles"></div></span>
<span style="cursor: pointer" onclick="hideAll(310);"><div class="cube310
volvelles"></div></span>
<span style="cursor: pointer" onclick="hideAll(1033);"><div class="cube1033
materiality"></div></span>
<span style="cursor: pointer" onclick="hideAll(1031);"><div class="cube1031
materiality"></div></span>
<span style="cursor: pointer" onclick="hideAll(1015);"><div class="cube1015
materiality"></div></span>
<span style="cursor: pointer" onclick="hideAll(42);"><div class="cube42
kabbalah"></div></span>
<span style="cursor: pointer" onclick="hideAll(1027);"><div class="cube1027
hand"></div></span>
<span style="cursor: pointer" onclick="hideAll(1014);"><div class="cube1014
hand"></div></span>
<span style="cursor: pointer" onclick="hideAll(1013);"><div class="cube1013
hand"></div></span>
<span style="cursor: pointer" onclick="hideAll(317);"><div class="cube317
volvelles"></div></span>
<span style="cursor: pointer" onclick="hideAll(316);"><div class="cube316
volvelles"></div></span>
<span style="cursor: pointer" onclick="hideAll(315);"><div class="cube315
volvelles"></div></span>
<span style="cursor: pointer" onclick="hideAll(314);"><div class="cube314
volvelles"></div></span>
<span style="cursor: pointer" onclick="hideAll(313);"><div class="cube313
volvelles"></div></span>
<span style="cursor: pointer" onclick="hideAll(312);"><div class="cube312
volvelles"></div></span>
<span style="cursor: pointer" onclick="hideAll(311);"><div class="cube311
volvelles"></div></span>
<span style="cursor: pointer" onclick="hideAll(53);"><div class="cube53
hand"></div></span>
```

```
<span style="cursor: pointer" onclick="hideAll(5511);"><div class="cube5511
hand"></div></span>
<span style="cursor: pointer" onclick="hideAll(5522);"><div class="cube5522
hand"></div></span>
<span style="cursor: pointer" onclick="hideAll(5622);"><div class="cube5622
hand"></div></span>
<span style="cursor: pointer" onclick="hideAll(5633);"><div class="cube5633
hand"></div></span>
<span style="cursor: pointer" onclick="hideAll(1001);"><div class="cube1001
hand"></div></span>
<span style="cursor: pointer" onclick="hideAll(1002);"><div class="cube1002
hand"></div></span>
<span style="cursor: pointer" onclick="hideAll(318);"><div class="cube318
volvelles"></div></span>
<span style="cursor: pointer" onclick="hideAll(319);"><div class="cube319
volvelles"></div></span>
<span style="cursor: pointer" onclick="hideAll(320);"><div class="cube320
volvelles"></div></span>
<span style="cursor: pointer" onclick="hideAll(321);"><div class="cube321
volvelles"></div></span>
<span style="cursor: pointer" onclick="hideAll(322);"><div class="cube322
volvelles"></div></span>
<span style="cursor: pointer" onclick="hideAll(323);"><div class="cube323
volvelles"></div></span>
<span style="cursor: pointer" onclick="hideAll(1021);"><div class="cube1021
volvelles"></div></span>
<span style="cursor: pointer" onclick="hideAll(54);"><div class="cube54
hand"></div></span>
<span style="cursor: pointer" onclick="hideAll(55);"><div class="cube55
hand"></div></span>
<span style="cursor: pointer" onclick="hideAll(56);"><div class="cube56
hand"></div></span>
<span style="cursor: pointer" onclick="hideAll(5611);"><div class="cube5611
hand"></div></span>
<span style="cursor: pointer" onclick="hideAll(5644);"><div class="cube5644
hand"></div></span>
<span style="cursor: pointer" onclick="hideAll(58);"><div class="cube58
hand"></div></span>
<span style="cursor: pointer" onclick="hideAll(1012);"><div class="cube1012
hand"></div></span>
<span style="cursor: pointer" onclick="hideAll(190);"><div class="cube190
volvelles"></div></span>
<span style="cursor: pointer" onclick="hideAll(1028);"><div class="cube1028
volvelles"></div></span>
<span style="cursor: pointer" onclick="hideAll(1026);"><div class="cube1026
volvelles"></div></span>
<span style="cursor: pointer" onclick="hideAll(1025);"><div class="cube1025
volvelles"></div></span>
<span style="cursor: pointer" onclick="hideAll(1024);"><div class="cube1024
volvelles"></div></span>
<span style="cursor: pointer" onclick="hideAll(1023);"><div class="cube1023
volvelles"></div></span>
<span style="cursor: pointer" onclick="hideAll(1022);"><div class="cube1022
volvelles"></div></span>

<span style="cursor: pointer" onclick="hideAll(14833);"><div class="back14833
g"></div></span>
```

```
<span style="cursor: pointer" onclick="hideAll(161);"><div class="back161
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(162);"><div class="back162
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(163);"><div class="back163
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(164);"><div class="back164
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(100);"><div class="back100
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(101);"><div class="back101
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(102);"><div class="back102
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(60);"><div class="back60
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(61);"><div class="back61
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(62);"><div class="back62
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(63);"><div class="back63
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(200);"><div class="back200
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(201);"><div class="back201
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(254);"><div class="back254
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(253);"><div class="back253
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(252);"><div class="back252
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(251);"><div class="back251
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(250);"><div class="back250
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(105);"><div class="back105
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(104);"><div class="back104
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(103);"><div class="back103
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(67);"><div class="back67
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(66);"><div class="back66
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(65);"><div class="back65
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(64);"><div class="back64
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(203);"><div class="back203
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(202);"><div class="back202
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(255);"><div class="back255
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(256);"><div class="back256
g"></div></span>
```

```
<span style="cursor: pointer" onclick="hideAll(257);"><div class="back257
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(258);"><div class="back258
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(259);"><div class="back259
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(106);"><div class="back106
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(107);"><div class="back107
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(108);"><div class="back108
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(109);"><div class="back109
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(110);"><div class="back110
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(68);"><div class="back68
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(1003);"><div class="back1003
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(204);"><div class="back204
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(205);"><div class="back205
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(263);"><div class="back263
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(262);"><div class="back262
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(261);"><div class="back261
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(260);"><div class="back260
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(116);"><div class="back116
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(115);"><div class="back115
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(114);"><div class="back114
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(113);"><div class="back113
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(112);"><div class="back112
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(111);"><div class="back111
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(1005);"><div class="back1005
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(1004);"><div class="back1004
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(207);"><div class="back207
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(206);"><div class="back206
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(264);"><div class="back264
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(265);"><div class="back265
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(266);"><div class="back266
g"></div></span>
```

```
<span style="cursor: pointer" onclick="hideAll(267);"><div class="back267
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(117);"><div class="back117
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(118);"><div class="back118
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(119);"><div class="back119
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(11911);"><div class="back11911
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(120);"><div class="back120
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(121);"><div class="back121
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(1006);"><div class="back1006
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(1016);"><div class="back1016
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(208);"><div class="back208
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(209);"><div class="back209
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(1020);"><div class="back1020
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(1034);"><div class="back1034
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(128);"><div class="back128
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(127);"><div class="back127
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(149);"><div class="back149
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(126);"><div class="back126
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(1011);"><div class="back1011
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(124);"><div class="back124
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(123);"><div class="back123
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(122);"><div class="back122
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(213);"><div class="back213
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(212);"><div class="back212
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(211);"><div class="back211
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(210);"><div class="back210
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(129);"><div class="back129
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(143);"><div class="back143
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(12911);"><div class="back12911
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(130);"><div class="back130
g"></div></span>
```

```
<span style="cursor: pointer" onclick="hideAll(131);"><div class="back131
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(13111);"><div class="back13111
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(13122);"><div class="back13122
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(132);"><div class="back132
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(133);"><div class="back133
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(134);"><div class="back134
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(214);"><div class="back214
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(215);"><div class="back215
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(216);"><div class="back216
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(219);"><div class="back219
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(146);"><div class="back146
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(144);"><div class="back144
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(141);"><div class="back141
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(140);"><div class="back140
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(139);"><div class="back139
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(138);"><div class="back138
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(137);"><div class="back137
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(136);"><div class="back136
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(135);"><div class="back135
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(224);"><div class="back224
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(223);"><div class="back223
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(222);"><div class="back222
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(221);"><div class="back221
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(220);"><div class="back220
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(150);"><div class="back150
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(151);"><div class="back151
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(142);"><div class="back142
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(147);"><div class="back147
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(14711);"><div class="back14711
g"></div></span>
```

```
<span style="cursor: pointer" onclick="hideAll(14722);"><div class="back14722
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(148);"><div class="back148
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(14811);"><div class="back14811
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(14822);"><div class="back14822
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(225);"><div class="back225
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(226);"><div class="back226
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(227);"><div class="back227
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(228);"><div class="back228
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(229);"><div class="back229
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(152);"><div class="back152
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(153);"><div class="back153
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(154);"><div class="back154
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(155);"><div class="back155
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(156);"><div class="back156
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(43);"><div class="back43
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(44);"><div class="back44
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(14);"><div class="back14
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(15);"><div class="back15
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(16);"><div class="back16
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(233);"><div class="back233
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(232);"><div class="back232
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(231);"><div class="back231
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(230);"><div class="back230
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(171);"><div class="back171
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(170);"><div class="back170
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(159);"><div class="back159
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(158);"><div class="back158
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(157);"><div class="back157
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(45);"><div class="back45
g"></div></span>
```

```
<span style="cursor: pointer" onclick="hideAll(1008);"><div class="back1008
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(19);"><div class="back19
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(18);"><div class="back18
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(17);"><div class="back17
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(234);"><div class="back234
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(235);"><div class="back235
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(1029);"><div class="back1029
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(1030);"><div class="back1030
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(46);"><div class="back46
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(4611);"><div class="back4611
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(4622);"><div class="back4622
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(172);"><div class="back172
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(173);"><div class="back173
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(174);"><div class="back174
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(20);"><div class="back20
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(21);"><div class="back21
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(22);"><div class="back22
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(23);"><div class="back23
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(1017);"><div class="back1017
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(1019);"><div class="back1019
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(1);"><div class="back1
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(2);"><div class="back2
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(4655);"><div class="back4655
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(4644);"><div class="back4644
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(4633);"><div class="back4633
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(177);"><div class="back177
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(176);"><div class="back176
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(175);"><div class="back175
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(28);"><div class="back28
g"></div></span>
```

```
<span style="cursor: pointer" onclick="hideAll(27);"><div class="back27
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(26);"><div class="back26
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(25);"><div class="back25
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(24);"><div class="back24
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(5);"><div class="back5
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(4);"><div class="back4
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(3);"><div class="back3
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(47);"><div class="back47
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(48);"><div class="back48
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(49);"><div class="back49
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(178);"><div class="back178
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(179);"><div class="back179
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(180);"><div class="back180
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(29);"><div class="back29
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(30);"><div class="back30
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(31);"><div class="back31
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(1007);"><div class="back1007
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(6);"><div class="back6
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(7);"><div class="back7
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(8);"><div class="back8
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(9);"><div class="back9
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(412);"><div class="back412
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(411);"><div class="back411
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(12011);"><div class="back12011
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(184);"><div class="back184
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(183);"><div class="back183
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(182);"><div class="back182
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(181);"><div class="back181
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(400);"><div class="back400
g"></div></span>
```

```
<span style="cursor: pointer" onclick="hideAll(57);"><div class="back57
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(1000);"><div class="back1000
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(13);"><div class="back13
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(12);"><div class="back12
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(11);"><div class="back11
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(10);"><div class="back10
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(413);"><div class="back413
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(414);"><div class="back414
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(415);"><div class="back415
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(32);"><div class="back32
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(33);"><div class="back33
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(185);"><div class="back185
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(1032);"><div class="back1032
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(401);"><div class="back401
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(402);"><div class="back402
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(403);"><div class="back403
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(404);"><div class="back404
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(405);"><div class="back405
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(406);"><div class="back406
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(407);"><div class="back407
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(418);"><div class="back418
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(417);"><div class="back417
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(416);"><div class="back416
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(34);"><div class="back34
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(35);"><div class="back35
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(36);"><div class="back36
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(37);"><div class="back37
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(303);"><div class="back303
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(302);"><div class="back302
g"></div></span>
```

```
<span style="cursor: pointer" onclick="hideAll(301);"><div class="back301
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(300);"><div class="back300
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(410);"><div class="back410
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(409);"><div class="back409
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(408);"><div class="back408
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(419);"><div class="back419
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(1009);"><div class="back1009
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(1010);"><div class="back1010
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(41);"><div class="back41
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(40);"><div class="back40
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(39);"><div class="back39
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(38);"><div class="back38
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(304);"><div class="back304
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(305);"><div class="back305
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(306);"><div class="back306
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(307);"><div class="back307
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(308);"><div class="back308
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(309);"><div class="back309
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(310);"><div class="back310
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(1033);"><div class="back1033
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(1031);"><div class="back1031
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(1015);"><div class="back1015
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(42);"><div class="back42
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(1027);"><div class="back1027
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(1014);"><div class="back1014
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(1013);"><div class="back1013
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(317);"><div class="back317
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(316);"><div class="back316
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(315);"><div class="back315
g"></div></span>
```

```
<span style="cursor: pointer" onclick="hideAll(314);"><div class="back314
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(313);"><div class="back313
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(312);"><div class="back312
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(311);"><div class="back311
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(53);"><div class="back53
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(5511);"><div class="back5511
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(5522);"><div class="back5522
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(5622);"><div class="back5622
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(5633);"><div class="back5633
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(1001);"><div class="back1001
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(1002);"><div class="back1002
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(318);"><div class="back318
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(319);"><div class="back319
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(320);"><div class="back320
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(321);"><div class="back321
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(322);"><div class="back322
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(323);"><div class="back323
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(1021);"><div class="back1021
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(54);"><div class="back54
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(55);"><div class="back55
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(56);"><div class="back56
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(5611);"><div class="back5611
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(5644);"><div class="back5644
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(58);"><div class="back58
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(1012);"><div class="back1012
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(190);"><div class="back190
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(1028);"><div class="back1028
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(1026);"><div class="back1026
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(1025);"><div class="back1025
g"></div></span>
```

```
<span style="cursor: pointer" onclick="hideAll(1024);"><div class="back1024
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(1023);"><div class="back1023
g"></div></span>
<span style="cursor: pointer" onclick="hideAll(1022);"><div class="back1022
g"></div></span>
</div>
<div id="labelbox-shadow" class="displaynone"></div>
<div id="labelbox" class="displaynone"></div>
<div class="topleft-shadow"></div>
<div class="topright-shadow"></div>
<div class="bottomright-shadow"></div>
<div class="bottomleft-shadow"></div>
<div id="tlv" style="display:none;"></div>
<div id="tlh" style="display:none;"></div>
<div id="blv" style="display:none;"></div>
<div id="blh" style="display:none;"></div>
<div id="brv" style="display:none;"></div>
<div id="brh" style="display:none;"></div>
<div id="trv" style="display:none;"></div>
<div id="trh" style="display:none;"></div>
<div id="tl" style="display:none;"></div>
<div id="tr" style="display:none;"></div>
<div id="br" style="display:none;"></div>
<div id="bl" style="display:none;"></div>
<div id="yourtext">
<p>
<span style="color:#E6399B;font-size:18px;"><b>------------------------
</b></span>
</p>
<p>your text</p>
</div>
<div id="thesis-shadow">
<div class="page-inner-shadow"></div>
</div>
<a name="thesis"></a><div id="thesis"><div class="page-inner"></div>
</div>
</body>
</html>
```

```
/*
___style.css___
by Whitney Anne trettien

This is the main stylesheet for index.html.
/*

body {
      background-color: #606060;
      background-repeat: no-repeat;
      font-family: Verdana, Helvetica, Arial, san-serif;
}div#titlebox {
      font-family: Arial, sanserif;
      text-transform: uppercase;
      width: 155px;
      height: auto;
      position:absolute;
      left:10px;
      line-height:110%;
      top:30px;
}p.titlebox {
      font-family: Arial, sanserif;
      text-transform: uppercase;
      color:yellow;
      line-height:190%;
      margin: 0px 0px 0px 0px
}.intro {
      font-family: Verdana, Helvetica, Arial, san-serif;
      font-size: 1.1em;
      font-weight:bold;
      margin:10px 15px 10px 15px;
      color: #888888;
}p {
      margin: 10px 10px 10px 10px;
      line-height: 130%;
      font-size: .8em;
      position:absolute;
}.blockquote {
      font-size: .7em;
      line-height: 130%;
      margin: 6px 6px 6px 6px;
}blockquote {
      border-style: dotted;
      border-color: #d8d8d8;
      background-color: #f0f0f0;
      margin-left:30px;
      margin-right:30px;
      margin-top:15px;
}a:visited {
    text-decoration: none;
    color: #888888;
}a {
      text-decoration: none;
      color: #888888;
}a:hover {
      background-color: yellow;
      position:relative;
}a span {
```

```
      display: none;
}a:hover span {
      display: block;
      position: absolute;
      top:10px;
      left: 0px;
/* formatting only styles */
      padding: 5px;
      margin: 10px;
      z-index: 100;
      background: #f8f8f8;
      border: 1px dotted #E667AF;
      font-size: smaller;
      color: #404040;
      opacity: 0.9;
      width: 300px;
/* end formatting */
}div#labelbox {
      width: 140px;
      position: absolute;
      top: 390px;
      left: 12px;
      font-size: .65em;
      font-weight:bold;
      text-transform: uppercase;
      color: #808080;
      text-align: center;
      background-color:white;
}div#labelbox-shadow {
      width: 140px;
      position: absolute;
      top: 393px;
      left: 16px;
      font-size: .65em;
      font-weight:bold;
      text-transform: uppercase;
      color: #CCCCCC;
      text-align: center;
      background-color: #CCCCCC;
}#yourtext{
      position:absolute;
      top: 695px;
      left:10px;
      color:yellow;
      font-size:1.9em;
      text-align:center;
      width:160px;
      margin: 0px 0px 0px 0px;
      height:auto;
      font-family: arial, sanserif;
      font-weight:bold;
      line-height:20%;
      text-transform:uppercase;
      margin:0px 0px 0px 0px;
}#thesis {
      position: absolute;
      background-color: white;
      top: 695px;
```

```
      left: 180px;
      width: 820px;
      height: auto;
      color:black;
}#thesis-shadow {
      position: absolute;
      background-color: #CCCCCC;
      top: 700px;
      left: 185px;
      width: 820px;
      height: auto;
}div.page-inner-shadow {
      width: 800px;
}div.page-inner {
      width: 800px;
      background-color: white;
}div#tl {
      width: 400px;
      height: 300px;
      top: 17px;
      left: 177px;
      background-color: white;
      position:absolute;
}div#tr{
      width: 400px;
      height: 300px;
      top: 17px;
      left: 597px;
      background-color: white;
      position:absolute;
}div#bl {
      width: 400px;
      height: 300px;
      top: 337px;
      left: 177px;
      background-color: white;
      position:absolute;
}div#br {
      width: 400px;
      height: 300px;
      top: 337px;
      left: 597px;
      background-color: white;
      position:absolute;
}div#tlv {
      position: absolute;
      left: 585px;
      top: 150px;
}div#tlh {
      position: absolute;
      top: 325px;
      left: 360px;
}div#trv {
      position: absolute;
      left: 585px;
      top: 150px;
}div#trh {
      position: absolute;
```

```
      top: 325px;
      left: 780px;
}div#brv {
      position: absolute;
      top: 470px;
      left: 585px;
}div#brh {
      position: absolute;
      top: 325px;
      left: 780px;
}div#blv {
      position: absolute;
      top: 470px;
      left: 585px;
}div#blh {
      position: absolute;
      top: 325px;
      left: 360px;
}div#left-arrow {
      font-size: 0px;
      line-height: 0%;
      width: 0px;
      border-top: 20px solid #606060;
      border-right: 10px solid #888888;
      border-bottom: 20px     solid #606060;
}div#right-arrow {
      font-size: 0px;
      line-height: 0%;
      width: 0px;
      border-top: 20px solid #606060;
      border-left: 10px solid #888888;
      border-bottom: 20px solid #606060;
}div#up-arrow {
      font-size: 0px;
      line-height: 0%;
      width: 0px;
      border-bottom: 10px solid #888888;
      border-left: 20px solid #606060;
      border-right: 20px solid #606060;
}div#down-arrow {
      font-size: 0px;
      line-height: 0%;
      width: 0px;
      border-top: 10px solid #888888;
      border-left: 20px solid #606060;
      border-right: 20px solid #606060;
}.selected {
      opacity: 0.4;
      filter: alpha(opacity=40);
      -ms-filter:"progid:DXImageTransform.Microsoft.Alpha(Opacity=40)";
      -moz-opacity: 0.4;
}.displaynone {
      display:none;
}div#cubebox {
      width: 150px;
      height: auto;
      position: absolute;
      top: 430px;
```

```
      left: 12px;
}.g {
      background-color: #888888;
      width: 9px;
      height: 9px;
      position: absolute;
}.permutation {
      background-color: #E6399B;
      color: #E6399B;
}.cutups {
      background-color: #E6399B;
      color: #E6399B;
}.thirtyyearswar {
      background-color: yellow;
      color: yellow;
}.leibniz {
      background-color: #C9F76F;
      color: #C9F76F;
}.volvelles {
      background-color: yellow;
      color: yellow;
}.stammworter {
      background-color: #9FEE00;
      color: #9FEE00;
}.towerofbabel {
      background-color: #E667AF;
      color: #E667AF;
}.computers {
      background-color:yellow;
      color: yellow;
}.materiality {
      background-color: #CD0074;
      color: #CD0074;
}.kabbalah {
      background-color: #679B00;
      color: #679B00;
}.hand {
      background-color: #FFFF73;
      color: #FFFF73;
}div.back14833 {
      top:0px;
      left:0px;
}div.back161 {
      top:0px;
      left:10px;
}div.back162 {
      top:0px;
      left:20px;
}div.back163 {
      top:0px;
      left:30px;
}div.back164 {
      top:0px;
      left:40px;
}div.back100 {
      top:0px;
      left:50px;
}div.back101 {
```

```
      top:0px;
      left:60px;
}div.back102 {
      top:0px;
      left:70px;
}div.back60 {
      top:0px;
      left:80px;
}div.back61 {
      top:0px;
      left:90px;
}div.back62 {
      top:0px;
      left:100px;
}div.back63 {
      top:0px;
      left:110px;
}div.back200 {
      top:0px;
      left:120px;
}div.back201 {
      top:0px;
      left:130px;
}div.back254 {
      top:10px;
      left:0px;
}div.back253 {
      top:10px;
      left:10px;
}div.back252 {
      top:10px;
      left:20px;
}div.back251 {
      top:10px;
      left:30px;
}div.back250 {
      top:10px;
      left:40px;
}div.back105 {
      top:10px;
      left:50px;
}div.back104 {
      top:10px;
      left:60px;
}div.back103 {
      top:10px;
      left:70px;
}div.back67 {
      top:10px;
      left:80px;
}div.back66 {
      top:10px;
      left:90px;
}div.back65 {
      top:10px;
      left:100px;
}div.back64 {
      top:10px;
```

```
      left:110px;
}div.back203 {
      top:10px;
      left:120px;
}div.back202 {
      top:10px;
      left:130px;
}div.back255 {
      top:20px;
      left:0px;
}div.back256 {
      top:20px;
      left:10px;
}div.back257 {
      top:20px;
      left:20px;
}div.back258 {
      top:20px;
      left:30px;
}div.back259 {
      top:20px;
      left:40px;
}div.back106 {
      top:20px;
      left:50px;
}div.back107 {
      top:20px;
      left:60px;
}div.back108 {
      top:20px;
      left:70px;
}div.back109 {
      top:20px;
      left:80px;
}div.back110 {
      top:20px;
      left:90px;
}div.back68 {
      top:20px;
      left:100px;
}div.back1003 {
      top:20px;
      left:110px;
}div.back204 {
      top:20px;
      left:120px;
}div.back205 {
      top:20px;
      left:130px;
}div.back263 {
      top:30px;
      left:0px;
}div.back262 {
      top:30px;
      left:10px;
}div.back261 {
      top:30px;
      left:20px;
```

```
}div.back260 {
      top:30px;
      left:30px;
}div.back116 {
      top:30px;
      left:40px;
}div.back115 {
      top:30px;
      left:50px;
}div.back114 {
      top:30px;
      left:60px;
}div.back113 {
      top:30px;
      left:70px;
}div.back112 {
      top:30px;
      left:80px;
}div.back111 {
      top:30px;
      left:90px;
}div.back1005 {
      top:30px;
      left:100px;
}div.back1004 {
      top:30px;
      left:110px;
}div.back207 {
      top:30px;
      left:120px;
}div.back206 {
      top:30px;
      left:130px;
}div.back264 {
      top:40px;
      left:0px;
}div.back265 {
      top:40px;
      left:10px;
}div.back266 {
      top:40px;
      left:20px;
}div.back267 {
      top:40px;
      left:30px;
}div.back117 {
      top:40px;
      left:40px;
}div.back118 {
      top:40px;
      left:50px;
}div.back119 {
      top:40px;
      left:60px;
}div.back11911 {
      top:40px;
      left:70px;
}div.back120 {
```

```
      top:40px;
      left:80px;
}div.back121 {
      top:40px;
      left:90px;
}div.back1006 {
      top:40px;
      left:100px;
}div.back1016 {
      top:40px;
      left:110px;
}div.back208 {
      top:40px;
      left:120px;
}div.back209 {
      top:40px;
      left:130px;
}div.back1020 {
      top:50px;
      left:0px;
}div.back1034 {
      top:50px;
      left:10px;
}div.back128 {
      top:50px;
      left:20px;
}div.back127 {
      top:50px;
      left:30px;
}div.back149 {
      top:50px;
      left:40px;
}div.back126 {
      top:50px;
      left:50px;
}div.back1011 {
      top:50px;
      left:60px;
}div.back124 {
      top:50px;
      left:70px;
}div.back123 {
      top:50px;
      left:80px;
}div.back122 {
      top:50px;
      left:90px;
}div.back213 {
      top:50px;
      left:100px;
}div.back212 {
      top:50px;
      left:110px;
}div.back211 {
      top:50px;
      left:120px;
}div.back210 {
      top:50px;
```

```
      left:130px;
}div.back129 {
      top:60px;
      left:0px;
}div.back143 {
      top:60px;
      left:10px;
}div.back12911 {
      top:60px;
      left:20px;
}div.back130 {
      top:60px;
      left:30px;
}div.back131 {
      top:60px;
      left:40px;
}div.back13111 {
      top:60px;
      left:50px;
}div.back13122 {
      top:60px;
      left:60px;
}div.back132 {
      top:60px;
      left:70px;
}div.back133 {
      top:60px;
      left:80px;
}div.back134 {
      top:60px;
      left:90px;
}div.back214 {
      top:60px;
      left:100px;
}div.back215 {
      top:60px;
      left:110px;
}div.back216 {
      top:60px;
      left:120px;
}div.back219 {
      top:60px;
      left:130px;
}div.back146 {
      top:70px;
      left:0px;
}div.back144 {
      top:70px;
      left:10px;
}div.back141 {
      top:70px;
      left:20px;
}div.back140 {
      top:70px;
      left:30px;
}div.back139 {
      top:70px;
      left:40px;
```

```
}div.back138 {
      top:70px;
      left:50px;
}div.back137 {
      top:70px;
      left:60px;
}div.back136 {
      top:70px;
      left:70px;
}div.back135 {
      top:70px;
      left:80px;
}div.back224 {
      top:70px;
      left:90px;
}div.back223 {
      top:70px;
      left:100px;
}div.back222 {
      top:70px;
      left:110px;
}div.back221 {
      top:70px;
      left:120px;
}div.back220 {
      top:70px;
      left:130px;
}div.back150 {
      top:80px;
      left:0px;
}div.back151 {
      top:80px;
      left:10px;
}div.back142 {
      top:80px;
      left:20px;
}div.back147 {
      top:80px;
      left:30px;
}div.back14711 {
      top:80px;
      left:40px;
}div.back14722 {
      top:80px;
      left:50px;
}div.back148 {
      top:80px;
      left:60px;
}div.back14811 {
      top:80px;
      left:70px;
}div.back14822 {
      top:80px;
      left:80px;
}div.back225 {
      top:80px;
      left:90px;
}div.back226 {
```

```
      top:80px;
      left:100px;
}div.back227 {
      top:80px;
      left:110px;
}div.back228 {
      top:80px;
      left:120px;
}div.back229 {
      top:80px;
      left:130px;
}div.back152 {
      top:90px;
      left:0px;
}div.back153 {
      top:90px;
      left:10px;
}div.back154 {
      top:90px;
      left:20px;
}div.back155 {
      top:90px;
      left:30px;
}div.back156 {
      top:90px;
      left:40px;
}div.back43 {
      top:90px;
      left:50px;
}div.back44 {
      top:90px;
      left:60px;
}div.back14 {
      top:90px;
      left:70px;
}div.back15 {
      top:90px;
      left:80px;
}div.back16 {
      top:90px;
      left:90px;
}div.back233 {
      top:90px;
      left:100px;
}div.back232 {
      top:90px;
      left:110px;
}div.back231 {
      top:90px;
      left:120px;
}div.back230 {
      top:90px;
      left:130px;
}div.back171 {
      top:100px;
      left:0px;
}div.back170 {
      top:100px;
```

```
      left:10px;
}div.back159 {
      top:100px;
      left:20px;
}div.back158 {
      top:100px;
      left:30px;
}div.back157 {
      top:100px;
      left:40px;
}div.back45 {
      top:100px;
      left:50px;
}div.back1008 {
      top:100px;
      left:60px;
}div.back19 {
      top:100px;
      left:70px;
}div.back18 {
      top:100px;
      left:80px;
}div.back17 {
      top:100px;
      left:90px;
}div.back234 {
      top:100px;
      left:100px;
}div.back235 {
      top:100px;
      left:110px;
}div.back1029 {
      top:100px;
      left:120px;
}div.back1030 {
      top:100px;
      left:130px;
}div.back46 {
      top:110px;
      left:0px;
}div.back4611 {
      top:110px;
      left:10px;
}div.back4622 {
      top:110px;
      left:20px;
}div.back172 {
      top:110px;
      left:30px;
}div.back173 {
      top:110px;
      left:40px;
}div.back174 {
      top:110px;
      left:50px;
}div.back20 {
      top:110px;
      left:60px;
```

```
}div.back21 {
      top:110px;
      left:70px;
}div.back22 {
      top:110px;
      left:80px;
}div.back23 {
      top:110px;
      left:90px;
}div.back1017 {
      top:110px;
      left:100px;
}div.back1019 {
      top:110px;
      left:110px;
}div.back1 {
      top:110px;
      left:120px;
}div.back2 {
      top:110px;
      left:130px;
}div.back4655 {
      top:120px;
      left:0px;
}div.back4644 {
      top:120px;
      left:10px;
}div.back4633 {
      top:120px;
      left:20px;
}div.back177 {
      top:120px;
      left:30px;
}div.back176 {
      top:120px;
      left:40px;
}div.back175 {
      top:120px;
      left:50px;
}div.back28 {
      top:120px;
      left:60px;
}div.back27 {
      top:120px;
      left:70px;
}div.back26 {
      top:120px;
      left:80px;
}div.back25 {
      top:120px;
      left:90px;
}div.back24 {
      top:120px;
      left:100px;
}div.back5 {
      top:120px;
      left:110px;
}div.back4 {
```

```
      top:120px;
      left:120px;
}div.back3 {
      top:120px;
      left:130px;
}div.back47 {
      top:130px;
      left:0px;
}div.back48 {
      top:130px;
      left:10px;
}div.back49 {
      top:130px;
      left:20px;
}div.back178 {
      top:130px;
      left:30px;
}div.back179 {
      top:130px;
      left:40px;
}div.back180 {
      top:130px;
      left:50px;
}div.back29 {
      top:130px;
      left:60px;
}div.back30 {
      top:130px;
      left:70px;
}div.back31 {
      top:130px;
      left:80px;
}div.back1007 {
      top:130px;
      left:90px;
}div.back6 {
      top:130px;
      left:100px;
}div.back7 {
      top:130px;
      left:110px;
}div.back8 {
      top:130px;
      left:120px;
}div.back9 {
      top:130px;
      left:130px;
}div.back412 {
      top:140px;
      left:0px;
}div.back411 {
      top:140px;
      left:10px;
}div.back12011 {
      top:140px;
      left:20px;
}div.back184 {
      top:140px;
```

```
      left:30px;
}div.back183 {
      top:140px;
      left:40px;
}div.back182 {
      top:140px;
      left:50px;
}div.back181 {
      top:140px;
      left:60px;
}div.back400 {
      top:140px;
      left:70px;
}div.back57 {
      top:140px;
      left:80px;
}div.back1000 {
      top:140px;
      left:90px;
}div.back13 {
      top:140px;
      left:100px;
}div.back12 {
      top:140px;
      left:110px;
}div.back11 {
      top:140px;
      left:120px;
}div.back10 {
      top:140px;
      left:130px;
}div.back413 {
      top:150px;
      left:0px;
}div.back414 {
      top:150px;
      left:10px;
}div.back415 {
      top:150px;
      left:20px;
}div.back32 {
      top:150px;
      left:30px;
}div.back33 {
      top:150px;
      left:40px;
}div.back185 {
      top:150px;
      left:50px;
}div.back1032 {
      top:150px;
      left:60px;
}div.back401 {
      top:150px;
      left:70px;
}div.back402 {
      top:150px;
      left:80px;
```

```
}div.back403 {
      top:150px;
      left:90px;
}div.back404 {
      top:150px;
      left:100px;
}div.back405 {
      top:150px;
      left:110px;
}div.back406 {
      top:150px;
      left:120px;
}div.back407 {
      top:150px;
      left:130px;
}div.back418 {
      top:160px;
      left:0px;
}div.back417 {
      top:160px;
      left:10px;
}div.back416 {
      top:160px;
      left:20px;
}div.back34 {
      top:160px;
      left:30px;
}div.back35 {
      top:160px;
      left:40px;
}div.back36 {
      top:160px;
      left:50px;
}div.back37 {
      top:160px;
      left:60px;
}div.back303 {
      top:160px;
      left:70px;
}div.back302 {
      top:160px;
      left:80px;
}div.back301 {
      top:160px;
      left:90px;
}div.back300 {
      top:160px;
      left:100px;
}div.back410 {
      top:160px;
      left:110px;
}div.back409 {
      top:160px;
      left:120px;
}div.back408 {
      top:160px;
      left:130px;
}div.back419 {
```

```
      top:170px;
      left:0px;
}div.back1009 {
      top:170px;
      left:10px;
}div.back1010 {
      top:170px;
      left:20px;
}div.back41 {
      top:170px;
      left:30px;
}div.back40 {
      top:170px;
      left:40px;
}div.back39 {
      top:170px;
      left:50px;
}div.back38 {
      top:170px;
      left:60px;
}div.back304 {
      top:170px;
      left:70px;
}div.back305 {
      top:170px;
      left:80px;
}div.back306 {
      top:170px;
      left:90px;
}div.back307 {
      top:170px;
      left:100px;
}div.back308 {
      top:170px;
      left:110px;
}div.back309 {
      top:170px;
      left:120px;
}div.back310 {
      top:170px;
      left:130px;
}div.back1033 {
      top:180px;
      left:0px;
}div.back1031 {
      top:180px;
      left:10px;
}div.back1015 {
      top:180px;
      left:20px;
}div.back42 {
      top:180px;
      left:30px;
}div.back1027 {
      top:180px;
      left:40px;
}div.back1014 {
      top:180px;
```

```
      left:50px;
}div.back1013 {
      top:180px;
      left:60px;
}div.back317 {
      top:180px;
      left:70px;
}div.back316 {
      top:180px;
      left:80px;
}div.back315 {
      top:180px;
      left:90px;
}div.back314 {
      top:180px;
      left:100px;
}div.back313 {
      top:180px;
      left:110px;
}div.back312 {
      top:180px;
      left:120px;
}div.back311 {
      top:180px;
      left:130px;
}div.back53 {
      top:190px;
      left:0px;
}div.back5511 {
      top:190px;
      left:10px;
}div.back5522 {
      top:190px;
      left:20px;
}div.back5622 {
      top:190px;
      left:30px;
}div.back5633 {
      top:190px;
      left:40px;
}div.back1001 {
      top:190px;
      left:50px;
}div.back1002 {
      top:190px;
      left:60px;
}div.back318 {
      top:190px;
      left:70px;
}div.back319{
      top:190px;
      left:80px;
}div.back320 {
      top:190px;
      left:90px;
}div.back321 {
      top:190px;
      left:100px;
```

```
}div.back322 {
      top:190px;
      left:110px;
}div.back323 {
      top:190px;
      left:120px;
}div.back1021 {
      top:190px;
      left:130px;
}div.back54 {
      top:200px;
      left:0px;
}div.back55 {
      top:200px;
      left:10px;
}div.back56 {
      top:200px;
      left:20px;
}div.back5611 {
      top:200px;
      left:30px;
}div.back5644 {
      top:200px;
      left:40px;
}div.back58 {
      top:200px;
      left:50px;
}div.back1012 {
      top:200px;
      left:60px;
}div.back190 {
      top:200px;
      left:70px;
}div.back1028 {
      top:200px;
      left:80px;
}div.back1026 {
      top:200px;
      left:90px;
}div.back1025 {
      top:200px;
      left:100px;
}div.back1024 {
      top:200px;
      left:110px;
}div.back1023 {
      top:200px;
      left:120px;
}div.back1022 {
      top:200px;
      left:130px;
}


div.cube14833 {
      width: 9px;
      height:9px;
      position:absolute;
```

```
      top:0px;
      left:0px;
}div.cube161 {
      width: 9px;
      height:9px;
      position:absolute;
      top:0px;
      left:10px;
}div.cube162 {
      width: 9px;
      height:9px;
      position:absolute;
      top:0px;
      left:20px;
}div.cube163 {
      width: 9px;
      height:9px;
      position:absolute;
      top:0px;
      left:30px;
}div.cube164 {
      width: 9px;
      height:9px;
      position:absolute;
      top:0px;
      left:40px;
}div.cube100 {
      width: 9px;
      height:9px;
      position:absolute;
      top:0px;
      left:50px;
}div.cube101 {
      width: 9px;
      height:9px;
      position:absolute;
      top:0px;
      left:60px;
}div.cube102 {
      width: 9px;
      height:9px;
      position:absolute;
      top:0px;
      left:70px;
}div.cube60 {
      width: 9px;
      height:9px;
      position:absolute;
      top:0px;
      left:80px;
}div.cube61 {
      width: 9px;
      height:9px;
      position:absolute;
      top:0px;
      left:90px;
}div.cube62 {
      width: 9px;
```

```
      height:9px;
      position:absolute;
      top:0px;
      left:100px;
}div.cube63 {
      width: 9px;
      height:9px;
      position:absolute;
      top:0px;
      left:110px;
}div.cube200 {
      width: 9px;
      height:9px;
      position:absolute;
      top:0px;
      left:120px;
}div.cube201 {
      width: 9px;
      height:9px;
      position:absolute;
      top:0px;
      left:130px;
}div.cube254 {
      width: 9px;
      height:9px;
      position:absolute;
      top:10px;
      left:0px;
}div.cube253 {
      width: 9px;
      height:9px;
      position:absolute;
      top:10px;
      left:10px;
}div.cube252 {
      width: 9px;
      height:9px;
      position:absolute;
      top:10px;
      left:20px;
}div.cube251 {
      width: 9px;
      height:9px;
      position:absolute;
      top:10px;
      left:30px;
}div.cube250 {
      width: 9px;
      height:9px;
      position:absolute;
      top:10px;
      left:40px;
}div.cube105 {
      width: 9px;
      height:9px;
      position:absolute;
      top:10px;
      left:50px;
```

```
}div.cube104 {
      width: 9px;
      height:9px;
      position:absolute;
      top:10px;
      left:60px;
}div.cube103 {
      width: 9px;
      height:9px;
      position:absolute;
      top:10px;
      left:70px;
}div.cube67 {
      width: 9px;
      height:9px;
      position:absolute;
      top:10px;
      left:80px;
}div.cube66 {
      width: 9px;
      height:9px;
      position:absolute;
      top:10px;
      left:90px;
}div.cube65 {
      width: 9px;
      height:9px;
      position:absolute;
      top:10px;
      left:100px;
}div.cube64 {
      width: 9px;
      height:9px;
      position:absolute;
      top:10px;
      left:110px;
}div.cube203 {
      width: 9px;
      height:9px;
      position:absolute;
      top:10px;
      left:120px;
}div.cube202 {
      width: 9px;
      height:9px;
      position:absolute;
      top:10px;
      left:130px;
}div.cube255 {
      width: 9px;
      height:9px;
      position:absolute;
      top:20px;
      left:0px;
}div.cube256 {
      width: 9px;
      height:9px;
      position:absolute;
```

```
      top:20px;
      left:10px;
}div.cube257 {
      width: 9px;
      height:9px;
      position:absolute;
      top:20px;
      left:20px;
}div.cube258 {
      width: 9px;
      height:9px;
      position:absolute;
      top:20px;
      left:30px;
}div.cube259 {
      width: 9px;
      height:9px;
      position:absolute;
      top:20px;
      left:40px;
}div.cube106 {
      width: 9px;
      height:9px;
      position:absolute;
      top:20px;
      left:50px;
}div.cube107 {
      width: 9px;
      height:9px;
      position:absolute;
      top:20px;
      left:60px;
}div.cube108 {
      width: 9px;
      height:9px;
      position:absolute;
      top:20px;
      left:70px;
}div.cube109 {
      width: 9px;
      height:9px;
      position:absolute;
      top:20px;
      left:80px;
}div.cube110 {
      width: 9px;
      height:9px;
      position:absolute;
      top:20px;
      left:90px;
}div.cube68 {
      width: 9px;
      height:9px;
      position:absolute;
      top:20px;
      left:100px;
}div.cube1003 {
      width: 9px;
```

```
      height:9px;
      position:absolute;
      top:20px;
      left:110px;
}div.cube204 {
      width: 9px;
      height:9px;
      position:absolute;
      top:20px;
      left:120px;
}div.cube205 {
      width: 9px;
      height:9px;
      position:absolute;
      top:20px;
      left:130px;
}div.cube263 {
      width: 9px;
      height:9px;
      position:absolute;
      top:30px;
      left:0px;
}div.cube262 {
      width: 9px;
      height:9px;
      position:absolute;
      top:30px;
      left:10px;
}div.cube261 {
      width: 9px;
      height:9px;
      position:absolute;
      top:30px;
      left:20px;
}div.cube260 {
      width: 9px;
      height:9px;
      position:absolute;
      top:30px;
      left:30px;
}div.cube116 {
      width: 9px;
      height:9px;
      position:absolute;
      top:30px;
      left:40px;
}div.cube115 {
      width: 9px;
      height:9px;
      position:absolute;
      top:30px;
      left:50px;
}div.cube114 {
      width: 9px;
      height:9px;
      position:absolute;
      top:30px;
      left:60px;
```

```
}div.cube113 {
      width: 9px;
      height:9px;
      position:absolute;
      top:30px;
      left:70px;
}div.cube112 {
      width: 9px;
      height:9px;
      position:absolute;
      top:30px;
      left:80px;
}div.cube111 {
      width: 9px;
      height:9px;
      position:absolute;
      top:30px;
      left:90px;
}div.cube1005 {
      width: 9px;
      height:9px;
      position:absolute;
      top:30px;
      left:100px;
}div.cube1004 {
      width: 9px;
      height:9px;
      position:absolute;
      top:30px;
      left:110px;
}div.cube207 {
      width: 9px;
      height:9px;
      position:absolute;
      top:30px;
      left:120px;
}div.cube206 {
      width: 9px;
      height:9px;
      position:absolute;
      top:30px;
      left:130px;
}div.cube264 {
      width: 9px;
      height:9px;
      position:absolute;
      top:40px;
      left:0px;
}div.cube265 {
      width: 9px;
      height:9px;
      position:absolute;
      top:40px;
      left:10px;
}div.cube266 {
      width: 9px;
      height:9px;
      position:absolute;
```

```
      top:40px;
      left:20px;
}div.cube267 {
      width: 9px;
      height:9px;
      position:absolute;
      top:40px;
      left:30px;
}div.cube117 {
      width: 9px;
      height:9px;
      position:absolute;
      top:40px;
      left:40px;
}div.cube118 {
      width: 9px;
      height:9px;
      position:absolute;
      top:40px;
      left:50px;
}div.cube119 {
      width: 9px;
      height:9px;
      position:absolute;
      top:40px;
      left:60px;
}div.cube11911 {
      width: 9px;
      height:9px;
      position:absolute;
      top:40px;
      left:70px;
}div.cube120 {
      width: 9px;
      height:9px;
      position:absolute;
      top:40px;
      left:80px;
}div.cube121 {
      width: 9px;
      height:9px;
      position:absolute;
      top:40px;
      left:90px;
}div.cube1006 {
      width: 9px;
      height:9px;
      position:absolute;
      top:40px;
      left:100px;
}div.cube1016 {
      width: 9px;
      height:9px;
      position:absolute;
      top:40px;
      left:110px;
}div.cube208 {
      width: 9px;
```

```
      height:9px;
      position:absolute;
      top:40px;
      left:120px;
}div.cube209 {
      width: 9px;
      height:9px;
      position:absolute;
      top:40px;
      left:130px;
}div.cube1020 {
      width: 9px;
      height:9px;
      position:absolute;
      top:50px;
      left:0px;
}div.cube1034 {
      width: 9px;
      height:9px;
      position:absolute;
      top:50px;
      left:10px;
}div.cube128 {
      width: 9px;
      height:9px;
      position:absolute;
      top:50px;
      left:20px;
}div.cube127 {
      width: 9px;
      height:9px;
      position:absolute;
      top:50px;
      left:30px;
}div.cube149 {
      width: 9px;
      height:9px;
      position:absolute;
      top:50px;
      left:40px;
}div.cube126 {
      width: 9px;
      height:9px;
      position:absolute;
      top:50px;
      left:50px;
}div.cube1011 {
      width: 9px;
      height:9px;
      position:absolute;
      top:50px;
      left:60px;
}div.cube124 {
      width: 9px;
      height:9px;
      position:absolute;
      top:50px;
      left:70px;
```

```
}div.cube123 {
      width: 9px;
      height:9px;
      position:absolute;
      top:50px;
      left:80px;
}div.cube122 {
      width: 9px;
      height:9px;
      position:absolute;
      top:50px;
      left:90px;
}div.cube213 {
      width: 9px;
      height:9px;
      position:absolute;
      top:50px;
      left:100px;
}div.cube212 {
      width: 9px;
      height:9px;
      position:absolute;
      top:50px;
      left:110px;
}div.cube211 {
      width: 9px;
      height:9px;
      position:absolute;
      top:50px;
      left:120px;
}div.cube210 {
      width: 9px;
      height:9px;
      position:absolute;
      top:50px;
      left:130px;
}div.cube129 {
      width: 9px;
      height:9px;
      position:absolute;
      top:60px;
      left:0px;
}div.cube143 {
      width: 9px;
      height:9px;
      position:absolute;
      top:60px;
      left:10px;
}div.cube12911 {
      width: 9px;
      height:9px;
      position:absolute;
      top:60px;
      left:20px;
}div.cube130 {
      width: 9px;
      height:9px;
      position:absolute;
```

```
      top:60px;
      left:30px;
}div.cube131 {
      width: 9px;
      height:9px;
      position:absolute;
      top:60px;
      left:40px;
}div.cube13111 {
      width: 9px;
      height:9px;
      position:absolute;
      top:60px;
      left:50px;
}div.cube13122 {
      width: 9px;
      height:9px;
      position:absolute;
      top:60px;
      left:60px;
}div.cube132 {
      width: 9px;
      height:9px;
      position:absolute;
      top:60px;
      left:70px;
}div.cube133 {
      width: 9px;
      height:9px;
      position:absolute;
      top:60px;
      left:80px;
}div.cube134 {
      width: 9px;
      height:9px;
      position:absolute;
      top:60px;
      left:90px;
}div.cube214 {
      width: 9px;
      height:9px;
      position:absolute;
      top:60px;
      left:100px;
}div.cube215 {
      width: 9px;
      height:9px;
      position:absolute;
      top:60px;
      left:110px;
}div.cube216 {
      width: 9px;
      height:9px;
      position:absolute;
      top:60px;
      left:120px;
}div.cube219 {
      width: 9px;
```

```
      height:9px;
      position:absolute;
      top:60px;
      left:130px;
}div.cube146 {
      width: 9px;
      height:9px;
      position:absolute;
      top:70px;
      left:0px;
}div.cube144 {
      width: 9px;
      height:9px;
      position:absolute;
      top:70px;
      left:10px;
}div.cube141 {
      width: 9px;
      height:9px;
      position:absolute;
      top:70px;
      left:20px;
}div.cube140 {
      width: 9px;
      height:9px;
      position:absolute;
      top:70px;
      left:30px;
}div.cube139 {
      width: 9px;
      height:9px;
      position:absolute;
      top:70px;
      left:40px;
}div.cube138 {
      width: 9px;
      height:9px;
      position:absolute;
      top:70px;
      left:50px;
}div.cube137 {
      width: 9px;
      height:9px;
      position:absolute;
      top:70px;
      left:60px;
}div.cube136 {
      width: 9px;
      height:9px;
      position:absolute;
      top:70px;
      left:70px;
}div.cube135 {
      width: 9px;
      height:9px;
      position:absolute;
      top:70px;
      left:80px;
```

```
}div.cube224 {
      width: 9px;
      height:9px;
      position:absolute;
      top:70px;
      left:90px;
}div.cube223 {
      width: 9px;
      height:9px;
      position:absolute;
      top:70px;
      left:100px;
}div.cube222 {
      width: 9px;
      height:9px;
      position:absolute;
      top:70px;
      left:110px;
}div.cube221 {
      width: 9px;
      height:9px;
      position:absolute;
      top:70px;
      left:120px;
}div.cube220 {
      width: 9px;
      height:9px;
      position:absolute;
      top:70px;
      left:130px;
}div.cube150 {
      width: 9px;
      height:9px;
      position:absolute;
      top:80px;
      left:0px;
}div.cube151 {
      width: 9px;
      height:9px;
      position:absolute;
      top:80px;
      left:10px;
}div.cube142 {
      width: 9px;
      height:9px;
      position:absolute;
      top:80px;
      left:20px;
}div.cube147 {
      width: 9px;
      height:9px;
      position:absolute;
      top:80px;
      left:30px;
}div.cube14711 {
      width: 9px;
      height:9px;
      position:absolute;
```

```
      top:80px;
      left:40px;
}div.cube14722 {
      width: 9px;
      height:9px;
      position:absolute;
      top:80px;
      left:50px;
}div.cube148 {
      width: 9px;
      height:9px;
      position:absolute;
      top:80px;
      left:60px;
}div.cube14811 {
      width: 9px;
      height:9px;
      position:absolute;
      top:80px;
      left:70px;
}div.cube14822 {
      width: 9px;
      height:9px;
      position:absolute;
      top:80px;
      left:80px;
}div.cube225 {
      width: 9px;
      height:9px;
      position:absolute;
      top:80px;
      left:90px;
}div.cube226 {
      width: 9px;
      height:9px;
      position:absolute;
      top:80px;
      left:100px;
}div.cube227 {
      width: 9px;
      height:9px;
      position:absolute;
      top:80px;
      left:110px;
}div.cube228 {
      width: 9px;
      height:9px;
      position:absolute;
      top:80px;
      left:120px;
}div.cube229 {
      width: 9px;
      height:9px;
      position:absolute;
      top:80px;
      left:130px;
}div.cube152 {
      width: 9px;
```

```
      height:9px;
      position:absolute;
      top:90px;
      left:0px;
}div.cube153 {
      width: 9px;
      height:9px;
      position:absolute;
      top:90px;
      left:10px;
}div.cube154 {
      width: 9px;
      height:9px;
      position:absolute;
      top:90px;
      left:20px;
}div.cube155 {
      width: 9px;
      height:9px;
      position:absolute;
      top:90px;
      left:30px;
}div.cube156 {
      width: 9px;
      height:9px;
      position:absolute;
      top:90px;
      left:40px;
}div.cube43 {
      width: 9px;
      height:9px;
      position:absolute;
      top:90px;
      left:50px;
}div.cube44 {
      width: 9px;
      height:9px;
      position:absolute;
      top:90px;
      left:60px;
}div.cube14 {
      width: 9px;
      height:9px;
      position:absolute;
      top:90px;
      left:70px;
}div.cube15 {
      width: 9px;
      height:9px;
      position:absolute;
      top:90px;
      left:80px;
}div.cube16 {
      width: 9px;
      height:9px;
      position:absolute;
      top:90px;
      left:90px;
```

```
}div.cube233 {
      width: 9px;
      height:9px;
      position:absolute;
      top:90px;
      left:100px;
}div.cube232 {
      width: 9px;
      height:9px;
      position:absolute;
      top:90px;
      left:110px;
}div.cube231 {
      width: 9px;
      height:9px;
      position:absolute;
      top:90px;
      left:120px;
}div.cube230 {
      width: 9px;
      height:9px;
      position:absolute;
      top:90px;
      left:130px;
}div.cube171 {
      width: 9px;
      height:9px;
      position:absolute;
      top:100px;
      left:0px;
}div.cube170 {
      width: 9px;
      height:9px;
      position:absolute;
      top:100px;
      left:10px;
}div.cube159 {
      width: 9px;
      height:9px;
      position:absolute;
      top:100px;
      left:20px;
}div.cube158 {
      width: 9px;
      height:9px;
      position:absolute;
      top:100px;
      left:30px;
}div.cube157 {
      width: 9px;
      height:9px;
      position:absolute;
      top:100px;
      left:40px;
}div.cube45 {
      width: 9px;
      height:9px;
      position:absolute;
```

```
      top:100px;
      left:50px;
}div.cube1008 {
      width: 9px;
      height:9px;
      position:absolute;
      top:100px;
      left:60px;
}div.cube19 {
      width: 9px;
      height:9px;
      position:absolute;
      top:100px;
      left:70px;
}div.cube18 {
      width: 9px;
      height:9px;
      position:absolute;
      top:100px;
      left:80px;
}div.cube17 {
      width: 9px;
      height:9px;
      position:absolute;
      top:100px;
      left:90px;
}div.cube234 {
      width: 9px;
      height:9px;
      position:absolute;
      top:100px;
      left:100px;
}div.cube235 {
      width: 9px;
      height:9px;
      position:absolute;
      top:100px;
      left:110px;
}div.cube1029 {
      width: 9px;
      height:9px;
      position:absolute;
      top:100px;
      left:120px;
}div.cube1030 {
      width: 9px;
      height:9px;
      position:absolute;
      top:100px;
      left:130px;
}div.cube46 {
      width: 9px;
      height:9px;
      position:absolute;
      top:110px;
      left:0px;
}div.cube4611 {
      width: 9px;
```

```
      height:9px;
      position:absolute;
      top:110px;
      left:10px;
}div.cube4622 {
      width: 9px;
      height:9px;
      position:absolute;
      top:110px;
      left:20px;
}div.cube172 {
      width: 9px;
      height:9px;
      position:absolute;
      top:110px;
      left:30px;
}div.cube173 {
      width: 9px;
      height:9px;
      position:absolute;
      top:110px;
      left:40px;
}div.cube174 {
      width: 9px;
      height:9px;
      position:absolute;
      top:110px;
      left:50px;
}div.cube20 {
      width: 9px;
      height:9px;
      position:absolute;
      top:110px;
      left:60px;
}div.cube21 {
      width: 9px;
      height:9px;
      position:absolute;
      top:110px;
      left:70px;
}div.cube22 {
      width: 9px;
      height:9px;
      position:absolute;
      top:110px;
      left:80px;
}div.cube23 {
      width: 9px;
      height:9px;
      position:absolute;
      top:110px;
      left:90px;
}div.cube1017 {
      width: 9px;
      height:9px;
      position:absolute;
      top:110px;
      left:100px;
```

```
}div.cube1019 {
      width: 9px;
      height:9px;
      position:absolute;
      top:110px;
      left:110px;
}div.cube1 {
      width: 9px;
      height:9px;
      position:absolute;
      top:110px;
      left:120px;
}div.cube2 {
      width: 9px;
      height:9px;
      position:absolute;
      top:110px;
      left:130px;
}div.cube4655 {
      width: 9px;
      height:9px;
      position:absolute;
      top:120px;
      left:0px;
}div.cube4644 {
      width: 9px;
      height:9px;
      position:absolute;
      top:120px;
      left:10px;
}div.cube4633 {
      width: 9px;
      height:9px;
      position:absolute;
      top:120px;
      left:20px;
}div.cube177 {
      width: 9px;
      height:9px;
      position:absolute;
      top:120px;
      left:30px;
}div.cube176 {
      width: 9px;
      height:9px;
      position:absolute;
      top:120px;
      left:40px;
}div.cube175 {
      width: 9px;
      height:9px;
      position:absolute;
      top:120px;
      left:50px;
}div.cube28 {
      width: 9px;
      height:9px;
      position:absolute;
```

```
      top:120px;
      left:60px;
}div.cube27 {
      width: 9px;
      height:9px;
      position:absolute;
      top:120px;
      left:70px;
}div.cube26 {
      width: 9px;
      height:9px;
      position:absolute;
      top:120px;
      left:80px;
}div.cube25 {
      width: 9px;
      height:9px;
      position:absolute;
      top:120px;
      left:90px;
}div.cube24 {
      width: 9px;
      height:9px;
      position:absolute;
      top:120px;
      left:100px;
}div.cube5 {
      width: 9px;
      height:9px;
      position:absolute;
      top:120px;
      left:110px;
}div.cube4 {
      width: 9px;
      height:9px;
      position:absolute;
      top:120px;
      left:120px;
}div.cube3 {
      width: 9px;
      height:9px;
      position:absolute;
      top:120px;
      left:130px;
}div.cube47 {
      width: 9px;
      height:9px;
      position:absolute;
      top:130px;
      left:0px;
}div.cube48 {
      width: 9px;
      height:9px;
      position:absolute;
      top:130px;
      left:10px;
}div.cube49 {
      width: 9px;
```

```
      height:9px;
      position:absolute;
      top:130px;
      left:20px;
}div.cube178 {
      width: 9px;
      height:9px;
      position:absolute;
      top:130px;
      left:30px;
}div.cube179 {
      width: 9px;
      height:9px;
      position:absolute;
      top:130px;
      left:40px;
}div.cube180 {
      width: 9px;
      height:9px;
      position:absolute;
      top:130px;
      left:50px;
}div.cube29 {
      width: 9px;
      height:9px;
      position:absolute;
      top:130px;
      left:60px;
}div.cube30 {
      width: 9px;
      height:9px;
      position:absolute;
      top:130px;
      left:70px;
}div.cube31 {
      width: 9px;
      height:9px;
      position:absolute;
      top:130px;
      left:80px;
}div.cube1007 {
      width: 9px;
      height:9px;
      position:absolute;
      top:130px;
      left:90px;
}div.cube6 {
      width: 9px;
      height:9px;
      position:absolute;
      top:130px;
      left:100px;
}div.cube7 {
      width: 9px;
      height:9px;
      position:absolute;
      top:130px;
      left:110px;
```

```
}div.cube8 {
      width: 9px;
      height:9px;
      position:absolute;
      top:130px;
      left:120px;
}div.cube9 {
      width: 9px;
      height:9px;
      position:absolute;
      top:130px;
      left:130px;
}div.cube412 {
      width: 9px;
      height:9px;
      position:absolute;
      top:140px;
      left:0px;
}div.cube411 {
      width: 9px;
      height:9px;
      position:absolute;
      top:140px;
      left:10px;
}div.cube12011 {
      width: 9px;
      height:9px;
      position:absolute;
      top:140px;
      left:20px;
}div.cube184 {
      width: 9px;
      height:9px;
      position:absolute;
      top:140px;
      left:30px;
}div.cube183 {
      width: 9px;
      height:9px;
      position:absolute;
      top:140px;
      left:40px;
}div.cube182 {
      width: 9px;
      height:9px;
      position:absolute;
      top:140px;
      left:50px;
}div.cube181 {
      width: 9px;
      height:9px;
      position:absolute;
      top:140px;
      left:60px;
}div.cube400 {
      width: 9px;
      height:9px;
      position:absolute;
```

```
      top:140px;
      left:70px;
}div.cube57 {
      width: 9px;
      height:9px;
      position:absolute;
      top:140px;
      left:80px;
}div.cube1000 {
      width: 9px;
      height:9px;
      position:absolute;
      top:140px;
      left:90px;
}div.cube13 {
      width: 9px;
      height:9px;
      position:absolute;
      top:140px;
      left:100px;
}div.cube12 {
      width: 9px;
      height:9px;
      position:absolute;
      top:140px;
      left:110px;
}div.cube11 {
      width: 9px;
      height:9px;
      position:absolute;
      top:140px;
      left:120px;
}div.cube10 {
      width: 9px;
      height:9px;
      position:absolute;
      top:140px;
      left:130px;
}div.cube413 {
      width: 9px;
      height:9px;
      position:absolute;
      top:150px;
      left:0px;
}div.cube414 {
      width: 9px;
      height:9px;
      position:absolute;
      top:150px;
      left:10px;
}div.cube415 {
      width: 9px;
      height:9px;
      position:absolute;
      top:150px;
      left:20px;
}div.cube32 {
      width: 9px;
```

```
      height:9px;
      position:absolute;
      top:150px;
      left:30px;
}div.cube33 {
      width: 9px;
      height:9px;
      position:absolute;
      top:150px;
      left:40px;
}div.cube185 {
      width: 9px;
      height:9px;
      position:absolute;
      top:150px;
      left:50px;
}div.cube1032 {
      width: 9px;
      height:9px;
      position:absolute;
      top:150px;
      left:60px;
}div.cube401 {
      width: 9px;
      height:9px;
      position:absolute;
      top:150px;
      left:70px;
}div.cube402 {
      width: 9px;
      height:9px;
      position:absolute;
      top:150px;
      left:80px;
}div.cube403 {
      width: 9px;
      height:9px;
      position:absolute;
      top:150px;
      left:90px;
}div.cube404 {
      width: 9px;
      height:9px;
      position:absolute;
      top:150px;
      left:100px;
}div.cube405 {
      width: 9px;
      height:9px;
      position:absolute;
      top:150px;
      left:110px;
}div.cube406 {
      width: 9px;
      height:9px;
      position:absolute;
      top:150px;
      left:120px;
```

```
}div.cube407 {
      width: 9px;
      height:9px;
      position:absolute;
      top:150px;
      left:130px;
}div.cube418 {
      width: 9px;
      height:9px;
      position:absolute;
      top:160px;
      left:0px;
}div.cube417 {
      width: 9px;
      height:9px;
      position:absolute;
      top:160px;
      left:10px;
}div.cube416 {
      width: 9px;
      height:9px;
      position:absolute;
      top:160px;
      left:20px;
}div.cube34 {
      width: 9px;
      height:9px;
      position:absolute;
      top:160px;
      left:30px;
}div.cube35 {
      width: 9px;
      height:9px;
      position:absolute;
      top:160px;
      left:40px;
}div.cube36 {
      width: 9px;
      height:9px;
      position:absolute;
      top:160px;
      left:50px;
}div.cube37 {
      width: 9px;
      height:9px;
      position:absolute;
      top:160px;
      left:60px;
}div.cube303 {
      width: 9px;
      height:9px;
      position:absolute;
      top:160px;
      left:70px;
}div.cube302 {
      width: 9px;
      height:9px;
      position:absolute;
```

```
      top:160px;
      left:80px;
}div.cube301 {
      width: 9px;
      height:9px;
      position:absolute;
      top:160px;
      left:90px;
}div.cube300 {
      width: 9px;
      height:9px;
      position:absolute;
      top:160px;
      left:100px;
}div.cube410 {
      width: 9px;
      height:9px;
      position:absolute;
      top:160px;
      left:110px;
}div.cube409 {
      width: 9px;
      height:9px;
      position:absolute;
      top:160px;
      left:120px;
}div.cube408 {
      width: 9px;
      height:9px;
      position:absolute;
      top:160px;
      left:130px;
}div.cube419 {
      width: 9px;
      height:9px;
      position:absolute;
      top:170px;
      left:0px;
}div.cube1009 {
      width: 9px;
      height:9px;
      position:absolute;
      top:170px;
      left:10px;
}div.cube1010 {
      width: 9px;
      height:9px;
      position:absolute;
      top:170px;
      left:20px;
}div.cube41 {
      width: 9px;
      height:9px;
      position:absolute;
      top:170px;
      left:30px;
}div.cube40 {
      width: 9px;
```

```
      height:9px;
      position:absolute;
      top:170px;
      left:40px;
}div.cube39 {
      width: 9px;
      height:9px;
      position:absolute;
      top:170px;
      left:50px;
}div.cube38 {
      width: 9px;
      height:9px;
      position:absolute;
      top:170px;
      left:60px;
}div.cube304 {
      width: 9px;
      height:9px;
      position:absolute;
      top:170px;
      left:70px;
}div.cube305 {
      width: 9px;
      height:9px;
      position:absolute;
      top:170px;
      left:80px;
}div.cube306 {
      width: 9px;
      height:9px;
      position:absolute;
      top:170px;
      left:90px;
}div.cube307 {
      width: 9px;
      height:9px;
      position:absolute;
      top:170px;
      left:100px;
}div.cube308 {
      width: 9px;
      height:9px;
      position:absolute;
      top:170px;
      left:110px;
}div.cube309 {
      width: 9px;
      height:9px;
      position:absolute;
      top:170px;
      left:120px;
}div.cube310 {
      width: 9px;
      height:9px;
      position:absolute;
      top:170px;
      left:130px;
```

```
}div.cube1033 {
      width: 9px;
      height:9px;
      position:absolute;
      top:180px;
      left:0px;
}div.cube1031 {
      width: 9px;
      height:9px;
      position:absolute;
      top:180px;
      left:10px;
}div.cube1015 {
      width: 9px;
      height:9px;
      position:absolute;
      top:180px;
      left:20px;
}div.cube42 {
      width: 9px;
      height:9px;
      position:absolute;
      top:180px;
      left:30px;
}div.cube1027 {
      width: 9px;
      height:9px;
      position:absolute;
      top:180px;
      left:40px;
}div.cube1014 {
      width: 9px;
      height:9px;
      position:absolute;
      top:180px;
      left:50px;
}div.cube1013 {
      width: 9px;
      height:9px;
      position:absolute;
      top:180px;
      left:60px;
}div.cube317 {
      width: 9px;
      height:9px;
      position:absolute;
      top:180px;
      left:70px;
}div.cube316 {
      width: 9px;
      height:9px;
      position:absolute;
      top:180px;
      left:80px;
}div.cube315 {
      width: 9px;
      height:9px;
      position:absolute;
```

```
      top:180px;
      left:90px;
}div.cube314 {
      width: 9px;
      height:9px;
      position:absolute;
      top:180px;
      left:100px;
}div.cube313 {
      width: 9px;
      height:9px;
      position:absolute;
      top:180px;
      left:110px;
}div.cube312 {
      width: 9px;
      height:9px;
      position:absolute;
      top:180px;
      left:120px;
}div.cube311 {
      width: 9px;
      height:9px;
      position:absolute;
      top:180px;
      left:130px;
}div.cube53 {
      width: 9px;
      height:9px;
      position:absolute;
      top:190px;
      left:0px;
}div.cube5511 {
      width: 9px;
      height:9px;
      position:absolute;
      top:190px;
      left:10px;
}div.cube5522 {
      width: 9px;
      height:9px;
      position:absolute;
      top:190px;
      left:20px;
}div.cube5622 {
      width: 9px;
      height:9px;
      position:absolute;
      top:190px;
      left:30px;
}div.cube5633 {
      width: 9px;
      height:9px;
      position:absolute;
      top:190px;
      left:40px;
}div.cube1001 {
      width: 9px;
```

```
      height:9px;
      position:absolute;
      top:190px;
      left:50px;
}div.cube1002 {
      width: 9px;
      height:9px;
      position:absolute;
      top:190px;
      left:60px;
}div.cube318 {
      width: 9px;
      height:9px;
      position:absolute;
      top:190px;
      left:70px;
}div.cube319{
      width: 9px;
      height:9px;
      position:absolute;
      top:190px;
      left:80px;
}div.cube320 {
      width: 9px;
      height:9px;
      position:absolute;
      top:190px;
      left:90px;
}div.cube321 {
      width: 9px;
      height:9px;
      position:absolute;
      top:190px;
      left:100px;
}div.cube322 {
      width: 9px;
      height:9px;
      position:absolute;
      top:190px;
      left:110px;
}div.cube323 {
      width: 9px;
      height:9px;
      position:absolute;
      top:190px;
      left:120px;
}div.cube1021 {
      width: 9px;
      height:9px;
      position:absolute;
      top:190px;
      left:130px;
}div.cube54 {
      width: 9px;
      height:9px;
      position:absolute;
      top:200px;
      left:0px;
```

```
}div.cube55 {
      width: 9px;
      height:9px;
      position:absolute;
      top:200px;
      left:10px;
}div.cube56 {
      width: 9px;
      height:9px;
      position:absolute;
      top:200px;
      left:20px;
}div.cube5611 {
      top:200px;
      left:30px;
      width: 9px;
      height: 9px;
      position: absolute;
}div.cube5644 {
      top:200px;
      left:40px;
      width: 9px;
      height: 9px;
      position: absolute;
}div.cube58 {
      top:200px;
      left:50px;
      width: 9px;
      height: 9px;
      position: absolute;
}div.cube1012 {
      top:200px;
      left:60px;
      width: 9px;
      height: 9px;
      position: absolute;
}div.cube190 {
      top:200px;
      left:70px;
      width: 9px;
      height: 9px;
      position: absolute;
}div.cube1028 {
      top:200px;
      left:80px;
      width: 9px;
      height: 9px;
      position: absolute;
}div.cube1026 {
      top:200px;
      left:90px;
      width: 9px;
      height: 9px;
      position: absolute;
}div.cube1025 {
      top:200px;
      left:100px;
      width: 9px;
```

```
      height: 9px;
      position: absolute;
}div.cube1024 {
      top:200px;
      left:110px;
      width: 9px;
      height: 9px;
      position: absolute;
}div.cube1023 {
      top:200px;
      left:120px;
      width: 9px;
      height: 9px;
      position: absolute;
}div.cube1022 {
      top:200px;
      left:130px;
      width: 9px;
      height: 9px;
      position: absolute;
}
```

```
/*
 * jQuery JavaScript Library v1.3.1
 * http://jquery.com/
 *
 * Copyright (c) 2009 John Resig
 * Dual licensed under the MIT and GPL licenses.
 * http://docs.jquery.com/License
 *
 * Date: 2009-01-21 20:42:16 -0500 (Wed, 21 Jan 2009)
 * Revision: 6158
 */
(function(){var l=this,g,y=l.jQuery,p=l.$,o=l.jQuery=l.$=function(E,F){return
new o.fn.init(E,F)},D=/^[^<]*(<(.|\s)+>)[^>]*$|^#([\w-
]+)$/,f=/^.[^:#\[\.,]*$/;o.fn=o.prototype={init:function(E,H){E=E||document;i
f(E.nodeType){this[0]=E;this.length=1;this.context=E;return this}if(typeof
E==="string"){var
G=D.exec(E);if(G&&(G[1]||!H)){if(G[1]){E=o.clean([G[1]],H)}else{var
I=document.getElementById(G[3]);if(I&&I.id!=G[3]){return o().find(E)}var
F=o(I||[]);F.context=document;F.selector=E;return F}}else{return
o(H).find(E)}}else{if(o.isFunction(E)){return
o(document).ready(E)}if(E.selector&&E.context){this.selector=E.selector;this
.context=E.context}return
this.setArray(o.makeArray(E))},selector:"",jquery:"1.3.1",size:function(){ret
urn this.length},get:function(E){return
E===g?o.makeArray(this):this[E]},pushStack:function(F,H,E){var
G=o(F);G.prevObject=this;G.context=this.context;if(H==="find"){G.selector=thi
s.selector+(this.selector?
":"")+E}else{if(H){G.selector=this.selector+"."+H+"("+E+")"}}return
G},setArray:function(E){this.length=0;Array.prototype.push.apply(this,E);retu
rn this},each:function(F,E){return o.each(this,F,E)},index:function(E){return
o.inArray(E&&E.jquery?E[0]:E,this)},attr:function(F,H,G){var E=F;if(typeof
F==="string"){if(H===g){return
this[0]&&o[G||"attr"](this[0],F)}else{E={};E[F]=H}}return
this.each(function(I){for(F in
E){o.attr(G?this.style:this,F,o.prop(this,E[F],G,I,F))}})},css:function(E,F){
if((E=="width"||E=="height")&&parseFloat(F)<0){F=g}return
this.attr(E,F,"curCSS")},text:function(F){if(typeof
F!=="object"&&F!=null){return
this.empty().append((this[0]&&this[0].ownerDocument||document).createTextNode
(F))}var
E="";o.each(F||this,function(){o.each(this.childNodes,function(){if(this.node
Type!=8){E+=this.nodeType!=1?this.nodeValue:o.fn.text([this])}})});return
E},wrapAll:function(E){if(this[0]){var
F=o(E,this[0].ownerDocument).clone();if(this[0].parentNode){F.insertBefore(th
is[0])}F.map(function(){var G=this;while(G.firstChild){G=G.firstChild}return
G}).append(this)}return this},wrapInner:function(E){return
this.each(function(){o(this).contents().wrapAll(E)})},wrap:function(E){return
this.each(function(){o(this).wrapAll(E)})},append:function(){return
this.domManip(arguments,true,function(E){if(this.nodeType==1){this.appendChil
d(E)}})},prepend:function(){return
this.domManip(arguments,true,function(E){if(this.nodeType==1){this.insertBefo
re(E,this.firstChild)}})},before:function(){return
this.domManip(arguments,false,function(E){this.parentNode.insertBefore(E,this
)})},after:function(){return
this.domManip(arguments,false,function(E){this.parentNode.insertBefore(E,this
.nextSibling)})},end:function(){return
this.prevObject||o([])},push:[].push,find:function(E){if(this.length===1&&!/,
/.test(E)){var
```

```
G=this.pushStack([],"find",E);G.length=0;o.find(E,this[0],G);return
G}else{var F=o.map(this,function(H){return o.find(E,H)});return
this.pushStack(/[^+>]
[^+>]/.test(E)?o.unique(F):F,"find",E)}},clone:function(F){var
E=this.map(function(){if(!o.support.noCloneEvent&&!o.isXMLDoc(this)){var
I=this.cloneNode(true),H=document.createElement("div");H.appendChild(I);retur
n o.clean([H.innerHTML])[0]}else{return this.cloneNode(true)}});var
G=E.find("*").andSelf().each(function(){if(this[h]!==g){this[h]=null}});if(F=
==true){this.find("*").andSelf().each(function(I){if(this.nodeType==3){return
}var H=o.data(this,"events");for(var K in H){for(var J in
H[K]){o.event.add(G[I],K,H[K][J],H[K][J].data)}}})}return
E},filter:function(E){return
this.pushStack(o.isFunction(E)&&o.grep(this,function(G,F){return
E.call(G,F)})||o.multiFilter(E,o.grep(this,function(F){return
F.nodeType===1})),"filter",E)},closest:function(E){var
F=o.expr.match.POS.test(E)?o(E):null;return this.map(function(){var
G=this;while(G&&G.ownerDocument){if(F?F.index(G)>-1:o(G).is(E)){return
G}G=G.parentNode}})},not:function(E){if(typeof
E==="string"){if(f.test(E)){return
this.pushStack(o.multiFilter(E,this,true),"not",E)}else{E=o.multiFilter(E,thi
s)}}var F=E.length&&E[E.length-1]!==g&&!E.nodeType;return
this.filter(function(){return
F?o.inArray(this,E)<0:this!=E})},add:function(E){return
this.pushStack(o.unique(o.merge(this.get(),typeof
E==="string"?o(E):o.makeArray(E))))},is:function(E){return
!!E&&o.multiFilter(E,this).length>0},hasClass:function(E){return
!!E&&this.is("."+E)},val:function(K){if(K===g){var
E=this[0];if(E){if(o.nodeName(E,"option")){return(E.attributes.value||{}).spe
cified?E.value:E.text}if(o.nodeName(E,"select")){var
I=E.selectedIndex,L=[],M=E.options,H=E.type=="select-one";if(I<0){return
null}for(var F=H?I:0,J=H?I+1:M.length;F<J;F++){var
G=M[F];if(G.selected){K=o(G).val();if(H){return K}L.push(K)}}return
L}return(E.value||"").replace(/\r/g,"")}return g}if(typeof
K==="number"){K+=""}return
this.each(function(){if(this.nodeType!=1){return}if(o.isArray(K)&&/radio|chec
kbox/.test(this.type)){this.checked=(o.inArray(this.value,K)>=0||o.inArray(th
is.name,K)>=0)}else{if(o.nodeName(this,"select")){var
N=o.makeArray(K);o("option",this).each(function(){this.selected=(o.inArray(th
is.value,N)>=0||o.inArray(this.text,N)>=0)});if(!N.length){this.selectedIndex
=-1}}else{this.value=K}}})},html:function(E){return
E===g?(this[0]?this[0].innerHTML:null):this.empty().append(E)},replaceWith:fu
nction(E){return this.after(E).remove()},eq:function(E){return
this.slice(E,+E+1)},slice:function(){return
this.pushStack(Array.prototype.slice.apply(this,arguments),"slice",Array.prot
otype.slice.call(arguments).join(","))},map:function(E){return
this.pushStack(o.map(this,function(G,F){return
E.call(G,F,G)}))},andSelf:function(){return
this.add(this.prevObject)},domManip:function(K,N,M){if(this[0]){var
J=(this[0].ownerDocument||this[0]).createDocumentFragment(),G=o.clean(K,(this
[0].ownerDocument||this[0]),J),I=J.firstChild,E=this.length>1?J.cloneNode(tru
e):J;if(I){for(var
H=0,F=this.length;H<F;H++){M.call(L(this[H],I),H>0?E.cloneNode(true):J)}}if(G
){o.each(G,z)}}return this;function L(O,P){return
N&&o.nodeName(O,"table")&&o.nodeName(P,"tr")?(O.getElementsByTagName("tbody")
[0]||O.appendChild(O.ownerDocument.createElement("tbody"))):O}};o.fn.init.pr
ototype=o.fn;function
z(E,F){if(F.src){o.ajax({url:F.src,async:false,dataType:"script"})}else{o.glo
balEval(F.text||F.textContent||F.innerHTML||"")}if(F.parentNode){F.parentNode
```

```
.removeChild(F)}}function e(){return +new
Date}o.extend=o.fn.extend=function(){var
J=arguments[0]||{},H=1,I=arguments.length,E=false,G;if(typeof
J==="boolean"){E=J;J=arguments[1]||{};H=2}if(typeof
J!=="object"&&!o.isFunction(J)){J={}}if(I==H){J=this;--
H}for(;H<I;H++){if((G=arguments[H])!=null){for(var F in G){var
K=J[F],L=G[F];if(J===L){continue}if(E&&L&&typeof
L==="object"&&!L.nodeType){J[F]=o.extend(E,K||(L.length!=null?[]:{}),L)}else{
if(L!==g){J[F]=L}}}}}return J};var b=/z-?index|font-
?weight|opacity|zoom|line-
?height/i,q=document.defaultView||{},s=Object.prototype.toString;o.extend({no
Conflict:function(E){l.$=p;if(E){l.jQuery=y}return
o},isFunction:function(E){return s.call(E)==="[object
Function]"},isArray:function(E){return s.call(E)==="[object
Array]"},isXMLDoc:function(E){return
E.nodeType===9&&E.documentElement.nodeName!=="HTML"||!!E.ownerDocument&&o.isX
MLDoc(E.ownerDocument)},globalEval:function(G){G=o.trim(G);if(G){var
F=document.getElementsByTagName("head")[0]||document.documentElement,E=docume
nt.createElement("script");E.type="text/javascript";if(o.support.scriptEval){
E.appendChild(document.createTextNode(G))}else{E.text=G}F.insertBefore(E,F.fi
rstChild);F.removeChild(E)}},nodeName:function(F,E){return
F.nodeName&&F.nodeName.toUpperCase()==E.toUpperCase()},each:function(G,K,F){v
ar E,H=0,I=G.length;if(F){if(I===g){for(E in
G){if(K.apply(G[E],F)===false){break}}}else{for(;H<I;){if(K.apply(G[H++],F)==
=false){break}}}}else{if(I===g){for(E in
G){if(K.call(G[E],E,G[E])===false){break}}}else{for(var
J=G[0];H<I&&K.call(J,H,J)!==false;J=G[++H]){}}}return
G},prop:function(H,I,G,F,E){if(o.isFunction(I)){I=I.call(H,F)}return typeof
I==="number"&&G=="curCSS"&&!b.test(E)?I+"px":I},className:{add:function(E,F){
o.each((F||"").split(/\s+/),function(G,H){if(E.nodeType==1&&!o.className.has(
E.className,H)){E.className+=(E.className?"
":"")+H}})},remove:function(E,F){if(E.nodeType==1){E.className=F!==g?o.grep(E
.className.split(/\s+/),function(G){return !o.className.has(F,G)}).join("
"):""}},has:function(F,E){return
F&&o.inArray(E,(F.className||F).toString().split(/\s+/))>-
1}},swap:function(H,G,I){var E={};for(var F in
G){E[F]=H.style[F];H.style[F]=G[F]}I.call(H);for(var F in
G){H.style[F]=E[F]}},css:function(G,E,I){if(E=="width"||E=="height"){var
K,F={position:"absolute",visibility:"hidden",display:"block"},J=E=="width"?["
Left","Right"]:["Top","Bottom"];function
H(){K=E=="width"?G.offsetWidth:G.offsetHeight;var
M=0,L=0;o.each(J,function(){M+=parseFloat(o.curCSS(G,"padding"+this,true))||0
;L+=parseFloat(o.curCSS(G,"border"+this+"Width",true))||0});K-
=Math.round(M+L)}if(o(G).is(":visible")){H()}else{o.swap(G,F,H)}return
Math.max(0,K)}return o.curCSS(G,E,I)},curCSS:function(I,F,G){var
L,E=I.style;if(F=="opacity"&&!o.support.opacity){L=o.attr(E,"opacity");return
L==""?"1":L}if(F.match(/float/i)){F=w}if(!G&&E&&E[F]){L=E[F]}else{if(q.getCom
putedStyle){if(F.match(/float/i)){F="float"}F=F.replace(/([A-Z])/g,"-
$1").toLowerCase();var
M=q.getComputedStyle(I,null);if(M){L=M.getPropertyValue(F)}if(F=="opacity"&&L
==""){L="1"}}else{if(I.currentStyle){var J=F.replace(/\-
(\w)/g,function(N,O){return
O.toUpperCase()});L=I.currentStyle[F]||I.currentStyle[J];if(!/^\d+(px)?$/i.te
st(L)&&/^\d/.test(L)){var
H=E.left,K=I.runtimeStyle.left;I.runtimeStyle.left=I.currentStyle.left;E.left
=L||0;L=E.pixelLeft+"px";E.left=H;I.runtimeStyle.left=K}}}}return
L},clean:function(F,K,I){K=K||document;if(typeof
K.createElement==="undefined"){K=K.ownerDocument||K[0]&&K[0].ownerDocument||d
```

```
ocument}if(!I&&F.length===1&&typeof F[0]==="string"){var
H=/^<(\w+)\s*\/?>$/.exec(F[0]);if(H){return[K.createElement(H[1])]}}var
G=[],E=[],L=K.createElement("div");o.each(F,function(P,R){if(typeof
R==="number"){R+=""}if(!R){return}if(typeof
R==="string"){R=R.replace(/(<(\w+)[^>]*?)\/>/g,function(T,U,S){return
S.match(/^(abbr|br|col|img|input|link|meta|param|hr|area|embed)$/i)?T:U+"></"
+S+">"});var O=o.trim(R).toLowerCase();var Q=!O.indexOf("<opt")&&[1,"<select
multiple='multiple'>","</select>"]||!O.indexOf("<leg")&&[1,"<fieldset>","</fi
eldset>"]||O.match(/^<(thead|tbody|tfoot|colg|cap)/)&&[1,"<table>","</table>"
]||!O.indexOf("<tr")&&[2,"<table><tbody>","</tbody></table>"]||(!O.indexOf("<
td")||!O.indexOf("<th"))&&[3,"<table><tbody><tr>","</tr></tbody></table>"]||!
O.indexOf("<col")&&[2,"<table><tbody></tbody><colgroup>","</colgroup></table>
"]||!o.support.htmlSerialize&&[1,"div<div>","</div>"]||[0,"",""];L.innerHTML=
Q[1]+R+Q[2];while(Q[0]--){L=L.lastChild}if(!o.support.tbody){var
N=!O.indexOf("<table")&&O.indexOf("<tbody")<0?L.firstChild&&L.firstChild.chil
dNodes:Q[1]=="<table>"&&O.indexOf("<tbody")<0?L.childNodes:[];for(var
M=N.length-1;M>=0;--
M){if(o.nodeName(N[M],"tbody")&&!N[M].childNodes.length){N[M].parentNode.remo
veChild(N[M])}}}if(!o.support.leadingWhitespace&&/^\s/.test(R)){L.insertBefor
e(K.createTextNode(R.match(/^\s*/)[0]),L.firstChild)}R=o.makeArray(L.childNod
es)}if(R.nodeType){G.push(R)}else{G=o.merge(G,R)}});if(I){for(var
J=0;G[J];J++){if(o.nodeName(G[J],"script")&&(!G[J].type||G[J].type.toLowerCas
e()==="text/javascript")){E.push(G[J].parentNode?G[J].parentNode.removeChild(
G[J]):G[J])}else{if(G[J].nodeType===1){G.splice.apply(G,[J+1,0].concat(o.make
Array(G[J].getElementsByTagName("script"))))}I.appendChild(G[J])}}return
E}return G},attr:function(J,G,K){if(!J||J.nodeType==3||J.nodeType==8){return
g}var H=!o.isXMLDoc(J),L=K!==g;G=H&&o.props[G]||G;if(J.tagName){var
F=/href|src|style/.test(G);if(G=="selected"&&J.parentNode){J.parentNode.selec
tedIndex}if(G in
J&&H&&!F){if(L){if(G=="type"&&o.nodeName(J,"input")&&J.parentNode){throw"type
property can't be
changed"}J[G]=K}if(o.nodeName(J,"form")&&J.getAttributeNode(G)){return
J.getAttributeNode(G).nodeValue}if(G=="tabIndex"){var
I=J.getAttributeNode("tabIndex");return
I&&I.specified?I.value:J.nodeName.match(/(button|input|object|select|textarea
)/i)?0:J.nodeName.match(/^(a|area)$/i)&&J.href?0:g}return
J[G]}if(!o.support.style&&H&&G=="style"){return
o.attr(J.style,"cssText",K)}if(L){J.setAttribute(G,""+K)}var
E=!o.support.hrefNormalized&&H&&F?J.getAttribute(G,2):J.getAttribute(G);retur
n
E===null?g:E}if(!o.support.opacity&&G=="opacity"){if(L){J.zoom=1;J.filter=(J.
filter||"").replace(/alpha\([^)]*\)/,"")+(parseInt(K)+""=="NaN"?"":"alpha(opa
city="+K*100+")")}return
J.filter&&J.filter.indexOf("opacity=")>=0?(parseFloat(J.filter.match(/opacity
=([^)]*)/)[1])/100)+"":""}G=G.replace(/-([a-z])/ig,function(M,N){return
N.toUpperCase()});if(L){J[G]=K}return
J[G]},trim:function(E){return(E||"").replace(/^\s+|\s+$/g,"")},makeArray:func
tion(G){var E=[];if(G!=null){var F=G.length;if(F==null||typeof
G==="string"||o.isFunction(G)||G.setInterval){E[0]=G}else{while(F){E[--
F]=G[F]}}}return E},inArray:function(G,H){for(var
E=0,F=H.length;E<F;E++){if(H[E]===G){return E}}return -
1},merge:function(H,E){var
F=0,G,I=H.length;if(!o.support.getAll){while((G=E[F++])!=null){if(G.nodeType!
=8){H[I++]=G}}}else{while((G=E[F++])!=null){H[I++]=G}}return
H},unique:function(K){var F=[],E={};try{for(var G=0,H=K.length;G<H;G++){var
J=o.data(K[G]);if(!E[J]){E[J]=true;F.push(K[G])}}}catch(I){F=K}return
F},grep:function(F,J,E){var G=[];for(var
H=0,I=F.length;H<I;H++){if(!E!=!J(F[H],H)){G.push(F[H])}}return
```

```
G},map:function(E,J){var F=[];for(var G=0,H=E.length;G<H;G++){var
I=J(E[G],G);if(I!=null){F[F.length]=I}}return F.concat.apply([],F)}});var
C=navigator.userAgent.toLowerCase();o.browser={version:(C.match(/.+(?:rv|it|r
a|ie)[\/:
]([\d.]+)/)||[0,"0"])[1],safari:/webkit/.test(C),opera:/opera/.test(C),msie:/
msie/.test(C)&&!/opera/.test(C),mozilla:/mozilla/.test(C)&&!/(compatible|webk
it)/.test(C)};o.each({parent:function(E){return
E.parentNode},parents:function(E){return
o.dir(E,"parentNode")},next:function(E){return
o.nth(E,2,"nextSibling")},prev:function(E){return
o.nth(E,2,"previousSibling")},nextAll:function(E){return
o.dir(E,"nextSibling")},prevAll:function(E){return
o.dir(E,"previousSibling")},siblings:function(E){return
o.sibling(E.parentNode.firstChild,E)},children:function(E){return
o.sibling(E.firstChild)},contents:function(E){return
o.nodeName(E,"iframe")?E.contentDocument||E.contentWindow.document:o.makeArra
y(E.childNodes)}},function(E,F){o.fn[E]=function(G){var
H=o.map(this,F);if(G&&typeof G=="string"){H=o.multiFilter(G,H)}return
this.pushStack(o.unique(H),E,G)}});o.each({appendTo:"append",prependTo:"prepe
nd",insertBefore:"before",insertAfter:"after",replaceAll:"replaceWith"},funct
ion(E,F){o.fn[E]=function(){var G=arguments;return
this.each(function(){for(var
H=0,I=G.length;H<I;H++){o(G[H])[F](this)}})}});o.each({removeAttr:function(E)
{o.attr(this,E,"");if(this.nodeType==1){this.removeAttribute(E)}},addClass:fu
nction(E){o.className.add(this,E)},removeClass:function(E){o.className.remove
(this,E)},toggleClass:function(F,E){if(typeof
E!=="boolean"){E=!o.className.has(this,F)}o.className[E?"add":"remove"](this,
F)},remove:function(E){if(!E||o.filter(E,[this]).length){o("*",this).add([thi
s]).each(function(){o.event.remove(this);o.removeData(this)});if(this.parentN
ode){this.parentNode.removeChild(this)}}},empty:function(){o(">*",this).remov
e();while(this.firstChild){this.removeChild(this.firstChild)}}},function(E,F)
{o.fn[E]=function(){return this.each(F,arguments)}});function j(E,F){return
E[0]&&parseInt(o.curCSS(E[0],F,true),10)||0}var
h="jQuery"+e(),v=0,A={};o.extend({cache:{},data:function(F,E,G){F=F==l?A:F;va
r
H=F[h];if(!H){H=F[h]=++v}if(E&&!o.cache[H]){o.cache[H]={}}if(G!==g){o.cache[H
][E]=G}return E?o.cache[H][E]:H},removeData:function(F,E){F=F==l?A:F;var
H=F[h];if(E){if(o.cache[H]){delete o.cache[H][E];E="";for(E in
o.cache[H]){break}if(!E){o.removeData(F)}}}else{try{delete
F[h]}catch(G){if(F.removeAttribute){F.removeAttribute(h)}}delete
o.cache[H]}},queue:function(F,E,H){if(F){E=(E||"fx")+"queue";var
G=o.data(F,E);if(!G||o.isArray(H)){G=o.data(F,E,o.makeArray(H))}else{if(H){G.
push(H)}}}return G},dequeue:function(H,G){var
E=o.queue(H,G),F=E.shift();if(!G||G==="fx"){F=E[0]}if(F!==g){F.call(H)}}});o.
fn.extend({data:function(E,G){var
H=E.split(".");H[1]=H[1]?"."+H[1]:"";if(G===g){var
F=this.triggerHandler("getData"+H[1]+"!",[H[0]]);if(F===g&&this.length){F=o.d
ata(this[0],E)}return F===g&&H[1]?this.data(H[0]):F}else{return
this.trigger("setData"+H[1]+"!",[H[0],G]).each(function(){o.data(this,E,G)})}
},removeData:function(E){return
this.each(function(){o.removeData(this,E)})},queue:function(E,F){if(typeof
E!=="string"){F=E;E="fx"}if(F===g){return o.queue(this[0],E)}return
this.each(function(){var
G=o.queue(this,E,F);if(E=="fx"&&G.length==1){G[0].call(this)}})},dequeue:func
tion(E){return this.each(function(){o.dequeue(this,E)})}});
/*
 * Sizzle CSS Selector Engine - v0.9.3
 *  Copyright 2009, The Dojo Foundation
```

```
 *  Released under the MIT, BSD, and GPL Licenses.
 *  More information: http://sizzlejs.com/
 */
(function(){var
Q=/((?:\((?:\([^()]+\)|[^()]+)+\)|\[(?:\[[^\]]*\]|['"][^'"]+['"]|[^\]'"]+)+
\]|\\.|[^ >+~,(\[]+)+|[>+~])(\s*,\s*)?/g,K=0,G=Object.prototype.toString;var
F=function(X,T,aa,ab){aa=aa||[];T=T||document;if(T.nodeType!==1&&T.nodeType!=
=9){return[]}if(!X||typeof X!=="string"){return aa}var
Y=[],V,ae,ah,S,ac,U,W=true;Q.lastIndex=0;while((V=Q.exec(X))!==null){Y.push(V
[1]);if(V[2]){U=RegExp.rightContext;break}}if(Y.length>1&&L.exec(X)){if(Y.len
gth===2&&H.relative[Y[0]]){ae=I(Y[0]+Y[1],T)}else{ae=H.relative[Y[0]]?[T]:F(Y
.shift(),T);while(Y.length){X=Y.shift();if(H.relative[X]){X+=Y.shift()}ae=I(X
,ae)}}}else{var
ad=ab?{expr:Y.pop(),set:E(ab)}:F.find(Y.pop(),Y.length===1&&T.parentNode?T.pa
rentNode:T,P(T));ae=F.filter(ad.expr,ad.set);if(Y.length>0){ah=E(ae)}else{W=f
alse}while(Y.length){var
ag=Y.pop(),af=ag;if(!H.relative[ag]){ag=""}else{af=Y.pop()}if(af==null){af=T}
H.relative[ag](ah,af,P(T))}}if(!ah){ah=ae}if(!ah){throw"Syntax error,
unrecognized expression: "+(ag||X)}if(G.call(ah)==="[object
Array]"){if(!W){aa.push.apply(aa,ah)}else{if(T.nodeType===1){for(var
Z=0;ah[Z]!=null;Z++){if(ah[Z]&&(ah[Z]===true||ah[Z].nodeType===1&&J(T,ah[Z]))
){aa.push(ae[Z])}}}else{for(var
Z=0;ah[Z]!=null;Z++){if(ah[Z]&&ah[Z].nodeType===1){aa.push(ae[Z])}}}}}else{E(
ah,aa)}if(U){F(U,T,aa,ab)}return aa};F.matches=function(S,T){return
F(S,null,null,T)};F.find=function(Z,S,aa){var Y,W;if(!Z){return[]}for(var
V=0,U=H.order.length;V<U;V++){var
X=H.order[V],W;if((W=H.match[X].exec(Z))){var
T=RegExp.leftContext;if(T.substr(T.length-
1)!=="\\"){W[1]=(W[1]||"").replace(/\\/g,"");Y=H.find[X](W,S,aa);if(Y!=null){
Z=Z.replace(H.match[X],"");break}}}}if(!Y){Y=S.getElementsByTagName("*")}retu
rn{set:Y,expr:Z}};F.filter=function(ab,aa,ae,V){var
U=ab,ag=[],Y=aa,X,S;while(ab&&aa.length){for(var Z in
H.filter){if((X=H.match[Z].exec(ab))!=null){var
T=H.filter[Z],af,ad;S=false;if(Y==ag){ag=[]}if(H.preFilter[Z]){X=H.preFilter[
Z](X,Y,ae,ag,V);if(!X){S=af=true}else{if(X===true){continue}}}if(X){for(var
W=0;(ad=Y[W])!=null;W++){if(ad){af=T(ad,X,W,Y);var
ac=V^!!af;if(ae&&af!=null){if(ac){S=true}else{Y[W]=false}}else{if(ac){ag.push
(ad);S=true}}}}}if(af!==g){if(!ae){Y=ag}ab=ab.replace(H.match[Z],"");if(!S){r
eturn[]}break}}ab=ab.replace(/\s*,\s*/,"");if(ab==U){if(S==null){throw"Synta
x error, unrecognized expression: "+ab}else{break}}U=ab}return Y};var
H=F.selectors={order:["ID","NAME","TAG"],match:{ID:/#((?:[\w\u00c0-\uFFFF_-
]|\\.)+)/,CLASS:/\.((?:[\w\u00c0-\uFFFF_-
]|\\.)+)/,NAME:/\[name=['"]*((?:[\w\u00c0-\uFFFF_-
]|\\.)+)['"]*\]/,ATTR:/\[\s*((?:[\w\u00c0-\uFFFF_-
]|\\.)+)\s*(?:(\S?=)\s*(['"]*)(.*?)\3|)\s*\]/,TAG:/^((?:[\w\u00c0-\uFFFF\*_-
]|\\.)+)/,CHILD:/:(only|nth|last|first)-child(?:\((even|odd|[\dn+-
]*)\))?/,POS:/:(nth|eq|gt|lt|first|last|even|odd)(?:\((\d*)\))?(?=[^-
]|$)/,PSEUDO:/:((?:[\w\u00c0-\uFFFF_-
]|\\.)+)(?:\((['"]*)((?:\([^\)]+\)|[^\2\(\)]*)+)\2\))?/},attrMap:{"class":"cl
assName","for":"htmlFor"},attrHandle:{href:function(S){return
S.getAttribute("href")}},relative:{"+":function(W,T){for(var
U=0,S=W.length;U<S;U++){var V=W[U];if(V){var
X=V.previousSibling;while(X&&X.nodeType!==1){X=X.previousSibling}W[U]=typeof
T==="string"?X||false:X===T}}if(typeof
T==="string"){F.filter(T,W,true)}},">":function(X,T,Y){if(typeof
T==="string"&&!/\W/.test(T)){T=Y?T:T.toUpperCase();for(var
U=0,S=X.length;U<S;U++){var W=X[U];if(W){var
V=W.parentNode;X[U]=V.nodeName===T?V:false}}}else{for(var
```

```
U=0,S=X.length;U<S;U++){var W=X[U];if(W){X[U]=typeof
T==="string"?W.parentNode:W.parentNode===T}}if(typeof
T==="string"){F.filter(T,X,true)}}},"":function(V,T,X){var
U="done"+(K++),S=R;if(!T.match(/\W/)){var
W=T=X?T:T.toUpperCase();S=O}S("parentNode",T,U,V,W,X)},"~":function(V,T,X){va
r U="done"+(K++),S=R;if(typeof T==="string"&&!T.match(/\W/)){var
W=T=X?T:T.toUpperCase();S=O}S("previousSibling",T,U,V,W,X)}},find:{ID:functio
n(T,U,V){if(typeof U.getElementById!=="undefined"&&!V){var
S=U.getElementById(T[1]);return S?[S]:[]}},NAME:function(S,T,U){if(typeof
T.getElementsByName!=="undefined"&&!U){return
T.getElementsByName(S[1])}},TAG:function(S,T){return
T.getElementsByTagName(S[1])}},preFilter:{CLASS:function(V,T,U,S,Y){V="
"+V[1].replace(/\\/g,"")+" ";var X;for(var
W=0;(X=T[W])!=null;W++){if(X){if(Y^(" "+X.className+"
").indexOf(V)>=0){if(!U){S.push(X)}}else{if(U){T[W]=false}}}}return
false},ID:function(S){return
S[1].replace(/\\/g,"")},TAG:function(T,S){for(var
U=0;S[U]===false;U++){}return
S[U]&&P(S[U])?T[1]:T[1].toUpperCase()},CHILD:function(S){if(S[1]=="nth"){var
T=/(-?)(\d*)n((?:\+|-
)?\d*)/.exec(S[2]=="even"&&"2n"||S[2]=="odd"&&"2n+1"||!/\D/.test(S[2])&&"0n+"
+S[2]||S[2]);S[2]=(T[1]+(T[2]||1))-0;S[3]=T[3]-0}S[0]="done"+(K++);return
S},ATTR:function(T){var
S=T[1].replace(/\\/g,"");if(H.attrMap[S]){T[1]=H.attrMap[S]}if(T[2]==="~="){T
[4]=" "+T[4]+" "}return
T},PSEUDO:function(W,T,U,S,X){if(W[1]==="not"){if(W[3].match(Q).length>1){W[3
]=F(W[3],null,null,T)}else{var
V=F.filter(W[3],T,U,true^X);if(!U){S.push.apply(S,V)}return
false}}else{if(H.match.POS.test(W[0])){return true}}return
W},POS:function(S){S.unshift(true);return
S}},filters:{enabled:function(S){return
S.disabled===false&&S.type!=="hidden"},disabled:function(S){return
S.disabled===true},checked:function(S){return
S.checked===true},selected:function(S){S.parentNode.selectedIndex;return
S.selected===true},parent:function(S){return
!!S.firstChild},empty:function(S){return
!S.firstChild},has:function(U,T,S){return
!!F(S[3],U).length},header:function(S){return/h\d/i.test(S.nodeName)},text:fu
nction(S){return"text"===S.type},radio:function(S){return"radio"===S.type},ch
eckbox:function(S){return"checkbox"===S.type},file:function(S){return"file"==
=S.type},password:function(S){return"password"===S.type},submit:function(S){r
eturn"submit"===S.type},image:function(S){return"image"===S.type},reset:funct
ion(S){return"reset"===S.type},button:function(S){return"button"===S.type||S.
nodeName.toUpperCase()==="BUTTON"},input:function(S){return/input|select|text
area|button/i.test(S.nodeName)}},setFilters:{first:function(T,S){return
S===0},last:function(U,T,S,V){return T===V.length-
1},even:function(T,S){return S%2===0},odd:function(T,S){return
S%2===1},lt:function(U,T,S){return T<S[3]-0},gt:function(U,T,S){return
T>S[3]-0},nth:function(U,T,S){return S[3]-0==T},eq:function(U,T,S){return
S[3]-0==T}},filter:{CHILD:function(S,V){var Y=V[1],Z=S.parentNode;var
X=V[0];if(Z&&(!Z[X]||!S.nodeIndex)){var W=1;for(var
T=Z.firstChild;T;T=T.nextSibling){if(T.nodeType==1){T.nodeIndex=W++}}Z[X]=W-
1}if(Y=="first"){return S.nodeIndex==1}else{if(Y=="last"){return
S.nodeIndex==Z[X]}else{if(Y=="only"){return Z[X]==1}else{if(Y=="nth"){var
ab=false,U=V[2],aa=V[3];if(U==1&&aa==0){return
true}if(U==0){if(S.nodeIndex==aa){ab=true}}else{if((S.nodeIndex-
aa)%U==0&&(S.nodeIndex-aa)/U>=0){ab=true}}return
ab}}}}},PSEUDO:function(Y,U,V,Z){var T=U[1],W=H.filters[T];if(W){return
```

```
W(Y,V,U,Z)}else{if(T==="contains"){return(Y.textContent||Y.innerText||"").ind
exOf(U[3])>=0}else{if(T==="not"){var X=U[3];for(var
V=0,S=X.length;V<S;V++){if(X[V]===Y){return false}}return
true}}}},ID:function(T,S){return
T.nodeType===1&&T.getAttribute("id")===S},TAG:function(T,S){return(S==="*"&&T
.nodeType===1)||T.nodeName===S},CLASS:function(T,S){return
S.test(T.className)},ATTR:function(W,U){var
S=H.attrHandle[U[1]]?H.attrHandle[U[1]](W):W[U[1]]||W.getAttribute(U[1]),X=S+
"",V=U[2],T=U[4];return
S==null?V==="!=":V==="="?X===T:V==="*="?X.indexOf(T)>=0:V==="~="?(" "+X+"
").indexOf(T)>=0:!U[4]?S:V==="!="?X!=T:V==="^="?X.indexOf(T)===0:V==="$="?X.s
ubstr(X.length-T.length)===T:V==="|="?X===T||X.substr(0,T.length+1)===T+"-
":false},POS:function(W,T,U,X){var S=T[2],V=H.setFilters[S];if(V){return
V(W,U,T,X)}}}};var L=H.match.POS;for(var N in
H.match){H.match[N]=RegExp(H.match[N].source+/(?![^\[]*\])(?![^\(]*\))/.sourc
e)}var
E=function(T,S){T=Array.prototype.slice.call(T);if(S){S.push.apply(S,T);retur
n S}return
T};try{Array.prototype.slice.call(document.documentElement.childNodes)}catch(
M){E=function(W,V){var T=V||[];if(G.call(W)==="[object
Array]"){Array.prototype.push.apply(T,W)}else{if(typeof
W.length==="number"){for(var
U=0,S=W.length;U<S;U++){T.push(W[U])}}else{for(var
U=0;W[U];U++){T.push(W[U])}}}return T}}(function(){var
T=document.createElement("form"),U="script"+(new
Date).getTime();T.innerHTML="<input name='"+U+"'/>";var
S=document.documentElement;S.insertBefore(T,S.firstChild);if(!!document.getEl
ementById(U)){H.find.ID=function(W,X,Y){if(typeof
X.getElementById!=="undefined"&&!Y){var V=X.getElementById(W[1]);return
V?V.id===W[1]||typeof
V.getAttributeNode!=="undefined"&&V.getAttributeNode("id").nodeValue===W[1]?[
V]:g:[]}};H.filter.ID=function(X,V){var W=typeof
X.getAttributeNode!=="undefined"&&X.getAttributeNode("id");return
X.nodeType===1&&W&&W.nodeValue===V}}S.removeChild(T)})();(function(){var
S=document.createElement("div");S.appendChild(document.createComment(""));if(
S.getElementsByTagName("*").length>0){H.find.TAG=function(T,X){var
W=X.getElementsByTagName(T[1]);if(T[1]==="*"){var V=[];for(var
U=0;W[U];U++){if(W[U].nodeType===1){V.push(W[U])}}W=V}return
W}}S.innerHTML="<a
href='#'></a>";if(S.firstChild&&S.firstChild.getAttribute("href")!=="#"){H.at
trHandle.href=function(T){return
T.getAttribute("href",2)}}})();if(document.querySelectorAll){(function(){var
S=F,T=document.createElement("div");T.innerHTML="<p
class='TEST'></p>";if(T.querySelectorAll&&T.querySelectorAll(".TEST").length=
==0){return}F=function(X,W,U,V){W=W||document;if(!V&&W.nodeType===9&&!P(W)){t
ry{return E(W.querySelectorAll(X),U)}catch(Y){}}return
S(X,W,U,V)};F.find=S.find;F.filter=S.filter;F.selectors=S.selectors;F.matches
=S.matches})()}if(document.getElementsByClassName&&document.documentElement.g
etElementsByClassName){H.order.splice(1,0,"CLASS");H.find.CLASS=function(S,T)
{return T.getElementsByClassName(S[1])}}function O(T,Z,Y,ac,aa,ab){for(var
W=0,U=ac.length;W<U;W++){var S=ac[W];if(S){S=S[T];var
X=false;while(S&&S.nodeType){var
V=S[Y];if(V){X=ac[V];break}if(S.nodeType===1&&!ab){S[Y]=W}if(S.nodeName===Z){
X=S;break}S=S[T]}ac[W]=X}}}function R(T,Y,X,ab,Z,aa){for(var
V=0,U=ab.length;V<U;V++){var S=ab[V];if(S){S=S[T];var
W=false;while(S&&S.nodeType){if(S[X]){W=ab[S[X]];break}if(S.nodeType===1){if(
!aa){S[X]=V}if(typeof
Y!=="string"){if(S===Y){W=true;break}}else{if(F.filter(Y,[S]).length>0){W=S;b
```

```
reak}}}S=S[T]}ab[V]=W}}}var
J=document.compareDocumentPosition?function(T,S){return
T.compareDocumentPosition(S)&16}:function(T,S){return
T!==S&&(T.contains?T.contains(S):true)};var P=function(S){return
S.nodeType===9&&S.documentElement.nodeName!=="HTML"||!!S.ownerDocument&&P(S.o
wnerDocument)};var I=function(S,Z){var
V=[],W="",X,U=Z.nodeType?[Z]:Z;while((X=H.match.PSEUDO.exec(S))){W+=X[0];S=S.
replace(H.match.PSEUDO,"")}S=H.relative[S]?S+"*":S;for(var
Y=0,T=U.length;Y<T;Y++){F(S,U[Y],V)}return
F.filter(W,V)};o.find=F;o.filter=F.filter;o.expr=F.selectors;o.expr[":"]=o.ex
pr.filters;F.selectors.filters.hidden=function(S){return"hidden"===S.type||o.
css(S,"display")==="none"||o.css(S,"visibility")==="hidden"};F.selectors.filt
ers.visible=function(S){return"hidden"!==S.type&&o.css(S,"display")!=="none"&
&o.css(S,"visibility")!=="hidden"};F.selectors.filters.animated=function(S){r
eturn o.grep(o.timers,function(T){return
S===T.elem}).length};o.multiFilter=function(U,S,T){if(T){U=":not("+U+")"}retu
rn F.matches(U,S)};o.dir=function(U,T){var
S=[],V=U[T];while(V&&V!=document){if(V.nodeType==1){S.push(V)}V=V[T]}return
S};o.nth=function(W,S,U,V){S=S||1;var
T=0;for(;W;W=W[U]){if(W.nodeType==1&&++T==S){break}}return
W};o.sibling=function(U,T){var
S=[];for(;U;U=U.nextSibling){if(U.nodeType==1&&U!=T){S.push(U)}}return
S};return;l.Sizzle=F})();o.event={add:function(I,F,H,K){if(I.nodeType==3||I.n
odeType==8){return}if(I.setInterval&&I!=l){I=l}if(!H.guid){H.guid=this.guid++
}if(K!==g){var G=H;H=this.proxy(G);H.data=K}var
E=o.data(I,"events")||o.data(I,"events",{}),J=o.data(I,"handle")||o.data(I,"h
andle",function(){return typeof
o!=="undefined"&&!o.event.triggered?o.event.handle.apply(arguments.callee.ele
m,arguments):g});J.elem=I;o.each(F.split(/\s+/),function(M,N){var
O=N.split(".");N=O.shift();H.type=O.slice().sort().join(".");var
L=E[N];if(o.event.specialAll[N]){o.event.specialAll[N].setup.call(I,K,O)}if(!
L){L=E[N]={};if(!o.event.special[N]||o.event.special[N].setup.call(I,K,O)===f
alse){if(I.addEventListener){I.addEventListener(N,J,false)}else{if(I.attachEv
ent){I.attachEvent("on"+N,J)}}}}L[H.guid]=H;o.event.global[N]=true});I=null},
guid:1,global:{},remove:function(K,H,J){if(K.nodeType==3||K.nodeType==8){retu
rn}var G=o.data(K,"events"),F,E;if(G){if(H===g||(typeof
H==="string"&&H.charAt(0)==".")){for(var I in
G){this.remove(K,I+(H||""))}}else{if(H.type){J=H.handler;H=H.type}o.each(H.sp
lit(/\s+/),function(M,O){var Q=O.split(".");O=Q.shift();var
N=RegExp("(^|\\.)"+Q.slice().sort().join(".*\\.")+"(\\.|$)");if(G[O]){if(J){d
elete G[O][J.guid]}else{for(var P in G[O]){if(N.test(G[O][P].type)){delete
G[O][P]}}}if(o.event.specialAll[O]){o.event.specialAll[O].teardown.call(K,Q)}
for(F in
G[O]){break}if(!F){if(!o.event.special[O]||o.event.special[O].teardown.call(K
,Q)===false){if(K.removeEventListener){K.removeEventListener(O,o.data(K,"hand
le"),false)}else{if(K.detachEvent){K.detachEvent("on"+O,o.data(K,"handle"))}}
}F=null;delete G[O]}}})}for(F in G){break}if(!F){var
L=o.data(K,"handle");if(L){L.elem=null}o.removeData(K,"events");o.removeData(
K,"handle")}}},trigger:function(I,K,H,E){var G=I.type||I;if(!E){I=typeof
I==="object"?I[h]?I:o.extend(o.Event(G),I):o.Event(G);if(G.indexOf("!")>=0){I
.type=G=G.slice(0,-
1);I.exclusive=true}if(!H){I.stopPropagation();if(this.global[G]){o.each(o.ca
che,function(){if(this.events&&this.events[G]){o.event.trigger(I,K,this.handl
e.elem)}})}}if(!H||H.nodeType==3||H.nodeType==8){return
g}I.result=g;I.target=H;K=o.makeArray(K);K.unshift(I)}I.currentTarget=H;var
J=o.data(H,"handle");if(J){J.apply(H,K)}if((!H[G]||(o.nodeName(H,"a")&&G=="cl
ick"))&&H["on"+G]&&H["on"+G].apply(H,K)===false){I.result=false}if(!E&&H[G]&&
!I.isDefaultPrevented()&&!(o.nodeName(H,"a")&&G=="click")){this.triggered=tru
```

```
e;try{H[G]()}catch(L){}}this.triggered=false;if(!I.isPropagationStopped()){va
r
F=H.parentNode||H.ownerDocument;if(F){o.event.trigger(I,K,F,true)}}},handle:f
unction(K){var J,E;K=arguments[0]=o.event.fix(K||l.event);var
L=K.type.split(".");K.type=L.shift();J=!L.length&&!K.exclusive;var
I=RegExp("(^|\\.)"+L.slice().sort().join(".*\\.")+"(\\.|$)");E=(o.data(this,"
events")||{})[K.type];for(var G in E){var
H=E[G];if(J||I.test(H.type)){K.handler=H;K.data=H.data;var
F=H.apply(this,arguments);if(F!==g){K.result=F;if(F===false){K.preventDefault
();K.stopPropagation()}}if(K.isImmediatePropagationStopped()){break}}}},props
:"altKey attrChange attrName bubbles button cancelable charCode clientX
clientY ctrlKey currentTarget data detail eventPhase fromElement handler
keyCode metaKey newValue originalTarget pageX pageY prevValue relatedNode
relatedTarget screenX screenY shiftKey srcElement target toElement view
wheelDelta which".split(" "),fix:function(H){if(H[h]){return H}var
F=H;H=o.Event(F);for(var G=this.props.length,J;G;){J=this.props[--
G];H[J]=F[J]}if(!H.target){H.target=H.srcElement||document}if(H.target.nodeTy
pe==3){H.target=H.target.parentNode}if(!H.relatedTarget&&H.fromElement){H.rel
atedTarget=H.fromElement==H.target?H.toElement:H.fromElement}if(H.pageX==null
&&H.clientX!=null){var
I=document.documentElement,E=document.body;H.pageX=H.clientX+(I&&I.scrollLeft
||E&&E.scrollLeft||0)-
(I.clientLeft||0);H.pageY=H.clientY+(I&&I.scrollTop||E&&E.scrollTop||0)-
(I.clientTop||0)}if(!H.which&&((H.charCode||H.charCode===0)?H.charCode:H.keyC
ode)){H.which=H.charCode||H.keyCode}if(!H.metaKey&&H.ctrlKey){H.metaKey=H.ctr
lKey}if(!H.which&&H.button){H.which=(H.button&1?1:(H.button&2?3:(H.button&4?2
:0)))}return H},proxy:function(F,E){E=E||function(){return
F.apply(this,arguments)};E.guid=F.guid=F.guid||E.guid||this.guid++;return
E},special:{ready:{setup:B,teardown:function(){}},specialAll:{live:{setup:fu
nction(E,F){o.event.add(this,F[0],c)},teardown:function(G){if(G.length){var
E=0,F=RegExp("(^|\\.)"+G[0]+"(\\.|$)");o.each((o.data(this,"events").live||{}
),function(){if(F.test(this.type)){E++}});if(E<1){o.event.remove(this,G[0],c)
}}}}}};o.Event=function(E){if(!this.preventDefault){return new
o.Event(E)}if(E&&E.type){this.originalEvent=E;this.type=E.type}else{this.type
=E}this.timeStamp=e();this[h]=true};function k(){return false}function
u(){return
true}o.Event.prototype={preventDefault:function(){this.isDefaultPrevented=u;v
ar
E=this.originalEvent;if(!E){return}if(E.preventDefault){E.preventDefault()}E.
returnValue=false},stopPropagation:function(){this.isPropagationStopped=u;var
E=this.originalEvent;if(!E){return}if(E.stopPropagation){E.stopPropagation()}
E.cancelBubble=true},stopImmediatePropagation:function(){this.isImmediateProp
agationStopped=u;this.stopPropagation()},isDefaultPrevented:k,isPropagationSt
opped:k,isImmediatePropagationStopped:k};var a=function(F){var
E=F.relatedTarget;while(E&&E!=this){try{E=E.parentNode}catch(G){E=this}}if(E!
=this){F.type=F.data;o.event.handle.apply(this,arguments)}};o.each({mouseover
:"mouseenter",mouseout:"mouseleave"},function(F,E){o.event.special[E]={setup:
function(){o.event.add(this,F,a,E)},teardown:function(){o.event.remove(this,F
,a)}}});o.fn.extend({bind:function(F,G,E){return
F=="unload"?this.one(F,G,E):this.each(function(){o.event.add(this,F,E||G,E&&G
)})},one:function(G,H,F){var
E=o.event.proxy(F||H,function(I){o(this).unbind(I,E);return(F||H).apply(this,
arguments)});return
this.each(function(){o.event.add(this,G,E,F&&H)})},unbind:function(F,E){retur
n
this.each(function(){o.event.remove(this,F,E)})},trigger:function(E,F){return
this.each(function(){o.event.trigger(E,F,this)})},triggerHandler:function(E,G
){if(this[0]){var
```

```
F=o.Event(E);F.preventDefault();F.stopPropagation();o.event.trigger(F,G,this[
0]);return F.result}},toggle:function(G){var
E=arguments,F=1;while(F<E.length){o.event.proxy(G,E[F++])}return
this.click(o.event.proxy(G,function(H){this.lastToggle=(this.lastToggle||0)%F
;H.preventDefault();return
E[this.lastToggle++].apply(this,arguments)||false}))},hover:function(E,F){ret
urn
this.mouseenter(E).mouseleave(F)},ready:function(E){B();if(o.isReady){E.call(
document,o)}else{o.readyList.push(E)}return this},live:function(G,F){var
E=o.event.proxy(F);E.guid+=this.selector+G;o(document).bind(i(G,this.selector
),this.selector,E);return
this},die:function(F,E){o(document).unbind(i(F,this.selector),E?{guid:E.guid+
this.selector+F}:null);return this}});function c(H){var
E=RegExp("(^|\\.)"+H.type+"(\\.|$)"),G=true,F=[];o.each(o.data(this,"events")
.live||[],function(I,J){if(E.test(J.type)){var
K=o(H.target).closest(J.data)[0];if(K){F.push({elem:K,fn:J})}}});o.each(F,fun
ction(){if(this.fn.call(this.elem,H,this.fn.data)===false){G=false}});return
G}function i(F,E){return["live",F,E.replace(/\./g,"`").replace(/
/g,"|")].join(".")}o.extend({isReady:false,readyList:[],ready:function(){if(!
o.isReady){o.isReady=true;if(o.readyList){o.each(o.readyList,function(){this.
call(document,o)});o.readyList=null}o(document).triggerHandler("ready")}}});v
ar x=false;function
B(){if(x){return}x=true;if(document.addEventListener){document.addEventListen
er("DOMContentLoaded",function(){document.removeEventListener("DOMContentLoad
ed",arguments.callee,false);o.ready()},false)}else{if(document.attachEvent){d
ocument.attachEvent("onreadystatechange",function(){if(document.readyState===
"complete"){document.detachEvent("onreadystatechange",arguments.callee);o.rea
dy()}});if(document.documentElement.doScroll&&typeof
l.frameElement==="undefined"){(function(){if(o.isReady){return}try{document.d
ocumentElement.doScroll("left")}catch(E){setTimeout(arguments.callee,0);retur
n}o.ready()})()}}}o.event.add(l,"load",o.ready)}o.each(("blur,focus,load,resi
ze,scroll,unload,click,dblclick,mousedown,mouseup,mousemove,mouseover,mouseou
t,mouseenter,mouseleave,change,select,submit,keydown,keypress,keyup,error").s
plit(","),function(F,E){o.fn[E]=function(G){return
G?this.bind(E,G):this.trigger(E)}});o(l).bind("unload",function(){for(var E
in
o.cache){if(E!=1&&o.cache[E].handle){o.event.remove(o.cache[E].handle.elem)}}
});(function(){o.support={};var
F=document.documentElement,G=document.createElement("script"),K=document.crea
teElement("div"),J="script"+(new
Date).getTime();K.style.display="none";K.innerHTML='
<link/><table></table><a href="/a"
style="color:red;float:left;opacity:.5;">a</a><select><option>text</option></
select><object><param/></object>';var
H=K.getElementsByTagName("*"),E=K.getElementsByTagName("a")[0];if(!H||!H.leng
th||!E){return}o.support={leadingWhitespace:K.firstChild.nodeType==3,tbody:!K
.getElementsByTagName("tbody").length,objectAll:!!K.getElementsByTagName("obj
ect")[0].getElementsByTagName("*").length,htmlSerialize:!!K.getElementsByTagN
ame("link").length,style:/red/.test(E.getAttribute("style")),hrefNormalized:E
.getAttribute("href")==="/a",opacity:E.style.opacity==="0.5",cssFloat:!!E.sty
le.cssFloat,scriptEval:false,noCloneEvent:true,boxModel:null};G.type="text/ja
vascript";try{G.appendChild(document.createTextNode("window."+J+"=1;"))}catch
(I){}F.insertBefore(G,F.firstChild);if(l[J]){o.support.scriptEval=true;delete
l[J]}F.removeChild(G);if(K.attachEvent&&K.fireEvent){K.attachEvent("onclick",
function(){o.support.noCloneEvent=false;K.detachEvent("onclick",arguments.cal
lee)});K.cloneNode(true).fireEvent("onclick")}o(function(){var
L=document.createElement("div");L.style.width="1px";L.style.paddingLeft="1px"
;document.body.appendChild(L);o.boxModel=o.support.boxModel=L.offsetWidth===2
```

```
;document.body.removeChild(L)})})();var
w=o.support.cssFloat?"cssFloat":"styleFloat";o.props={"for":"htmlFor","class"
:"className","float":w,cssFloat:w,styleFloat:w,readonly:"readOnly",maxlength:
"maxLength",cellspacing:"cellSpacing",rowspan:"rowSpan",tabindex:"tabIndex"};
o.fn.extend({_load:o.fn.load,load:function(G,J,K){if(typeof
G!=="string"){return this._load(G)}var I=G.indexOf(" ");if(I>=0){var
E=G.slice(I,G.length);G=G.slice(0,I)}var
H="GET";if(J){if(o.isFunction(J)){K=J;J=null}else{if(typeof
J==="object"){J=o.param(J);H="POST"}}}var
F=this;o.ajax({url:G,type:H,dataType:"html",data:J,complete:function(M,L){if(
L=="success"||L=="notmodified"){F.html(E?o("<div/>").append(M.responseText.re
place(/<script(.|\s)*?\/script>/g,"")).find(E):M.responseText)}if(K){F.each(K
,[M.responseText,L,M])}}});return this},serialize:function(){return
o.param(this.serializeArray())},serializeArray:function(){return
this.map(function(){return
this.elements?o.makeArray(this.elements):this}).filter(function(){return
this.name&&!this.disabled&&(this.checked||/select|textarea/i.test(this.nodeNa
me)||/text|hidden|password/i.test(this.type))}).map(function(E,F){var
G=o(this).val();return
G==null?null:o.isArray(G)?o.map(G,function(I,H){return{name:F.name,value:I}})
:{name:F.name,value:G}}).get()}});o.each("ajaxStart,ajaxStop,ajaxComplete,aja
xError,ajaxSuccess,ajaxSend".split(","),function(E,F){o.fn[F]=function(G){ret
urn this.bind(F,G)}});var
r=e();o.extend({get:function(E,G,H,F){if(o.isFunction(G)){H=G;G=null}return
o.ajax({type:"GET",url:E,data:G,success:H,dataType:F})},getScript:function(E,
F){return o.get(E,null,F,"script")},getJSON:function(E,F,G){return
o.get(E,F,G,"json")},post:function(E,G,H,F){if(o.isFunction(G)){H=G;G={}}retu
rn
o.ajax({type:"POST",url:E,data:G,success:H,dataType:F})},ajaxSetup:function(E
){o.extend(o.ajaxSettings,E)},ajaxSettings:{url:location.href,global:true,typ
e:"GET",contentType:"application/x-www-form-
urlencoded",processData:true,async:true,xhr:function(){return
l.ActiveXObject?new ActiveXObject("Microsoft.XMLHTTP"):new
XMLHttpRequest()},accepts:{xml:"application/xml,
text/xml",html:"text/html",script:"text/javascript,
application/javascript",json:"application/json,
text/javascript",text:"text/plain",_default:"*/*"}},lastModified:{},ajax:func
tion(M){M=o.extend(true,M,o.extend(true,{},o.ajaxSettings,M));var
W,F=/=\?(&|$)/g,R,V,G=M.type.toUpperCase();if(M.data&&M.processData&&typeof
M.data!=="string"){M.data=o.param(M.data)}if(M.dataType=="jsonp"){if(G=="GET"
){if(!M.url.match(F)){M.url+=(M.url.match(/\?/)?"&":"?")+(M.jsonp||"callback"
)+"=?"}}else{if(!M.data||!M.data.match(F)){M.data=(M.data?M.data+"&":"")+(M.j
sonp||"callback")+"=?"}}M.dataType="json"}if(M.dataType=="json"&&(M.data&&M.d
ata.match(F)||M.url.match(F))){W="jsonp"+r++;if(M.data){M.data=(M.data+"").re
place(F,"="+W+"$1")}M.url=M.url.replace(F,"="+W+"$1");M.dataType="script";l[W
]=function(X){V=X;I();L();l[W]=g;try{delete
l[W]}catch(Y){}if(H){H.removeChild(T)}}}if(M.dataType=="script"&&M.cache==nul
l){M.cache=false}if(M.cache===false&&G=="GET"){var E=e();var
U=M.url.replace(/(\?|&)_=.*?(&|$)/,"$1_="+E+"$2");M.url=U+((U==M.url)?(M.url.
match(/\?/)?"&":"?")+"_="+E:"")}if(M.data&&G=="GET"){M.url+=(M.url.match(/\?/
)?"&":"?")+M.data;M.data=null}if(M.global&&!o.active++){o.event.trigger("ajax
Start")}var
Q=/^(\w+:)?\/\/([^\/?#]+)/.exec(M.url);if(M.dataType=="script"&&G=="GET"&&Q&&
(Q[1]&&Q[1]!=location.protocol||Q[2]!=location.host)){var
H=document.getElementsByTagName("head")[0];var
T=document.createElement("script");T.src=M.url;if(M.scriptCharset){T.charset=
M.scriptCharset}if(!W){var
O=false;T.onload=T.onreadystatechange=function(){if(!O&&(!this.readyState||th
```

```
is.readyState=="loaded"||this.readyState=="complete")){O=true;I();L();H.remov
eChild(T)}}}H.appendChild(T);return g}var K=false;var
J=M.xhr();if(M.username){J.open(G,M.url,M.async,M.username,M.password)}else{J
.open(G,M.url,M.async)}try{if(M.data){J.setRequestHeader("Content-
Type",M.contentType)}if(M.ifModified){J.setRequestHeader("If-Modified-
Since",o.lastModified[M.url]||"Thu, 01 Jan 1970 00:00:00
GMT")}J.setRequestHeader("X-Requested-
With","XMLHttpRequest");J.setRequestHeader("Accept",M.dataType&&M.accepts[M.d
ataType]?M.accepts[M.dataType]+",
*/*":M.accepts._default)}catch(S){}if(M.beforeSend&&M.beforeSend(J,M)===false
){if(M.global&&!--o.active){o.event.trigger("ajaxStop")}J.abort();return
false}if(M.global){o.event.trigger("ajaxSend",[J,M])}var
N=function(X){if(J.readyState==0){if(P){clearInterval(P);P=null;if(M.global&&
!--
o.active){o.event.trigger("ajaxStop")}}}else{if(!K&&J&&(J.readyState==4||X=="
timeout")){K=true;if(P){clearInterval(P);P=null}R=X=="timeout"?"timeout":!o.h
ttpSuccess(J)?"error":M.ifModified&&o.httpNotModified(J,M.url)?"notmodified":
"success";if(R=="success"){try{V=o.httpData(J,M.dataType,M)}catch(Z){R="parse
rerror"}}if(R=="success"){var Y;try{Y=J.getResponseHeader("Last-
Modified")}catch(Z){}if(M.ifModified&&Y){o.lastModified[M.url]=Y}if(!W){I()}}
else{o.handleError(M,J,R)}L();if(X){J.abort()}if(M.async){J=null}}}};if(M.asy
nc){var
P=setInterval(N,13);if(M.timeout>0){setTimeout(function(){if(J&&!K){N("timeou
t")}},M.timeout)}}try{J.send(M.data)}catch(S){o.handleError(M,J,null,S)}if(!M
.async){N()}function
I(){if(M.success){M.success(V,R)}if(M.global){o.event.trigger("ajaxSuccess",[
J,M])}}function
L(){if(M.complete){M.complete(J,R)}if(M.global){o.event.trigger("ajaxComplete
",[J,M])}if(M.global&&!--o.active){o.event.trigger("ajaxStop")}}return
J},handleError:function(F,H,E,G){if(F.error){F.error(H,E,G)}if(F.global){o.ev
ent.trigger("ajaxError",[H,F,G])}},active:0,httpSuccess:function(F){try{retur
n
!F.status&&location.protocol=="file:"||(F.status>=200&&F.status<300)||F.statu
s==304||F.status==1223}catch(E){}return
false},httpNotModified:function(G,E){try{var H=G.getResponseHeader("Last-
Modified");return G.status==304||H==o.lastModified[E]}catch(F){}return
false},httpData:function(J,H,G){var F=J.getResponseHeader("content-
type"),E=H=="xml"||!H&&F&&F.indexOf("xml")>=0,I=E?J.responseXML:J.responseTex
t;if(E&&I.documentElement.tagName=="parsererror"){throw"parsererror"}if(G&&G.
dataFilter){I=G.dataFilter(I,H)}if(typeof
I==="string"){if(H=="script"){o.globalEval(I)}if(H=="json"){I=l["eval"]("("+I
+")")}}return I},param:function(E){var G=[];function
H(I,J){G[G.length]=encodeURIComponent(I)+"="+encodeURIComponent(J)}if(o.isArr
ay(E)||E.jquery){o.each(E,function(){H(this.name,this.value)})}else{for(var F
in
E){if(o.isArray(E[F])){o.each(E[F],function(){H(F,this)})}else{H(F,o.isFuncti
on(E[F])?E[F]():E[F])}}}return G.join("&").replace(/%20/g,"+")}});var
m={},n,d=[["height","marginTop","marginBottom","paddingTop","paddingBottom"],
["width","marginLeft","marginRight","paddingLeft","paddingRight"],["opacity"]
];function t(F,E){var
G={};o.each(d.concat.apply([],d.slice(0,E)),function(){G[this]=F});return
G}o.fn.extend({show:function(J,L){if(J){return
this.animate(t("show",3),J,L)}else{for(var H=0,F=this.length;H<F;H++){var
E=o.data(this[H],"olddisplay");this[H].style.display=E||"";if(o.css(this[H],"
display")==="none"){var G=this[H].tagName,K;if(m[G]){K=m[G]}else{var
I=o("<"+G+"
/>").appendTo("body");K=I.css("display");if(K==="none"){K="block"}I.remove();
m[G]=K}this[H].style.display=o.data(this[H],"olddisplay",K)}}return
```

```
this}},hide:function(H,I){if(H){return
this.animate(t("hide",3),H,I)}else{for(var G=0,F=this.length;G<F;G++){var
E=o.data(this[G],"olddisplay");if(!E&&E!=="none"){o.data(this[G],"olddisplay"
,o.css(this[G],"display"))}this[G].style.display="none"}return
this}},_toggle:o.fn.toggle,toggle:function(G,F){var E=typeof
G==="boolean";return
o.isFunction(G)&&o.isFunction(F)?this._toggle.apply(this,arguments):G==null||
E?this.each(function(){var
H=E?G:o(this).is(":hidden");o(this)[H?"show":"hide"]()}):this.animate(t("togg
le",3),G,F)},fadeTo:function(E,G,F){return
this.animate({opacity:G},E,F)},animate:function(I,F,H,G){var
E=o.speed(F,H,G);return this[E.queue===false?"each":"queue"](function(){var
K=o.extend({},E),M,L=this.nodeType==1&&o(this).is(":hidden"),J=this;for(M in
I){if(I[M]=="hide"&&L||I[M]=="show"&&!L){return
K.complete.call(this)}if((M=="height"||M=="width")&&this.style){K.display=o.c
ss(this,"display");K.overflow=this.style.overflow}}if(K.overflow!=null){this.
style.overflow="hidden"}K.curAnim=o.extend({},I);o.each(I,function(O,S){var
R=new
o.fx(J,K,O);if(/toggle|show|hide/.test(S)){R[S=="toggle"?L?"show":"hide":S](I
)}else{var Q=S.toString().match(/^([+-]=)?([\d+-
.]+)(.*)$/),T=R.cur(true)||0;if(Q){var
N=parseFloat(Q[2]),P=Q[3]||"px";if(P!="px"){J.style[O]=(N||1)+P;T=((N||1)/R.c
ur(true))*T;J.style[O]=T+P}if(Q[1]){N=((Q[1]=="-="?-
1:1)*N)+T}R.custom(T,N,P)}else{R.custom(T,S,"")}}});return
true})},stop:function(F,E){var
G=o.timers;if(F){this.queue([])}this.each(function(){for(var H=G.length-
1;H>=0;H--
){if(G[H].elem==this){if(E){G[H](true)}G.splice(H,1)}});if(!E){this.dequeue(
)}return
this}});o.each({slideDown:t("show",1),slideUp:t("hide",1),slideToggle:t("togg
le",1),fadeIn:{opacity:"show"},fadeOut:{opacity:"hide"}},function(E,F){o.fn[E
]=function(G,H){return
this.animate(F,G,H)}});o.extend({speed:function(G,H,F){var E=typeof
G==="object"?G:{complete:F||!F&&H||o.isFunction(G)&&G,duration:G,easing:F&&H|
|H&&!o.isFunction(H)&&H};E.duration=o.fx.off?0:typeof
E.duration==="number"?E.duration:o.fx.speeds[E.duration]||o.fx.speeds._defaul
t;E.old=E.complete;E.complete=function(){if(E.queue!==false){o(this).dequeue(
)}if(o.isFunction(E.old)){E.old.call(this)}};return
E},easing:{linear:function(G,H,E,F){return
E+F*G},swing:function(G,H,E,F){return((-
Math.cos(G*Math.PI)/2)+0.5)*F+E}},timers:[],fx:function(F,E,G){this.options=E
;this.elem=F;this.prop=G;if(!E.orig){E.orig={}}});o.fx.prototype={update:fun
ction(){if(this.options.step){this.options.step.call(this.elem,this.now,this)
}(o.fx.step[this.prop]||o.fx.step._default)(this);if((this.prop=="height"||th
is.prop=="width")&&this.elem.style){this.elem.style.display="block"}},cur:fun
ction(F){if(this.elem[this.prop]!=null&&(!this.elem.style||this.elem.style[th
is.prop]==null)){return this.elem[this.prop]}var
E=parseFloat(o.css(this.elem,this.prop,F));return E&&E>-
10000?E:parseFloat(o.curCSS(this.elem,this.prop))||0},custom:function(I,H,G){
this.startTime=e();this.start=I;this.end=H;this.unit=G||this.unit||"px";this.
now=this.start;this.pos=this.state=0;var E=this;function F(J){return
E.step(J)}F.elem=this.elem;if(F()&&o.timers.push(F)==1){n=setInterval(functio
n(){var K=o.timers;for(var J=0;J<K.length;J++){if(!K[J]()){K.splice(J--
,1)}}if(!K.length){clearInterval(n)}},13)}},show:function(){this.options.orig
[this.prop]=o.attr(this.elem.style,this.prop);this.options.show=true;this.cus
tom(this.prop=="width"||this.prop=="height"?1:0,this.cur());o(this.elem).show
()},hide:function(){this.options.orig[this.prop]=o.attr(this.elem.style,this.
prop);this.options.hide=true;this.custom(this.cur(),0)},step:function(H){var
```

```
G=e();if(H||G>=this.options.duration+this.startTime){this.now=this.end;this.p
os=this.state=1;this.update();this.options.curAnim[this.prop]=true;var
E=true;for(var F in
this.options.curAnim){if(this.options.curAnim[F]!==true){E=false}}if(E){if(th
is.options.display!=null){this.elem.style.overflow=this.options.overflow;this
.elem.style.display=this.options.display;if(o.css(this.elem,"display")=="none
"){this.elem.style.display="block"}}if(this.options.hide){o(this.elem).hide()
}if(this.options.hide||this.options.show){for(var I in
this.options.curAnim){o.attr(this.elem.style,I,this.options.orig[I])}}this.op
tions.complete.call(this.elem)}return false}else{var J=G-
this.startTime;this.state=J/this.options.duration;this.pos=o.easing[this.opti
ons.easing||(o.easing.swing?"swing":"linear")](this.state,J,0,1,this.options.
duration);this.now=this.start+((this.end-
this.start)*this.pos);this.update()}return
true}};o.extend(o.fx,{speeds:{slow:600,fast:200,_default:400},step:{opacity:f
unction(E){o.attr(E.elem.style,"opacity",E.now)},_default:function(E){if(E.el
em.style&&E.elem.style[E.prop]!=null){E.elem.style[E.prop]=E.now+E.unit}else{
E.elem[E.prop]=E.now}}}});if(document.documentElement.getBoundingClientRect){
o.fn.offset=function(){if(!this[0]){return{top:0,left:0}}if(this[0]===this[0]
.ownerDocument.body){return o.offset.bodyOffset(this[0])}var
G=this[0].getBoundingClientRect(),J=this[0].ownerDocument,F=J.body,E=J.docume
ntElement,L=E.clientTop||F.clientTop||0,K=E.clientLeft||F.clientLeft||0,I=G.t
op+(self.pageYOffset||o.boxModel&&E.scrollTop||F.scrollTop)-
L,H=G.left+(self.pageXOffset||o.boxModel&&E.scrollLeft||F.scrollLeft)-
K;return{top:I,left:H}}}else{o.fn.offset=function(){if(!this[0]){return{top:0
,left:0}}if(this[0]===this[0].ownerDocument.body){return
o.offset.bodyOffset(this[0])}o.offset.initialized||o.offset.initialize();var
J=this[0],G=J.offsetParent,F=J,O=J.ownerDocument,M,H=O.documentElement,K=O.bo
dy,L=O.defaultView,E=L.getComputedStyle(J,null),N=J.offsetTop,I=J.offsetLeft;
while((J=J.parentNode)&&J!==K&&J!==H){M=L.getComputedStyle(J,null);N-
=J.scrollTop,I-
=J.scrollLeft;if(J===G){N+=J.offsetTop,I+=J.offsetLeft;if(o.offset.doesNotAdd
Border&&!(o.offset.doesAddBorderForTableAndCells&&/^t(able|d|h)$/i.test(J.tag
Name))){N+=parseInt(M.borderTopWidth,10)||0,I+=parseInt(M.borderLeftWidth,10)
||0}F=G,G=J.offsetParent}if(o.offset.subtractsBorderForOverflowNotVisible&&M.
overflow!=="visible"){N+=parseInt(M.borderTopWidth,10)||0,I+=parseInt(M.borde
rLeftWidth,10)||0}E=M}if(E.position==="relative"||E.position==="static"){N+=K
.offsetTop,I+=K.offsetLeft}if(E.position==="fixed"){N+=Math.max(H.scrollTop,K
.scrollTop),I+=Math.max(H.scrollLeft,K.scrollLeft)}return{top:N,left:I}}}o.of
fset={initialize:function(){if(this.initialized){return}var
L=document.body,F=document.createElement("div"),H,G,N,I,M,E,J=L.style.marginT
op,K='<div style="position:absolute;top:0;left:0;margin:0;border:5px solid
#000;padding:0;width:1px;height:1px;"><div></div></div><table
style="position:absolute;top:0;left:0;margin:0;border:5px solid
#000;padding:0;width:1px;height:1px;" cellpadding="0"
cellspacing="0"><tr><td></td></tr></table>';M={position:"absolute",top:0,left
:0,margin:0,border:0,width:"1px",height:"1px",visibility:"hidden"};for(E in
M){F.style[E]=M[E]}F.innerHTML=K;L.insertBefore(F,L.firstChild);H=F.firstChil
d,G=H.firstChild,I=H.nextSibling.firstChild.firstChild;this.doesNotAddBorder=
(G.offsetTop!==5);this.doesAddBorderForTableAndCells=(I.offsetTop===5);H.styl
e.overflow="hidden",H.style.position="relative";this.subtractsBorderForOverfl
owNotVisible=(G.offsetTop===-
5);L.style.marginTop="1px";this.doesNotIncludeMarginInBodyOffset=(L.offsetTop
===0);L.style.marginTop=J;L.removeChild(F);this.initialized=true},bodyOffset:
function(E){o.offset.initialized||o.offset.initialize();var
G=E.offsetTop,F=E.offsetLeft;if(o.offset.doesNotIncludeMarginInBodyOffset){G+
=parseInt(o.curCSS(E,"marginTop",true),10)||0,F+=parseInt(o.curCSS(E,"marginL
eft",true),10)||0}return{top:G,left:F}}};o.fn.extend({position:function(){var
```

```
I=0,H=0,F;if(this[0]){var
G=this.offsetParent(),J=this.offset(),E=/^body|html$/i.test(G[0].tagName)?{to
p:0,left:0}:G.offset();J.top-=j(this,"marginTop");J.left-
=j(this,"marginLeft");E.top+=j(G,"borderTopWidth");E.left+=j(G,"borderLeftWid
th");F={top:J.top-E.top,left:J.left-E.left}}return
F},offsetParent:function(){var
E=this[0].offsetParent||document.body;while(E&&(!/^body|html$/i.test(E.tagNam
e)&&o.css(E,"position")=="static")){E=E.offsetParent}return
o(E)}});o.each(["Left","Top"],function(F,E){var
G="scroll"+E;o.fn[G]=function(H){if(!this[0]){return null}return
H!==g?this.each(function(){this==l||this==document?l.scrollTo(!F?H:o(l).scrol
lLeft(),F?H:o(l).scrollTop()):this[G]=H}):this[0]==l||this[0]==document?self[
F?"pageYOffset":"pageXOffset"]||o.boxModel&&document.documentElement[G]||docu
ment.body[G]:this[0][G]}});o.each(["Height","Width"],function(H,F){var
E=H?"Left":"Top",G=H?"Right":"Bottom";o.fn["inner"+F]=function(){return
this[F.toLowerCase()]()+j(this,"padding"+E)+j(this,"padding"+G)};o.fn["outer"
+F]=function(J){return
this["inner"+F]()+j(this,"border"+E+"Width")+j(this,"border"+G+"Width")+(J?j(
this,"margin"+E)+j(this,"margin"+G):0)};var
I=F.toLowerCase();o.fn[I]=function(J){return
this[0]==l?document.compatMode=="CSS1Compat"&&document.documentElement["clien
t"+F]||document.body["client"+F]:this[0]==document?Math.max(document.document
Element["client"+F],document.body["scroll"+F],document.documentElement["scrol
l"+F],document.body["offset"+F],document.documentElement["offset"+F]):J===g?(
this.length?o.css(this[0],I):null):this.css(I,typeof
J==="string"?J:J+"px")}})})();
```

```
/*
 * jQuery UI Effects 1.5.3
 *
 * Copyright (c) 2008 Aaron Eisenberger (aaronchi@gmail.com)
 * Dual licensed under the MIT (MIT-LICENSE.txt)
 * and GPL (GPL-LICENSE.txt) licenses.
 *
 * http://docs.jquery.com/UI/Effects/
 */
;(function($) {

$.effects = $.effects || {}; //Add the 'effects' scope

$.extend($.effects, {
	save: function(el, set) {
		for(var i=0;i<set.length;i++) {
			if(set[i] !== null) $.data(el[0], "ec.storage."+set[i],
el[0].style[set[i]]);
		}
	},
	restore: function(el, set) {
		for(var i=0;i<set.length;i++) {
			if(set[i] !== null) el.css(set[i], $.data(el[0],
"ec.storage."+set[i]));
		}
	},
	setMode: function(el, mode) {
		if (mode == 'toggle') mode = el.is(':hidden') ? 'show' : 'hide';
// Set for toggle
		return mode;
	},
	getBaseline: function(origin, original) { // Translates a [top,left]
array into a baseline value
		// this should be a little more flexible in the future to handle
a string & hash
		var y, x;
		switch (origin[0]) {
			case 'top': y = 0; break;
			case 'middle': y = 0.5; break;
			case 'bottom': y = 1; break;
			default: y = origin[0] / original.height;
		};
		switch (origin[1]) {
			case 'left': x = 0; break;
			case 'center': x = 0.5; break;
			case 'right': x = 1; break;
			default: x = origin[1] / original.width;
		};
		return {x: x, y: y};
	},
	createWrapper: function(el) {
		if (el.parent().attr('id') == 'fxWrapper')
			return el;
		var props = {width: el.outerWidth({margin:true}), height:
el.outerHeight({margin:true}), 'float': el.css('float')};
		el.wrap('<div id="fxWrapper" style="font-
size:100%;background:transparent;border:none;margin:0;padding:0"></div>');
		var wrapper = el.parent();
```

```
            if (el.css('position') == 'static'){
                  wrapper.css({position: 'relative'});
                  el.css({position: 'relative'});
            } else {
                  var top = el.css('top'); if(isNaN(parseInt(top))) top =
'auto';
                  var left = el.css('left'); if(isNaN(parseInt(left))) left =
'auto';
                  wrapper.css({ position: el.css('position'), top: top, left:
left, zIndex: el.css('z-index') }).show();
                  el.css({position: 'relative', top:0, left:0});
            }
            wrapper.css(props);
            return wrapper;
      },
      removeWrapper: function(el) {
            if (el.parent().attr('id') == 'fxWrapper')
                  return el.parent().replaceWith(el);
            return el;
      },
      setTransition: function(el, list, factor, val) {
            val = val || {};
            $.each(list,function(i, x){
                  unit = el.cssUnit(x);
                  if (unit[0] > 0) val[x] = unit[0] * factor + unit[1];
            });
            return val;
      },
      animateClass: function(value, duration, easing, callback) {

            var cb = (typeof easing == "function" ? easing : (callback ?
callback : null));
            var ea = (typeof easing == "object" ? easing : null);

            return this.each(function() {

                  var offset = {}; var that = $(this); var oldStyleAttr =
that.attr("style") || '';
                  if(typeof oldStyleAttr == 'object') oldStyleAttr =
oldStyleAttr["cssText"]; /* Stupidly in IE, style is a object.. */
                  if(value.toggle) { that.hasClass(value.toggle) ?
value.remove = value.toggle : value.add = value.toggle; }

                  //Let's get a style offset
                  var oldStyle = $.extend({}, (document.defaultView ?
document.defaultView.getComputedStyle(this,null) : this.currentStyle));
                  if(value.add) that.addClass(value.add); if(value.remove)
that.removeClass(value.remove);
                  var newStyle = $.extend({}, (document.defaultView ?
document.defaultView.getComputedStyle(this,null) : this.currentStyle));
                  if(value.add) that.removeClass(value.add); if(value.remove)
that.addClass(value.remove);

                  // The main function to form the object for animation
                  for(var n in newStyle) {
                        if( typeof newStyle[n] != "function" && newStyle[n]
/* No functions and null properties */
```

```
                        && n.indexOf("Moz") == -1 && n.indexOf("length") == -
1 /* No mozilla spezific render properties. */
                        && newStyle[n] != oldStyle[n] /* Only values that
have changed are used for the animation */
                        && (n.match(/color/i) || (!n.match(/color/i) &&
!isNaN(parseInt(newStyle[n],10)))) /* Only things that can be parsed to
integers or colors */
                        && (oldStyle.position != "static" ||
(oldStyle.position == "static" && !n.match(/left|top|bottom|right/))) /* No
need for positions when dealing with static positions */
                        ) offset[n] = newStyle[n];
                }

                that.animate(offset, duration, ea, function() { // Animate
the newly constructed offset object
                        // Change style attribute back to original. For
stupid IE, we need to clear the damn object.
                        if(typeof $(this).attr("style") == 'object') {
$(this).attr("style")["cssText"] = ""; $(this).attr("style")["cssText"] =
oldStyleAttr; } else $(this).attr("style", oldStyleAttr);
                        if(value.add) $(this).addClass(value.add);
if(value.remove) $(this).removeClass(value.remove);
                        if(cb) cb.apply(this, arguments);
                });

        });
    }
});

//Extend the methods of jQuery
$.fn.extend({
    //Save old methods
    _show: $.fn.show,
    _hide: $.fn.hide,
    __toggle: $.fn.toggle,
    _addClass: $.fn.addClass,
    _removeClass: $.fn.removeClass,
    _toggleClass: $.fn.toggleClass,
    // New ec methods
    effect: function(fx,o,speed,callback) {
        return $.effects[fx] ? $.effects[fx].call(this, {method: fx,
options: o || {}, duration: speed, callback: callback }) : null;
    },
    show: function() {
        if(!arguments[0] || (arguments[0].constructor == Number ||
/(slow|normal|fast)/.test(arguments[0])))
            return this._show.apply(this, arguments);
        else {
            var o = arguments[1] || {}; o['mode'] = 'show';
            return this.effect.apply(this, [arguments[0], o,
arguments[2] || o.duration, arguments[3] || o.callback]);
        }
    },
    hide: function() {
        if(!arguments[0] || (arguments[0].constructor == Number ||
/(slow|normal|fast)/.test(arguments[0])))
            return this._hide.apply(this, arguments);
        else {
```

```
                var o = arguments[1] || {}; o['mode'] = 'hide';
                return this.effect.apply(this, [arguments[0], o,
arguments[2] || o.duration, arguments[3] || o.callback]);
            }
      },
      toggle: function(){
            if(!arguments[0] || (arguments[0].constructor == Number ||
/(slow|normal|fast)/.test(arguments[0])) || (arguments[0].constructor ==
Function))
                return this.__toggle.apply(this, arguments);
            else {
                var o = arguments[1] || {}; o['mode'] = 'toggle';
                return this.effect.apply(this, [arguments[0], o,
arguments[2] || o.duration, arguments[3] || o.callback]);
            }
      },
      addClass: function(classNames,speed,easing,callback) {
            return speed ? $.effects.animateClass.apply(this, [{ add:
classNames },speed,easing,callback]) : this._addClass(classNames);
      },
      removeClass: function(classNames,speed,easing,callback) {
            return speed ? $.effects.animateClass.apply(this, [{ remove:
classNames },speed,easing,callback]) : this._removeClass(classNames);
      },
      toggleClass: function(classNames,speed,easing,callback) {
            return speed ? $.effects.animateClass.apply(this, [{ toggle:
classNames },speed,easing,callback]) : this._toggleClass(classNames);
      },
      morph: function(remove,add,speed,easing,callback) {
            return $.effects.animateClass.apply(this, [{ add: add, remove:
remove },speed,easing,callback]);
      },
      switchClass: function() {
            return this.morph.apply(this, arguments);
      },
      // helper functions
      cssUnit: function(key) {
            var style = this.css(key), val = [];
            $.each( ['em','px','%','pt'], function(i, unit){
                if(style.indexOf(unit) > 0)
                    val = [parseFloat(style), unit];
            });
            return val;
      }
});

/*
 * jQuery Color Animations
 * Copyright 2007 John Resig
 * Released under the MIT and GPL licenses.
 */

// We override the animation for all of these color styles
jQuery.each(['backgroundColor', 'borderBottomColor', 'borderLeftColor',
'borderRightColor', 'borderTopColor', 'color', 'outlineColor'],
function(i,attr){
            jQuery.fx.step[attr] = function(fx){
                if ( fx.state == 0 ) {
```

```
                                        fx.start = getColor( fx.elem, attr );
                                        fx.end = getRGB( fx.end );
                                }

                                fx.elem.style[attr] = "rgb(" + [
                                        Math.max(Math.min( parseInt((fx.pos *
(fx.end[0] - fx.start[0])) + fx.start[0]), 255), 0),
                                        Math.max(Math.min( parseInt((fx.pos *
(fx.end[1] - fx.start[1])) + fx.start[1]), 255), 0),
                                        Math.max(Math.min( parseInt((fx.pos *
(fx.end[2] - fx.start[2])) + fx.start[2]), 255), 0)
                                ].join(",") + ")";
                        }
});

// Color Conversion functions from highlightFade
// By Blair Mitchelmore
// http://jquery.offput.ca/highlightFade/

// Parse strings looking for color tuples [255,255,255]
function getRGB(color) {
                var result;

                // Check if we're already dealing with an array of colors
                if ( color && color.constructor == Array && color.length == 3 )
                        return color;

                // Look for rgb(num,num,num)
                if (result = /rgb\(\s*([0-9]{1,3})\s*,\s*([0-9]{1,3})\s*,\s*([0-
9]{1,3})\s*\)/.exec(color))
                        return [parseInt(result[1]), parseInt(result[2]),
parseInt(result[3])];

                // Look for rgb(num%,num%,num%)
                if (result = /rgb\(\s*([0-9]+(?:\.[0-9]+)?)\%\s*,\s*([0-
9]+(?:\.[0-9]+)?)\%\s*,\s*([0-9]+(?:\.[0-9]+)?)\%\s*\)/.exec(color))
                        return [parseFloat(result[1])*2.55,
parseFloat(result[2])*2.55, parseFloat(result[3])*2.55];

                // Look for #a0b1c2
                if (result = /#([a-fA-F0-9]{2})([a-fA-F0-9]{2})([a-fA-F0-
9]{2})/.exec(color))
                        return [parseInt(result[1],16),
parseInt(result[2],16), parseInt(result[3],16)];

                // Look for #fff
                if (result = /#([a-fA-F0-9])([a-fA-F0-9])([a-fA-F0-
9])/.exec(color))
                        return [parseInt(result[1]+result[1],16),
parseInt(result[2]+result[2],16), parseInt(result[3]+result[3],16)];

                // Look for rgba(0, 0, 0, 0) == transparent in Safari 3
                if (result = /rgba\(0, 0, 0, 0\)/.exec(color))
                        return colors['transparent']

                // Otherwise, we're most likely dealing with a named color
                return colors[jQuery.trim(color).toLowerCase()];
}
```

```
function getColor(elem, attr) {
            var color;

            do {
                        color = jQuery.curCSS(elem, attr);

                        // Keep going until we find an element that has
color, or we hit the body
                        if ( color != '' && color != 'transparent' ||
jQuery.nodeName(elem, "body") )
                                    break;

                        attr = "backgroundColor";
            } while ( elem = elem.parentNode );

            return getRGB(color);
};

// Some named colors to work with
// From Interface by Stefan Petre
// http://interface.eyecon.ro/

var colors = {
      aqua:[0,255,255],
      azure:[240,255,255],
      beige:[245,245,220],
      black:[0,0,0],
      blue:[0,0,255],
      brown:[165,42,42],
      cyan:[0,255,255],
      darkblue:[0,0,139],
      darkcyan:[0,139,139],
      darkgrey:[169,169,169],
      darkgreen:[0,100,0],
      darkkhaki:[189,183,107],
      darkmagenta:[139,0,139],
      darkolivegreen:[85,107,47],
      darkorange:[255,140,0],
      darkorchid:[153,50,204],
      darkred:[139,0,0],
      darksalmon:[233,150,122],
      darkviolet:[148,0,211],
      fuchsia:[255,0,255],
      gold:[255,215,0],
      green:[0,128,0],
      indigo:[75,0,130],
      khaki:[240,230,140],
      lightblue:[173,216,230],
      lightcyan:[224,255,255],
      lightgreen:[144,238,144],
      lightgrey:[211,211,211],
      lightpink:[255,182,193],
      lightyellow:[255,255,224],
      lime:[0,255,0],
      magenta:[255,0,255],
      maroon:[128,0,0],
      navy:[0,0,128],
```

```
      olive:[128,128,0],
      orange:[255,165,0],
      pink:[255,192,203],
      purple:[128,0,128],
      violet:[128,0,128],
      red:[255,0,0],
      silver:[192,192,192],
      white:[255,255,255],
      yellow:[255,255,0],
      transparent: [255,255,255]
};

/*
 * jQuery Easing v1.3 - http://gsgd.co.uk/sandbox/jquery/easing/
 *
 * Uses the built in easing capabilities added In jQuery 1.1
 * to offer multiple easing options
 *
 * TERMS OF USE - jQuery Easing
 *
 * Open source under the BSD License.
 *
 * Copyright © 2008 George McGinley Smith
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
modification,
 * are permitted provided that the following conditions are met:
 *
 * Redistributions of source code must retain the above copyright notice,
this list of
 * conditions and the following disclaimer.
 * Redistributions in binary form must reproduce the above copyright notice,
this list
 * of conditions and the following disclaimer in the documentation and/or
other materials
 * provided with the distribution.
 *
 * Neither the name of the author nor the names of contributors may be used
to endorse
 * or promote products derived from this software without specific prior
written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS
IS" AND ANY
 * EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF
 * MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO
EVENT SHALL THE
 * COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL,
 * EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE
 * GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED
 * AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR
TORT (INCLUDING
```

```
 * NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED
 * OF THE POSSIBILITY OF SUCH DAMAGE.
 *
*/

// t: current time, b: begInnIng value, c: change In value, d: duration
jQuery.easing['jswing'] = jQuery.easing['swing'];

jQuery.extend( jQuery.easing,
{
      def: 'easeOutQuad',
      swing: function (x, t, b, c, d) {
            //alert(jQuery.easing.default);
            return jQuery.easing[jQuery.easing.def](x, t, b, c, d);
      },
      easeInQuad: function (x, t, b, c, d) {
            return c*(t/=d)*t + b;
      },
      easeOutQuad: function (x, t, b, c, d) {
            return -c *(t/=d)*(t-2) + b;
      },
      easeInOutQuad: function (x, t, b, c, d) {
            if ((t/=d/2) < 1) return c/2*t*t + b;
            return -c/2 * ((--t)*(t-2) - 1) + b;
      },
      easeInCubic: function (x, t, b, c, d) {
            return c*(t/=d)*t*t + b;
      },
      easeOutCubic: function (x, t, b, c, d) {
            return c*((t=t/d-1)*t*t + 1) + b;
      },
      easeInOutCubic: function (x, t, b, c, d) {
            if ((t/=d/2) < 1) return c/2*t*t*t + b;
            return c/2*((t-=2)*t*t + 2) + b;
      },
      easeInQuart: function (x, t, b, c, d) {
            return c*(t/=d)*t*t*t + b;
      },
      easeOutQuart: function (x, t, b, c, d) {
            return -c * ((t=t/d-1)*t*t*t - 1) + b;
      },
      easeInOutQuart: function (x, t, b, c, d) {
            if ((t/=d/2) < 1) return c/2*t*t*t*t + b;
            return -c/2 * ((t-=2)*t*t*t - 2) + b;
      },
      easeInQuint: function (x, t, b, c, d) {
            return c*(t/=d)*t*t*t*t + b;
      },
      easeOutQuint: function (x, t, b, c, d) {
            return c*((t=t/d-1)*t*t*t*t + 1) + b;
      },
      easeInOutQuint: function (x, t, b, c, d) {
            if ((t/=d/2) < 1) return c/2*t*t*t*t*t + b;
            return c/2*((t-=2)*t*t*t*t + 2) + b;
      },
      easeInSine: function (x, t, b, c, d) {
            return -c * Math.cos(t/d * (Math.PI/2)) + c + b;
```

```
	},
	easeOutSine: function (x, t, b, c, d) {
		return c * Math.sin(t/d * (Math.PI/2)) + b;
	},
	easeInOutSine: function (x, t, b, c, d) {
		return -c/2 * (Math.cos(Math.PI*t/d) - 1) + b;
	},
	easeInExpo: function (x, t, b, c, d) {
		return (t==0) ? b : c * Math.pow(2, 10 * (t/d - 1)) + b;
	},
	easeOutExpo: function (x, t, b, c, d) {
		return (t==d) ? b+c : c * (-Math.pow(2, -10 * t/d) + 1) + b;
	},
	easeInOutExpo: function (x, t, b, c, d) {
		if (t==0) return b;
		if (t==d) return b+c;
		if ((t/=d/2) < 1) return c/2 * Math.pow(2, 10 * (t - 1)) + b;
		return c/2 * (-Math.pow(2, -10 * --t) + 2) + b;
	},
	easeInCirc: function (x, t, b, c, d) {
		return -c * (Math.sqrt(1 - (t/=d)*t) - 1) + b;
	},
	easeOutCirc: function (x, t, b, c, d) {
		return c * Math.sqrt(1 - (t=t/d-1)*t) + b;
	},
	easeInOutCirc: function (x, t, b, c, d) {
		if ((t/=d/2) < 1) return -c/2 * (Math.sqrt(1 - t*t) - 1) + b;
		return c/2 * (Math.sqrt(1 - (t-=2)*t) + 1) + b;
	},
	easeInElastic: function (x, t, b, c, d) {
		var s=1.70158;var p=0;var a=c;
		if (t==0) return b;  if ((t/=d)==1) return b+c;  if (!p) p=d*.3;
		if (a < Math.abs(c)) { a=c; var s=p/4; }
		else var s = p/(2*Math.PI) * Math.asin (c/a);
		return -(a*Math.pow(2,10*(t-=1)) * Math.sin( (t*d-
s)*(2*Math.PI)/p )) + b;
	},
	easeOutElastic: function (x, t, b, c, d) {
		var s=1.70158;var p=0;var a=c;
		if (t==0) return b;  if ((t/=d)==1) return b+c;  if (!p) p=d*.3;
		if (a < Math.abs(c)) { a=c; var s=p/4; }
		else var s = p/(2*Math.PI) * Math.asin (c/a);
		return a*Math.pow(2,-10*t) * Math.sin( (t*d-s)*(2*Math.PI)/p ) +
c + b;
	},
	easeInOutElastic: function (x, t, b, c, d) {
		var s=1.70158;var p=0;var a=c;
		if (t==0) return b;  if ((t/=d/2)==2) return b+c;  if (!p)
p=d*(.3*1.5);
		if (a < Math.abs(c)) { a=c; var s=p/4; }
		else var s = p/(2*Math.PI) * Math.asin (c/a);
		if (t < 1) return -.5*(a*Math.pow(2,10*(t-=1)) * Math.sin( (t*d-
s)*(2*Math.PI)/p )) + b;
		return a*Math.pow(2,-10*(t-=1)) * Math.sin( (t*d-s)*(2*Math.PI)/p
)*.5 + c + b;
	},
	easeInBack: function (x, t, b, c, d, s) {
		if (s == undefined) s = 1.70158;
```

```
		return c*(t/=d)*t*((s+1)*t - s) + b;
	},
	easeOutBack: function (x, t, b, c, d, s) {
		if (s == undefined) s = 1.70158;
		return c*((t=t/d-1)*t*((s+1)*t + s) + 1) + b;
	},
	easeInOutBack: function (x, t, b, c, d, s) {
		if (s == undefined) s = 1.70158;
		if ((t/=d/2) < 1) return c/2*(t*t*(((s*=(1.525))+1)*t - s)) + b;
		return c/2*((t-=2)*t*(((s*=(1.525))+1)*t + s) + 2) + b;
	},
	easeInBounce: function (x, t, b, c, d) {
		return c - jQuery.easing.easeOutBounce (x, d-t, 0, c, d) + b;
	},
	easeOutBounce: function (x, t, b, c, d) {
		if ((t/=d) < (1/2.75)) {
			return c*(7.5625*t*t) + b;
		} else if (t < (2/2.75)) {
			return c*(7.5625*(t-=(1.5/2.75))*t + .75) + b;
		} else if (t < (2.5/2.75)) {
			return c*(7.5625*(t-=(2.25/2.75))*t + .9375) + b;
		} else {
			return c*(7.5625*(t-=(2.625/2.75))*t + .984375) + b;
		}
	},
	easeInOutBounce: function (x, t, b, c, d) {
		if (t < d/2) return jQuery.easing.easeInBounce (x, t*2, 0, c, d)
* .5 + b;
		return jQuery.easing.easeOutBounce (x, t*2-d, 0, c, d) * .5 +
c*.5 + b;
	}
});

/*
 *
 * TERMS OF USE - EASING EQUATIONS
 *
 * Open source under the BSD License.
 *
 * Copyright © 2001 Robert Penner
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
modification,
 * are permitted provided that the following conditions are met:
 *
 * Redistributions of source code must retain the above copyright notice,
this list of
 * conditions and the following disclaimer.
 * Redistributions in binary form must reproduce the above copyright notice,
this list
 * of conditions and the following disclaimer in the documentation and/or
other materials
 * provided with the distribution.
 *
 * Neither the name of the author nor the names of contributors may be used
to endorse
```

```
 * or promote products derived from this software without specific prior
written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS
IS" AND ANY
 * EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF
 * MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO
EVENT SHALL THE
 * COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL,
 * EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE
 * GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED
 * AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR
TORT (INCLUDING
 * NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED
 * OF THE POSSIBILITY OF SUCH DAMAGE.
 *
 */

})(jQuery);
```

```
/*
 * jQuery UI Effects Slide
 *
 * Copyright (c) 2008 Aaron Eisenberger (aaronchi@gmail.com)
 * Dual licensed under the MIT (MIT-LICENSE.txt)
 * and GPL (GPL-LICENSE.txt) licenses.
 *
 * http://docs.jquery.com/UI/Effects/Slide
 *
 * Depends:
 *    effects.core.js
 */
(function($) {

$.effects.slide = function(o) {

      return this.queue(function() {

            // Create element
            var el = $(this), props = ['position','top','left'];

            // Set options
            var mode = $.effects.setMode(el, o.options.mode || 'show'); //
Set Mode
            var direction = o.options.direction || 'left'; // Default
Direction

            // Adjust
            $.effects.save(el, props); el.show(); // Save & Show
            $.effects.createWrapper(el).css({overflow:'hidden'}); // Create
Wrapper
            var ref = (direction == 'up' || direction == 'down') ? 'top' :
'left';
            var motion = (direction == 'up' || direction == 'left') ? 'pos' :
'neg';
            var distance = o.options.distance || (ref == 'top' ?
el.outerHeight({margin:true}) : el.outerWidth({margin:true}));
            if (mode == 'show') el.css(ref, motion == 'pos' ? -distance :
distance); // Shift

            // Animation
            var animation = {};
            animation[ref] = (mode == 'show' ? (motion == 'pos' ? '+=' : '-
=') : (motion == 'pos' ? '-=' : '+=')) + distance;

            // Animate
            el.animate(animation, { queue: false, duration: o.duration,
easing: o.options.easing, complete: function() {
                  if(mode == 'hide') el.hide(); // Hide
                  $.effects.restore(el, props); $.effects.removeWrapper(el);
// Restore
                  if(o.callback) o.callback.apply(this, arguments); //
Callback
                  el.dequeue();
            }});

      });
```

```
};

})(jQuery);
```

```
/*
____thesis.js____
by Whitney Anne Trettien

This file, written in JQuery, governs the showing and hiding of the four main
text blocks (divs "#tl," "#tr," "#bl," "#br"), as well the visual map.
*/

$(document).ready(function(){

var volvelles =
".back1029,.back1030,.back1,.back2,.back3,.back4,.back5,.back6,.back7,.back8,
.back9,.back10,.back11,.back12,.back13,.back1000,.back57,.back400,.back401,.b
ack402,.back403,.back404,.back405,.back406,.back407,.back408,.back409,.back41
0,.back300,.back301,.back302,.back303,.back304,.back305,.back306,.back307,.ba
ck308,.back309,.back310,.back311,.back312,.back313,.back314,.back315,.back316
,.back317,.back318,.back319,.back320,.back321,.back322,.back323,.back1021,.ba
ck1022,.back1023,.back1024,.back1025,.back1026,.back1028,.back190"
var stammworter =
".back14,.back15,.back16,.back17,.back18,.back19,.back20,.back21,.back22,.bac
k23,.back24,.back25,.back26,.back27,.back28,.back29,.back30,.back31,.back1007
"
var kabbalah =
".back32,.back33,.back34,.back35,.back36,.back37,.back38,.back39,.back40,.bac
k41,.back42"
var materiality =
".back46,.back4611,.back4622,.back4633,.back4644,.back4655,.back47,.back48,.b
ack49,.back12011,.back411,.back412,.back413,.back414,.back415,.back416,.back4
17,.back418,.back419,.back1009,.back1010,.back1015,.back1031,.back1033"
var hands =
".back53,.back54,.back55,.back5511,.back5522,.back56,.back5611,.back5622,.bac
k5633,.back5644,.back58,.back1001,.back1002,.back1012,.back1013,.back1014,.ba
ck1027";
var thirtyyearswar =
".back59,.back60,.back61,.back62,.back63,.back64,.back65,.back66,.back67,.bac
k68,.back1003,.back1004,.back1005,.back1006,.back1016";
var cutups =
".back100,.back101,.back102,.back103,.back104,.back105,.back106,.back107,.bac
k108,.back109,.back110,.back111,.back112,.back113,.back114,.back115,.back116,
.back117,.back118,.back119,.back11911,.back120,.back121,.back122,.back123,.ba
ck124,.back126,.back127,.back128,.back12911,.back130,.back131,.back13111,.bac
k13122,.back132,.back133,.back134,.back135,.back136,.back137,.back138,.back13
9,.back140,.back141,.back142,.back147,.back14711,.back14722,.back148,.back148
11,.back14822,.back149,.back1011";
var leibniz =
".back200,.back201,.back202,.back203,.back204,.back205,.back206,.back207,.bac
k208,.back209,.back210,.back211,.back212,.back213,.back214,.back215,.back216,
.back219,.back220,.back221,.back222,.back223,.back224,.back225,.back226,.back
227,.back228,.back229,.back230,.back231,.back232,.back233,.back234,.back235,.
back1017,.back1019";
var towerofbabel = ".back43,.back44,.back1008,.back45";
var computers =
".back129,.back143,.back144,.back146,.back150,.back151,.back152,.back153,.bac
k154,.back155,.back156,.back157,.back158,.back159,.back170,.back171,.back172,
.back173,.back174,.back175,.back176,.back177,.back178,.back179,.back180,.back
181,.back182,.back183,.back184,.back185,.back1032";
var permutation =
".back14833,.back161,.back162,.back163,.back164,.back250,.back251,.back252,.b
```

```
ack253,.back254,.back255,.back256,.back257,.back258,.back259,.back260,.back26
1,.back262,.back263,.back264,.back265,.back266,.back267,.back1020,.back1034";
var volvelles_c =
".cube1029,.cube1030,.cube1,.cube2,.cube3,.cube4,.cube5,.cube6,.cube7,.cube8,
.cube9,.cube10,.cube11,.cube12,.cube13,.cube1000,.cube57,.cube400,.cube401,.c
ube402,.cube403,.cube404,.cube405,.cube406,.cube407,.cube408,.cube409,.cube41
0,.cube300,.cube301,.cube302,.cube303,.cube304,.cube305,.cube306,.cube307,.cu
be308,.cube309,.cube310,.cube311,.cube312,.cube313,.cube314,.cube315,.cube316
,.cube317,.cube318,.cube319,.cube320,.cube321,.cube322,.cube323,.cube1021,.cu
be1022,.cube1023,.cube1024,.cube1025,.cube1026,.cube1028,.cube190"
var stammworter_c =
".cube14,.cube15,.cube16,.cube17,.cube18,.cube19,.cube20,.cube21,.cube22,.cub
e23,.cube24,.cube25,.cube26,.cube27,.cube28,.cube29,.cube30,.cube31,.cube1007
"
var kabbalah_c =
".cube32,.cube33,.cube34,.cube35,.cube36,.cube37,.cube38,.cube39,.cube40,.cub
e41,.cube42"
var materiality_c =
".cube46,.cube4611,.cube4622,.cube4633,.cube4644,.cube4655,.cube47,.cube48,.c
ube49,.cube12011,.cube411,.cube412,.cube413,.cube414,.cube415,.cube416,.cube4
17,.cube418,.cube419,.cube1009,.cube1010,.cube1015,.cube1031,.cube1033"
var hands_c =
".cube53,.cube54,.cube55,.cube5511,.cube5522,.cube56,.cube5611,.cube5622,.cub
e5633,.cube5644,.cube58,.cube1001,.cube1002,.cube1012,.cube1013,.cube1014,.cu
be1027";
var thirtyyearswar_c =
".cube59,.cube60,.cube61,.cube62,.cube63,.cube64,.cube65,.cube66,.cube67,.cub
e68,.cube1003,.cube1004,.cube1005,.cube1006,.cube1016";
var cutups_c =
".cube100,.cube101,.cube102,.cube103,.cube104,.cube105,.cube106,.cube107,.cub
e108,.cube109,.cube110,.cube111,.cube112,.cube113,.cube114,.cube115,.cube116,
.cube117,.cube118,.cube119,.cube11911,.cube120,.cube121,.cube122,.cube123,.cu
be124,.cube126,.cube127,.cube128,.cube12911,.cube130,.cube131,.cube13111,.cub
e13122,.cube132,.cube133,.cube134,.cube135,.cube136,.cube137,.cube138,.cube13
9,.cube140,.cube141,.cube142,.cube147,.cube14711,.cube14722,.cube148,.cube148
11,.cube14822,.cube149,.cube1011";
var leibniz_c =
".cube200,.cube201,.cube202,.cube203,.cube204,.cube205,.cube206,.cube207,.cub
e208,.cube209,.cube210,.cube211,.cube212,.cube213,.cube214,.cube215,.cube216,
.cube219,.cube220,.cube221,.cube222,.cube223,.cube224,.cube225,.cube226,.cube
227,.cube228,.cube229,.cube230,.cube231,.cube232,.cube233,.cube234,.cube235,.
cube1017,.cube1019";
var towerofbabel_c = ".cube43,.cube44,.cube1008,.cube45";
var computers_c =
".cube129,.cube143,.cube144,.cube146,.cube150,.cube151,.cube152,.cube153,.cub
e154,.cube155,.cube156,.cube157,.cube158,.cube159,.cube170,.cube171,.cube172,
.cube173,.cube174,.cube175,.cube176,.cube177,.cube178,.cube179,.cube180,.cube
181,.cube182,.cube183,.cube184,.cube185,.cube1032";
var permutation_c =
".cube14833,.cube161,.cube162,.cube163,.cube164,.cube250,.cube251,.cube252,.c
ube253,.cube254,.cube255,.cube256,.cube257,.cube258,.cube259,.cube260,.cube26
1,.cube262,.cube263,.cube264,.cube265,.cube266,.cube267,.cube1020,.cube1034";
var lb = $("#labelbox,#labelbox-shadow");

   $(volvelles+volvelles_c).mouseover(function () {
        $(volvelles).addClass('selected');
        $("#labelbox").css("color","#A6A600");
        $(lb).text("combinatory volvelles").removeClass("displaynone");
```

```
   }).mouseout(function () {
     $(volvelles).removeClass('selected');
     $(lb).text("").addClass("displaynone");
   });
$(stammworter+stammworter_c).mouseover(function () {
     $(stammworter).addClass('selected');
     $("#labelbox").css("color","#679B00").text("German stem-word theory
     of language");
     $(lb).text("German stem-word theory of
     language").removeClass("displaynone");
   }).mouseout(function () {
     $(stammworter).removeClass('selected');
     $(lb).text("").addClass("displaynone");
   });
$(kabbalah+kabbalah_c).mouseover(function () {
     $(kabbalah).addClass('selected');
     $("#labelbox").css("color","#679B00");
     $(lb).text("kabbalah").removeClass("displaynone");
   }).mouseout(function () {
     $(kabbalah).removeClass('selected');
     $(lb).text("").addClass("displaynone");
   });
$(materiality+materiality_c).mouseover(function () {
     $(materiality).addClass('selected');
     $("#labelbox").css("color","#CD0074");
     $(lb).text("materiality of letters").removeClass("displaynone");
   }).mouseout(function () {
     $(materiality).removeClass('selected');
     $(lb).text("").addClass("displaynone");
   });
$(hands+hands_c).mouseover(function () {
     $(hands).addClass('selected');
     $("#labelbox").css("color","#A6A600");
     $(lb).text("hands in books").removeClass("displaynone");
   }).mouseout(function () {
     $(hands).removeClass('selected');
     $(lb).text("").addClass("displaynone");
   });
$(thirtyyearswar+thirtyyearswar_c).mouseover(function () {
     $(thirtyyearswar).addClass('selected');
     $("#labelbox").css("color","#A6A600");
     $(lb).text("Thirty Years War").removeClass("displaynone");
   }).mouseout(function () {
     $(thirtyyearswar).removeClass('selected');
     $(lb).text("").addClass("displaynone");
   });
$(cutups+cutups_c).mouseover(function () {
     $(cutups).addClass('selected');
     $("#labelbox").css("color","#E6399B");
     $(lb).text("cut-ups").removeClass("displaynone");
   }).mouseout(function () {
     $(cutups).removeClass('selected');
     $(lb).text("").addClass("displaynone");
   });
$(towerofbabel+towerofbabel_c).mouseover(function () {
     $(towerofbabel).addClass('selected');
     $("#labelbox").css("color","#E667AF");
     $(lb).text("tower of babel").removeClass("displaynone");
```

```
      }).mouseout(function () {
        $(towerofbabel).removeClass('selected');
        $(lb).text("").addClass("displaynone");
      });
   $(leibniz+leibniz_c).mouseover(function () {
        $(leibniz).addClass('selected');
        $("#labelbox").css("color","#679B00");
        $(lb).text("Leibniz's alphabet of human
        thoughts").removeClass("displaynone");
      }).mouseout(function () {
        $(leibniz).removeClass('selected');
        $(lb).text("").addClass("displaynone");
      });
   $(computers+computers_c).mouseover(function () {
        $(computers).addClass('selected');
        $("#labelbox").css("color","#A6A600");
        $(lb).text("computers").removeClass("displaynone");
      }).mouseout(function () {
        $(computers).removeClass('selected');
        $(lb).text("").addClass("displaynone");
      });
   $(permutation+permutation_c).mouseover(function () {
        $(permutation).addClass('selected');
        $("#labelbox").css("color","#E6399B");
        $(lb).text("permutational poetry").removeClass("displaynone");
      }).mouseout(function () {
        $(permutation).removeClass('selected');
        $(lb).text("").addClass("displaynone");
      });
})
function yourText(id) {
     $(".back"+id+"").css("display","none");
     $(".page-inner-shadow").clone().insertBefore(".page-inner-
     shadow").removeClass("page-inner-
     shadow").load("textwithdots.php?o="+id+"");
     $(".page-inner").clone().insertBefore(".page-inner").removeClass("page-
     inner").load("textwithdots.php?o="+id+"");
}
function tr(id) {
     $('.present-topright').hide().removeClass("present-topright");
     $(".cube"+id+"").clone().css("height","300px").css("width","400px").css(
     "top","22px").css("left","602px").insertAfter(".topright-
     shadow").addClass('present-topright');
     $('#brh,#tlv').hide();
     $('#trv').load("toprightvert.php?o="+id+"").show("fast");
     $('#trh').load("toprighthori.php?o="+id+"").show("fast");
}
function tl(id){
     $('.present-topleft').hide().removeClass("present-topleft");
     $(".cube"+id+"").clone().css("height","300px").css("width","400px").css(
     "top","22px").css("left","182px").insertAfter(".topleft-
     shadow").addClass("present-topleft");
     $('#blh,#trv').hide();
     $('#tlv').load("topleftvert.php?o="+id+"").show("fast");
     $('#tlh').load("toplefthori.php?o="+id+"").show("fast");
}
function br(id) {
```

```
        $('.present-bottomright').hide().removeClass("present-bottomright");
        $(".cube"+id+"").clone().css("height","300px").css("width","400px").css(
        "top","342px").css("left","602px").insertAfter(".bottomright-
        shadow").addClass("present-bottomright");
        $('#trh,#blv').hide();
        $('#brv').load("bottomrightvert.php?o="+id+"").show("fast");
        $('#brh').load("bottomrighthori.php?o="+id+"").show("fast");
}
function bl(id) {
        $('.present-bottomleft').hide().removeClass("present-bottomleft");
        $(".cube"+id+"").clone().css("height","300px").css("width","400px").css(
        "top","342px").css("left","182px").insertAfter(".bottomleft-
        shadow").addClass("present-bottomleft");
        $('#tlh,#brv').hide();
        $('#blv').load("bottomleftvert.php?o="+id+"").show("fast");
        $('#blh').load("bottomlefthori.php?o="+id+"").show("fast");
}
function loadContentToprightvert(id) {
        tr(id);
        $('.present-topright').show("slide", { direction: "left" }, 1000);
        $("#tr").load("text.php?o="+id+"").show("slide", { direction: "left" },
        1000);
        yourText(id);
}
function loadContentToprighthori(id) {
        tr(id);
        $(".present-topright").show("slide", { direction: "down" }, 1000);
        $("#tr").load("text.php?o="+id+"").show("slide", { direction: "down" },
        1000);
        yourText(id);
}
function loadContentBottomleftvert(id) {
        bl(id);
        $(".present-bottomleft").show("slide", { direction: "right" }, 1000);
        $("#bl").load("text.php?o="+id+"").show("slide", { direction: "right" },
        1000);
        yourText(id);
}
function loadContentBottomlefthori(id) {
        bl(id);
        $(".present-bottomleft").show("slide", { direction: "up" }, 1000);
        $("#bl").load("text.php?o="+id+"").show("slide", { direction: "up" },
        1000);
        yourText(id);
}
function loadContentTopleftvert(id) {
        tl(id);
        $(".present-topleft").show("slide", { direction: "right" }, 1000);
        $("#tl").load("text.php?o="+id+"").show("slide", { direction: "right" },
        1000);
        yourText(id);
}
function loadContentToplefthori(id) {
        tl(id);
        $(".present-topleft").show("slide", { direction: "down" }, 1000);
        $("#tl").load("text.php?o="+id+"").show("slide", { direction: "down" },
        1000);
        yourText(id);
```

```
}
function loadContentBottomrightvert(id) {
      br(id);
      $(".present-bottomright").show("slide", { direction: "left" }, 1000);
      $("#br").load("text.php?o="+id+"").show("slide", { direction: "left" },
      1000);
      yourText(id);
}
function loadContentBottomrighthori(id) {
      br(id);
      $(".present-bottomright").show("slide", { direction: "up" },1000);
      $("#br").load("text.php?o="+id+"").show("slide", { direction: "up" },
      1000);
      yourText(id);
}
function hideAll(id) {
      tl(id);
      $(".present-topleft").show("slide", { direction: "right" }, 1000);
      $("#tl").load("text.php?o="+id+"").show("slide", { direction: "right" },
      1000);
      $("#tr,.present-topright,#br,.present-bottomright,#bl,.present-
      bottomleft,#trv,#trh,#blv,#blh,#tlv,#tlh,#brv,#brh").hide();
      yourText(id);
}
```

```
/*
 * Thickbox 3.1 - One Box To Rule Them All.
 * By Cody Lindley (http://www.codylindley.com)
 * Copyright (c) 2007 cody lindley
 * Licensed under the MIT License: http://www.opensource.org/licenses/mit-
license.php
*/

var tb_pathToImage = "images/loadingAnimation.gif";

/*!!!!!!!!!!!!!!!!! edit below this line at your own risk
!!!!!!!!!!!!!!!!!!!!!!!*/

//on page load call tb_init
$(document).ready(function(){
      tb_init('a.thickbox, area.thickbox, input.thickbox');//pass where to
apply thickbox
      imgLoader = new Image();// preload image
      imgLoader.src = tb_pathToImage;
});

//add thickbox to href & area elements that have a class of .thickbox
function tb_init(domChunk){
      $(domChunk).click(function(){
      var t = this.title || this.name || null;
      var a = this.href || this.alt;
      var g = this.rel || false;
      tb_show(t,a,g);
      this.blur();
      return false;
      });
}

function tb_show(caption, url, imageGroup) {//function called when the user
clicks on a thickbox link

      try {
            if (typeof document.body.style.maxHeight === "undefined") {//if
IE 6
                  $("body","html").css({height: "100%", width: "100%"});
                  $("html").css("overflow","hidden");
                  if (document.getElementById("TB_HideSelect") === null)
{//iframe to hide select elements in ie6
                        $("body").append("<iframe
id='TB_HideSelect'></iframe><div id='TB_overlay'></div><div
id='TB_window'></div>");
                        $("#TB_overlay").click(tb_remove);
                  }
            }else{//all others
                  if(document.getElementById("TB_overlay") === null){
                        $("body").append("<div id='TB_overlay'></div><div
id='TB_window'></div>");
                        $("#TB_overlay").click(tb_remove);
                  }
            }

            if(tb_detectMacXFF()){
```

```
                $("#TB_overlay").addClass("TB_overlayMacFFBGHack");//use
png overlay so hide flash
            }else{
                $("#TB_overlay").addClass("TB_overlayBG");//use background
and opacity
            }

            if(caption===null){caption="";}
            $("body").append("<div id='TB_load'><img src='"+imgLoader.src+"'
/></div>");//add loader to the page
            $('#TB_load').show();//show loader

            var baseURL;
         if(url.indexOf("?")!==-1){ //ff there is a query string involved
                baseURL = url.substr(0, url.indexOf("?"));
         }else{
                baseURL = url;
         }

         var urlString = /\.jpg$|\.jpeg$|\.png$|\.gif$|\.bmp$/;
         var urlType = baseURL.toLowerCase().match(urlString);

            if(urlType == '.jpg' || urlType == '.jpeg' || urlType == '.png'
|| urlType == '.gif' || urlType == '.bmp'){//code to show images

                TB_PrevCaption = "";
                TB_PrevURL = "";
                TB_PrevHTML = "";
                TB_NextCaption = "";
                TB_NextURL = "";
                TB_NextHTML = "";
                TB_imageCount = "";
                TB_FoundURL = false;
                if(imageGroup){
                      TB_TempArray = $("a[@rel="+imageGroup+"]").get();
                      for (TB_Counter = 0; ((TB_Counter <
TB_TempArray.length) && (TB_NextHTML === "")); TB_Counter++) {
                            var urlTypeTemp =
TB_TempArray[TB_Counter].href.toLowerCase().match(urlString);
                            if (!(TB_TempArray[TB_Counter].href ==
url)) {
                                  if (TB_FoundURL) {
                                        TB_NextCaption =
TB_TempArray[TB_Counter].title;
                                        TB_NextURL =
TB_TempArray[TB_Counter].href;
                                        TB_NextHTML = "<span
id='TB_next'>  <a href='#'>Next &gt;</a></span>";
                                  } else {
                                        TB_PrevCaption =
TB_TempArray[TB_Counter].title;
                                        TB_PrevURL =
TB_TempArray[TB_Counter].href;
                                        TB_PrevHTML = "<span
id='TB_prev'>  <a href='#'>&lt; Prev</a></span>";
                                  }
                            } else {
                                  TB_FoundURL = true;
```

```
                                        TB_imageCount = "Image " +
(TB_Counter + 1) +" of "+ (TB_TempArray.length);

                                    }
                              }
                        }

                        imgPreloader = new Image();
                        imgPreloader.onload = function(){
                        imgPreloader.onload = null;

                        // Resizing large images - orginal by Christian Montoya
edited by me.
                        var pagesize = tb_getPageSize();
                        var x = pagesize[0] - 150;
                        var y = pagesize[1] - 150;
                        var imageWidth = imgPreloader.width;
                        var imageHeight = imgPreloader.height;
                        if (imageWidth > x) {
                              imageHeight = imageHeight * (x / imageWidth);
                              imageWidth = x;
                              if (imageHeight > y) {
                                    imageWidth = imageWidth * (y / imageHeight);
                                    imageHeight = y;
                              }
                        } else if (imageHeight > y) {
                              imageWidth = imageWidth * (y / imageHeight);
                              imageHeight = y;
                              if (imageWidth > x) {
                                    imageHeight = imageHeight * (x / imageWidth);
                                    imageWidth = x;
                              }
                        }
                        // End Resizing

                        TB_WIDTH = imageWidth + 30;
                        TB_HEIGHT = imageHeight + 60;
                        $("#TB_window").append("<a href='' id='TB_ImageOff'
title='Close'><img id='TB_Image' src='"+url+"' width='"+imageWidth+"'
height='"+imageHeight+"' alt='"+caption+"'/></a>" + "<div
id='TB_caption'>"+caption+"<div id='TB_secondLine'>" + TB_imageCount +
TB_PrevHTML + TB_NextHTML + "</div></div><div id='TB_closeWindow'><a href='#'
id='TB_closeWindowButton' title='Close'>close</a> or Esc Key</div>");

                        $("#TB_closeWindowButton").click(tb_remove);

                        if (!(TB_PrevHTML === "")) {
                              function goPrev(){

      if($(document).unbind("click",goPrev)){$(document).unbind("click",goPre
v);}
                                    $("#TB_window").remove();
                                    $("body").append("<div id='TB_window'></div>");
                                    tb_show(TB_PrevCaption, TB_PrevURL,
imageGroup);
                                    return false;
                              }
                              $("#TB_prev").click(goPrev);
```

```
                }

                if (!(TB_NextHTML === "")) {
                        function goNext(){
                                $("#TB_window").remove();
                                $("body").append("<div id='TB_window'></div>");
                                tb_show(TB_NextCaption, TB_NextURL,
imageGroup);
                                return false;
                        }
                        $("#TB_next").click(goNext);

                }

                document.onkeydown = function(e){
                        if (e == null) { // ie
                                keycode = event.keyCode;
                        } else { // mozilla
                                keycode = e.which;
                        }
                        if(keycode == 27){ // close
                                tb_remove();
                        } else if(keycode == 190){ // display previous image
                                if(!(TB_NextHTML == "")){
                                        document.onkeydown = "";
                                        goNext();
                                }
                        } else if(keycode == 188){ // display next image
                                if(!(TB_PrevHTML == "")){
                                        document.onkeydown = "";
                                        goPrev();
                                }
                        }
                };

                tb_position();
                $("#TB_load").remove();
                $("#TB_ImageOff").click(tb_remove);
                $("#TB_window").css({display:"block"}); //for safari using
css instead of show
                };

                imgPreloader.src = url;
        }else{//code to show html

                var queryString = url.replace(/^[^\?]+\??/,'');
                var params = tb_parseQuery( queryString );

                TB_WIDTH = (params['width']*1) + 30 || 630; //defaults to
630 if no paramaters were added to URL
                TB_HEIGHT = (params['height']*1) + 40 || 440; //defaults to
440 if no paramaters were added to URL
                ajaxContentW = TB_WIDTH - 30;
                ajaxContentH = TB_HEIGHT - 45;

                if(url.indexOf('TB_iframe') != -1){// either iframe or ajax
window
                        urlNoQuery = url.split('TB_');
```

```
                                $("#TB_iframeContent").remove();
                                if(params['modal'] != "true"){//iframe no modal
                                      $("#TB_window").append("<div
id='TB_title'><div id='TB_ajaxWindowTitle'>"+caption+"</div><div
id='TB_closeAjaxWindow'><a href='#' id='TB_closeWindowButton'
title='Close'>CLOSE</a></div></div><iframe frameborder='0' hspace='0'
src='"+urlNoQuery[0]+"' id='TB_iframeContent'
name='TB_iframeContent"+Math.round(Math.random()*1000)+"'
onload='tb_showIframe()' style='width:"+(ajaxContentW +
29)+"px;height:"+(ajaxContentH + 17)+"px;' > </iframe>");
                                }else{//iframe modal
                                $("#TB_overlay").unbind();
                                      $("#TB_window").append("<iframe
frameborder='0' hspace='0' src='"+urlNoQuery[0]+"' id='TB_iframeContent'
name='TB_iframeContent"+Math.round(Math.random()*1000)+"'
onload='tb_showIframe()' style='width:"+(ajaxContentW +
29)+"px;height:"+(ajaxContentH + 17)+"px;'> </iframe>");
                                }
                  }else{// not an iframe, ajax
                                if($("#TB_window").css("display") != "block"){
                                      if(params['modal'] != "true"){//ajax no
modal
                                      $("#TB_window").append("<div
id='TB_title'><div id='TB_ajaxWindowTitle'>"+caption+"</div><div
id='TB_closeAjaxWindow'><a href='#' id='TB_closeWindowButton'>close</a> or
Esc Key</div></div><div id='TB_ajaxContent'
style='width:"+ajaxContentW+"px;height:"+ajaxContentH+"px'></div>");
                                      }else{//ajax modal
                                      $("#TB_overlay").unbind();
                                      $("#TB_window").append("<div
id='TB_ajaxContent' class='TB_modal'
style='width:"+ajaxContentW+"px;height:"+ajaxContentH+"px;'></div>");
                                      }
                                }else{//this means the window is already up, we
are just loading new content via ajax
                                      $("#TB_ajaxContent")[0].style.width =
ajaxContentW +"px";
                                      $("#TB_ajaxContent")[0].style.height =
ajaxContentH +"px";
                                      $("#TB_ajaxContent")[0].scrollTop = 0;
                                      $("#TB_ajaxWindowTitle").html(caption);
                                }
                  }

                  $("#TB_closeWindowButton").click(tb_remove);

                        if(url.indexOf('TB_inline') != -1){
                                $("#TB_ajaxContent").append($('#' +
params['inlineId']).children());
                                $("#TB_window").unload(function () {
                                      $('#' + params['inlineId']).append(
$("#TB_ajaxContent").children() ); // move elements back when you're finished
                                });
                                tb_position();
                                $("#TB_load").remove();
                                $("#TB_window").css({display:"block"});
                        }else if(url.indexOf('TB_iframe') != -1){
                                tb_position();
```

```
                              if($.browser.safari){//safari needs help
because it will not fire iframe onload
                                    $("#TB_load").remove();
                                    $("#TB_window").css({display:"block"});
                              }
                        }else{
                              $("#TB_ajaxContent").load(url += "&random=" +
(new Date().getTime()),function(){//to do a post change this load method
                                    tb_position();
                                    $("#TB_load").remove();
                                    tb_init("#TB_ajaxContent a.thickbox");
                                    $("#TB_window").css({display:"block"});
                              });
                        }

            }

            if(!params['modal']){
                  document.onkeyup = function(e){
                        if (e == null) { // ie
                              keycode = event.keyCode;
                        } else { // mozilla
                              keycode = e.which;
                        }
                        if(keycode == 27){ // close
                              tb_remove();
                        }
                  };
            }

      } catch(e) {
            //nothing here
      }
}

//helper functions below
function tb_showIframe(){
      $("#TB_load").remove();
      $("#TB_window").css({display:"block"});
}

function tb_remove() {
      $("#TB_imageOff").unbind("click");
      $("#TB_closeWindowButton").unbind("click");
      $("#TB_window").fadeOut("fast",function(){$('#TB_window,#TB_overlay,#TB
_HideSelect').trigger("unload").unbind().remove();});
      $("#TB_load").remove();
      if (typeof document.body.style.maxHeight == "undefined") {//if IE 6
            $("body","html").css({height: "auto", width: "auto"});
            $("html").css("overflow","");
      }
      document.onkeydown = "";
      document.onkeyup = "";
      return false;
}

function tb_position() {
```

```
$("#TB_window").css({marginLeft: '-' + parseInt((TB_WIDTH / 2),10) + 'px',
width: TB_WIDTH + 'px'});
      if ( !(jQuery.browser.msie && jQuery.browser.version < 7)) { // take
away IE6
            $("#TB_window").css({marginTop: '-' + parseInt((TB_HEIGHT /
2),10) + 'px'});
      }
}

function tb_parseQuery ( query ) {
   var Params = {};
   if ( ! query ) {return Params;}// return empty object
   var Pairs = query.split(/[;&]/);
   for ( var i = 0; i < Pairs.length; i++ ) {
      var KeyVal = Pairs[i].split('=');
      if ( ! KeyVal || KeyVal.length != 2 ) {continue;}
      var key = unescape( KeyVal[0] );
      var val = unescape( KeyVal[1] );
      val = val.replace(/\+/g, ' ');
      Params[key] = val;
   }
   return Params;
}

function tb_getPageSize(){
      var de = document.documentElement;
      var w = window.innerWidth || self.innerWidth || (de&&de.clientWidth) ||
document.body.clientWidth;
      var h = window.innerHeight || self.innerHeight || (de&&de.clientHeight)
|| document.body.clientHeight;
      arrayPageSize = [w,h];
      return arrayPageSize;
}

function tb_detectMacXFF() {
  var userAgent = navigator.userAgent.toLowerCase();
  if (userAgent.indexOf('mac') != -1 && userAgent.indexOf('firefox')!=-1) {
    return true;
  }
}
```

```
/* ----------------------------------------------------------------------------
-------------------------------------*/
/* ---------->>> global settings needed for thickbox <<<----------------------
-------------------------------------*/
/* ----------------------------------------------------------------------------
-------------------------------------*/
*{padding: 0; margin: 0;}

/* ----------------------------------------------------------------------------
-------------------------------------*/
/* ---------->>> thickbox specific link and font settings <<<-----------------
-------------------------------------*/
/* ----------------------------------------------------------------------------
-------------------------------------*/
#TB_window {
      font: 12px Arial, Helvetica, sans-serif;
      color: #333333;
}

#TB_secondLine {
      font: 10px Arial, Helvetica, sans-serif;
      color:#666666;
}

#TB_window a:link {color: #666666;}
#TB_window a:visited {color: #666666;}
#TB_window a:hover {color: #000;}
#TB_window a:active {color: #666666;}
#TB_window a:focus{color: #666666;}

/* ----------------------------------------------------------------------------
-------------------------------------*/
/* ---------->>> thickbox settings <<<-----------------------------------------
-------------------------------------*/
/* ----------------------------------------------------------------------------
-------------------------------------*/
#TB_overlay {
      position: fixed;
      z-index:100;
      top: 0px;
      left: 0px;
      height:100%;
      width:100%;
}

.TB_overlayMacFFBGHack {background: url(macFFBgHack.png) repeat;}
.TB_overlayBG {
      background-color:#000;
      filter:alpha(opacity=75);
      -moz-opacity: 0.75;
      opacity: 0.75;
}

* html #TB_overlay { /* ie6 hack */
     position: absolute;
     height: expression(document.body.scrollHeight >
document.body.offsetHeight ? document.body.scrollHeight :
document.body.offsetHeight + 'px');
```

```
}

#TB_window {
	position: fixed;
	background: #ffffff;
	z-index: 102;
	color:#000000;
	display:none;
	border: 4px solid #525252;
	text-align:left;
	top:50%;
	left:50%;
}

* html #TB_window { /* ie6 hack */
position: absolute;
margin-top: expression(0 - parseInt(this.offsetHeight / 2) + (TBWindowMargin
= document.documentElement && document.documentElement.scrollTop ||
document.body.scrollTop) + 'px');
}

#TB_window img#TB_Image {
	display:block;
	margin: 15px 0 0 15px;
	border-right: 1px solid #ccc;
	border-bottom: 1px solid #ccc;
	border-top: 1px solid #666;
	border-left: 1px solid #666;
}

#TB_caption{
	height:25px;
	padding:7px 30px 10px 25px;
	float:left;
}

#TB_closeWindow{
	height:25px;
	padding:11px 25px 10px 0;
	float:right;
}

#TB_closeAjaxWindow{
	padding:7px 10px 5px 0;
	margin-bottom:1px;
	text-align:right;
	float:right;
}

#TB_ajaxWindowTitle{
	float:left;
	padding:7px 0 5px 10px;
	margin-bottom:1px;
	font-weight:bold;
	font-size:1.5em;
	color:#CCCCCC;
}
```

```
#TB_title{
	background-color:white;
	height:27px;
}

#TB_ajaxContent{
	clear:both;
	padding:2px 15px 15px 15px;
	overflow:auto;
	text-align:left;
	line-height:1.4em;
}

#TB_ajaxContent.TB_modal{
	padding:15px;
}

#TB_ajaxContent p{
	padding:5px 0px 5px 0px;
}

#TB_load{
	position: fixed;
	display:none;
	height:13px;
	width:208px;
	z-index:103;
	top: 50%;
	left: 50%;
	margin: -6px 0 0 -104px; /* -height/2 0 0 -width/2 */
}

* html #TB_load { /* ie6 hack */
position: absolute;
margin-top: expression(0 - parseInt(this.offsetHeight / 2) + (TBWindowMargin
= document.documentElement && document.documentElement.scrollTop ||
document.body.scrollTop) + 'px');
}

#TB_HideSelect{
	z-index:99;
	position:fixed;
	top: 0;
	left: 0;
	background-color:#fff;
	border:none;
	filter:alpha(opacity=0);
	-moz-opacity: 0;
	opacity: 0;
	height:100%;
	width:100%;
}

* html #TB_HideSelect { /* ie6 hack */
     position: absolute;
     height: expression(document.body.scrollHeight >
document.body.offsetHeight ? document.body.scrollHeight :
document.body.offsetHeight + 'px');
```

```
}

#TB_iframeContent{
      clear:both;
      border:none;
      margin-bottom:-1px;
      margin-top:1px;
      _margin-bottom:1px;
}

/*
___style-intro.css___
by Whitney Anne Trettien

This is the stylesheet for all intro pages (intro1.html, intro2.html,
intro3.html, intro4.html, intro5.html, intro6.html, intro7.html, intro8.html,
intro9.html, coda1.html, coda2.html).
*/

body {
      background-color: white;
      font-family: Verdana, Helvetica, Arial, san-serif;
}p {
      margin: 10px 10px 10px 10px;
      line-height: 140%;
      font-size: .8em;
}.blockquote {
      font-size: .7em;
      line-height: 130%;
      margin: 6px 6px 6px 6px;
}blockquote {
      border-style: dotted;
      border-color: #d8d8d8;
      background-color: #f0f0f0;
}a:visited {
    text-decoration: none;
    color: #666666;
}a {
      text-decoration: none;
      color: #666666;
}a:hover {
      background-color: yellow;
      position:relative;
}a span {
      display: none;
}a:hover span {
      display: block;
      position:fixed;
      bottom:80px;
      right: 20px;
      /* formatting only styles */
      padding: 5px;
      margin: 10px;
      z-index: 100;
      background: #f8f8f8;
      border: 1px dotted #E667AF;
      font-size: .9em;
      color: #404040;
```

```
      opacity: 0.9;
      width: 300px;
      /* end formatting */
}img {
      margin-left: auto;
      margin-right: auto;
      margin-top: 10px;
      display: block;
      border:none;
}.cube {
      position:absolute;
      top:0px;
      background-color:#CCCCCC;
      height:9px;
      width:9px;
}
```

```
<!--
___intro1.html___
by Whitney Anne Trettien

The first page of the introduction, which pops up in a Thickbox.
-->

<!doctype html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>ARS COMBINATORIA, by whitney anne trettien</title>
<link rel="stylesheet" type="text/css" href="style-intro.css" />
<script src="scripts/jquery-1.3.1.min.js" type="text/javascript">
</script><script src="scripts/thickbox.js" type="text/javascript"></script>
</head>
<body>
<p style="text-align:center;"><i>The present determines where, in the object
from the past, that object&#39;s fore-history and after-history diverge so as
to circumscribe its nucleus.</i> &mdash; Walter Benjamin</p><p>Consider two
historical moments &mdash; nuclei, in Walter Benjamin&#39;s sense of the word
&mdash; marking both the beginning and the end of the book as a media
form.</p><p>The first is the publication of St&eacute;phane
Mallarm&eacute;&#39;s poem <a
href="http://www.google.com/books?id=O1xcAAAAMAAJ&printsec=copyright"
target="_blank"><i>Un coup de d&eacute;s Jamais N'Abolira Le
Hasard</i><span>A Throw of the Dice will Never Abolish Chance</span></a>
(1897), imagined as a metonym for his multivolume, combinatorial Book, <i>Le
Livre</i>. A radical experiment in design and typography, <i>Un coup de
d&eacute;s</i> privileges form over content &mdash or rather, form <i>as</i>
content, such that blank spaces, typography and the material folds of the
book, rather than semantics, generate what Mallarm&eacute; calls
&quot;prismatic subdivisions&quot; of meaning on the page.</p><p>This unusual
use of the book&#39;s architecture leaves the reader, rather than the writer,
to cull and combine these scattered fragments of text through multimodal acts
of association; thus the reader &mdash; Mallarm&eacute; prefers the word
&quot;operator,&quot; etymologically linked to &quot;work,&quot;
<i>oeuvre</i>, from the Latin <i>opus</i><sup><a href="#">1<span>Maurice
Blanchot, <i>The Book to Come</i>, trans. by Charlotte Mandell (Stanford, CA:
Stanford University Press, 2003): 242.</span></a></sup> &mdash; becomes an
(inter)active participant in the poem&#39;s construction. </p>
<table width="450px" style="position:absolute;bottom:5px;">
<tr>
<td width="33%">
<img src="white.PNG"/>
</td>
<td width="34%">
<div style="position:absolute;right:auto;left:auto;width:auto;">
<div class="cube" style="left:10px;background-color:yellow;"></div>
<div class="cube" style="left:20px;"></div><div class="cube"
style="left:30px;"></div><div class="cube" style="left:40px;"></div>
<div class="cube" style="left:50px;"></div><div class="cube"
style="left:60px;"></div><div class="cube" style="left:70px;"></div>
<div class="cube" style="left:80px;"></div><div class="cube"
style="left:90px;"></div><div class="cube" style="left:100px;"></div>
<div class="cube" style="left:110px;"></div></div></td><td width="33%"><a
href="intro2.html"><img src="right.PNG"/></a>
</td>
</tr>
```

```
</table>
</body>
</html>
```

```
<!--
___intro2.html___
by Whitney Anne Trettien

The second page of the introduction, which pops up in a Thickbox.
-->

<!doctype html>
 <head>
 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
 <title>ARS COMBINATORIA, by whitney anne trettien</title>
 <link rel="stylesheet" type="text/css" href="style-intro.css" />
 </head>
 <body>
 <p>Consider a second Benjaminian nucleus for our electronic present: Jacob
Rabinow&#39;s 1952 model for a hard disk drive, the first random access
storage device for computers. As Rabinow himself described it,</p>
 <blockquote><p class="blockquote">The notched-disk memory
&quot;doughnut&quot; can be thought of as a kind of book in which round pages
are slotted in such a way that each line on each page can be read by merely
spinning the page for one revolution; the notches in the pages provide the
&quot;windows&quot; through which the selected page can be read. In other
words, the book can be read without being opened.</p></blockquote>
 <p>In <i>Of Grammatology</i>, Derrida foretells the death of the &quot;onto-
encyclopedic or neo-Hegelian model of the great total book, the book of
absolute knowledge linking its own infinite dispersion to itself, in a
circle&quot;; yet this random access, infinitely re-writeable doughnut-shaped
disk drive in many ways promises precisely that &mdash; a &quot;book [that]
can be read without being opened.&quot;<sup><a href="#">2<span>Jacques
Derrida, &quot;The Book to Come,&quot; in <i>Paper Machine</i>, trans. by
Rachel Bowlby (Stanford, CA: Stanford University Press, 2005):
15.</span></a></sup> In fact, as Matthew Kirschenbaum points out,
Rabinow&#39;s descriptions seems &quot;to anticipate much in our own
contemporary response to electronic storage media: the book has become a
black box, and whatever is inscribed within its pages is designed for other
than human eyes.&quot;<sup><a href="#">3<span>I am indebted to the work of
Matthew Kirschenbaum for identifying and contextualizing this fascinating
early model for a hard disk drive; see Kirschenbaum, <i>Mechanisms: New Media
and the Forensic Imagination</i> (Cambridge, MA: The MIT Press, 2008): 80-
81.</span></a></sup></p>
 <table width="450px" style="position:absolute;bottom:5px;">
 <tr>
 <td width="33%">
 <a href="intro1.html"><img src="left.PNG"></a>
 </td>
 <td width="34%">
 <div style="position:absolute;right:auto;left:auto;width:auto;">
 <div class="cube" style="left:10px;"></div>
 <div class="cube" style="left:20px;background-color:yellow;"></div>
 <div class="cube" style="left:30px;"></div>
 <div class="cube" style="left:40px;"></div>
 <div class="cube" style="left:50px;"></div>
 <div class="cube" style="left:60px;"></div>
 <div class="cube" style="left:70px;"></div>
 <div class="cube" style="left:80px;"></div>
 <div class="cube" style="left:90px;"></div>
 <div class="cube" style="left:100px;"></div>
 <div class="cube" style="left:110px;"></div>
```

```
</div>
</td>
<td width="33%">
<a href="intro3.html"><img src="right.PNG"></a>
</td>
</tr>
</table>
</body>
</html>
```

```
<!--
___intro3.html___
by Whitney Anne Trettien

The third page of the introduction, which pops up in a Thickbox.
-->

<!doctype html>
<head>
 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
 <title>ARS COMBINATORIA, by whitney anne trettien</title>
 <link rel="stylesheet" type="text/css" href="style-intro.css" />
 <script src="scripts/jquery-1.3.1.min.js" type="text/javascript"></script>
 <script src="scripts/thickbox.js" type="text/javascript"></script>
 </head>
 <body style="background-color:white;">
 <p>On the one hand, then, is a book &mdash; importantly, a book written
forty years before the production of the first electronic digital computer
&mdash; that has retroactively come to signal the beginning of the digital
revolution and the emergence of postmodern hypertext theory;<sup><a
href="#">4<span>
 On Mallarm&eacute;&#39;s <i>Un coup de d&eacute;s</i> as the father of the
&quot;typographical revolution&quot;, see Marjorie Perloff, <i>The Futurist
Moment: Avant-garde, Avant Guerre, and the Language of Rupture</i> (Chicago,
IL: University of Chicago Press, 2003): 253n26. On Mallarm&eacute;&#39;s
<i>Un coup de d&eacute;s</i> as a bellwether for digital media, see Jerome
McGann, <i>Radiant Textuality: Literature After the World Wide Web</i> (New
York, NY: Palgrave Macmillan, 2004): 210-211; see also Christopher
Funkhouser, <i>Prehistoric Digital Poetry: An Archaeology of Forms 1959-
1995</i> (Tuscaloosa, AL: University of Alabama Press, 2007): 11; Jay David
Bolter, <i>Writing Space: Computers, Hypertext, and the Remediation of
Print</i> (Lawrence Erlbaum Associates, 2001): 153.</span></a></sup> on the
other is a digital storage model that projects both an imagined past and a
future promise as-yet unfulfilled by our current media ecology. Although
neither project was realized, both Mallarm&eacute;&#39;s <i>Le Livre</i> and
Rabinow&#39;s doughnut-shaped disk drive mark speculative moments in the
history of reading and writing &mdash; ruptures between our pre-digital past,
&quot;the book,&quot; and our now always/already digital present, theorized
as the &quot;book-to-come.&quot; As such, they help us map the history of
literacy and inscription from the epistemic stance of our rapidly-changing
digital present. For who &quot;read&quot; these computational devices? Who
&quot;wrote&quot; <i>on</i> and <i>in</i> and <i>to</i> them? And how do
these never-realized Books not only store and retrieve information &mdash;
the persistent &quot;container&quot; theory of media &mdash; but act as
language machines<sup><a href="#">5<span>On the notion of &quot;language
machines,&quot; see the introduction to <i>Language Machines: Technologies of
Literary and Cultural Production</i>, ed. by Jeffrey Masten, Peter
Stallybrass and Nancy J. Vickers (New York: Routledge, 1997). See also Lisa
Gitelman, <i>Scripts, Grooves and Writing Machines: Representing Technology
in the Edison Era</i> (Stanford, CA: Stanford University Press, 1999): 94;
Steve Caffrey and bpNichol,<i> Rational Geomancy: The Kids of the Book
Machine, The Collected Research Reports of The Toronto Research Group, 1973-
1982 </i>(Vancouver: Talonbooks, 1992): 60.</span></a></sup>, generating new
knowledge?</p>
 <table width="450px" style="position:absolute;bottom:5px;">
 <tr>
 <td width="33%">
 <a href="intro2.html"><img src="left.PNG"></a>
```

```
</td>
<td width="34%">
<div style="position:absolute;right:auto;left:auto;width:auto;">
<div class="cube" style="left:10px;"></div>
<div class="cube" style="left:20px;"></div>
<div class="cube" style="left:30px;background-color:yellow;"></div>
<div class="cube" style="left:40px;"></div>
<div class="cube" style="left:50px;"></div>
<div class="cube" style="left:60px;"></div>
<div class="cube" style="left:70px;"></div>
<div class="cube" style="left:80px;"></div>
<div class="cube" style="left:90px;"></div>
<div class="cube" style="left:100px;"></div>
<div class="cube" style="left:110px;"></div>
</div>
</td>
<td width="33%">
<a href="intro4.html"><img src="right.PNG"></a>
</td>
</tr>
</table>
</body>
</html>
```

```
<!--
___intro4.html___
by Whitney Anne Trettien

The fourth page of the introduction, which pops up in a Thickbox.
-->

<!doctype html>
 <head>
 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
 <title>ARS COMBINATORIA, by whitney anne trettien</title>
 <link rel="stylesheet" type="text/css" href="style-intro.css" />
 </head>
 <body style="background-color:white;">
 <p>Through an archaeology of text-generating mechanisms, the present work
excavates the deep history of reading and writing as material, combinatory
practices. On the one hand, by positing the physical manipulation of language
as a form of reading and writing, this archaeology answers Roger
Chartier&amp;#39;s call for book historians to &amp;quot;take on the task of
retracing forgotten gestures and habits&amp;quot; that do not fit &amp;quot;the
genealogy of our own contemporary manner of reading,&amp;quot; a call echoed in
much recent work on the &amp;quot;use&amp;quot; of early modern books.<sup><a
href="#">6<span>Chartier, <i>The Order of Books</i> (Stanford, CA: Stanford
University Press, 1994), 8-9. On book &amp;quot;use,&amp;quot; see Sherman, <i>Used
Books: Marking Readers in Renaissance England</i> (Philadelphia, PA:
University of Pennsylvania Press, 2008); Mazzio and Cormack, <i>Book Use,
Book Theory 1500-1700</i> (Chicago, IL: University of Chicago,
2005).</span></a></sup> It thus challenges our assumptions about how readers
and writers of the past made meaning from printed texts and, more broadly,
the expressive potentials of the printed book itself. Yet this archaeology of
<i>ars combinatoria</i>, the art of combination, also presents a imaginative
challenge to historians of the book. For if we accept physically cutting
paper or spinning a volvelle as a readerly <i>and</i> writerly act, then we
must also, like Mallarm&amp;eacute; or Rabinow before us, erase the boundaries we
have drawn between &amp;quot;the book&amp;quot; as a material form and "the digital"
as an epistemology, reconsidering the various literacies each facilitates or
forecloses. </p>
 <table width="450px" style="position:absolute;bottom:5px;">
 <tr>
 <td width="33%">
 <a href="intro3.html"><img src="left.PNG"></a>
 </td>
 <td width="34%">
 <div style="position:absolute;right:auto;left:auto;width:auto;">
 <div class="cube" style="left:10px;"></div>
 <div class="cube" style="left:20px;"></div>
 <div class="cube" style="left:30px;"></div>
 <div class="cube" style="left:40px;background-color:yellow;"></div>
 <div class="cube" style="left:50px;"></div>
 <div class="cube" style="left:60px;"></div>
 <div class="cube" style="left:70px;"></div>
 <div class="cube" style="left:80px;"></div>
 <div class="cube" style="left:90px;"></div>
 <div class="cube" style="left:100px;"></div>
 <div class="cube" style="left:110px;"></div>
 </div>
 </td>
 <td width="33%">
```

```
<a href="intro5.html"><img src="right.PNG"></a>
</td>
</tr>
</table>
</body>
</html>
```

```
<!--
___intro5.html___
by Whitney Anne Trettien

The fifth page of the introduction, which pops up in a Thickbox.
-->


<!doctype html>
 <head>
 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
 <title>ARS COMBINATORIA, by whitney anne trettien</title>
 <link rel="stylesheet" type="text/css" href="style-intro.css" />
 </head>
 <body style="background-color:white;">
 <p>As an archaeology of forms, the present work participates in the
methodology of <i>media archaeology</i>, a recent trend in media studies that
rejects the evolutionary model of technological change. As Erkki Huhtamo
defines it, media archaeology is:<p>
 <blockquote><p class="blockquote">the &quot;excavation&quot; of the ways
discursive traditions and formulations have been &#39;imprinted&#39; on
specific media machines and systems in different historical contexts,
contributing to their identity in terms of socially and ideologically
specific webs of signification.<sup><a href="#">7<span>Huhtamo, &quot;From
kaleidoscomaniac to cybernerd: Notes toward an archaeology of the
media,&quot; in <i>Electronic Culture: Technology and Visual
Representation</i>, ed. Timothy Druckrey (London: Aperture Foundation, 1996):
303.</span></a></sup></p></blockquote>
 <p>Rather than ideating history as a progressive movement from technological
simplicity to complexity, then, media archaeology uses the present episteme
&mdash; in our case, the digitally-mediated moment &mdash; as a map for
excavating diverse, little-studied forms and practices from the past, thereby
problematizing traditional histories. In other words, to return to Benjamin,
the <i>present</i> determines a media machine&#39;s nucleus, identifying when
it transitions from the unrecognizable into the familiar.</p>
 <table width="450px" style="position:absolute;bottom:5px;">
 <tr>
 <td width="33%">
 <a href="intro4.html"><img src="left.PNG"></a>
 </td>
 <td width="34%">
 <div style="position:absolute;right:auto;left:auto;width:auto;">
 <div class="cube" style="left:10px;"></div>
 <div class="cube" style="left:20px;"></div>
 <div class="cube" style="left:30px;"></div>
 <div class="cube" style="left:40px;"></div>
 <div class="cube" style="left:50px;background-color:yellow;"></div>
 <div class="cube" style="left:60px;"></div>
 <div class="cube" style="left:70px;"></div>
 <div class="cube" style="left:80px;"></div>
 <div class="cube" style="left:90px;"></div>
 <div class="cube" style="left:100px;"></div>
 <div class="cube" style="left:110px;"></div>
 </div>
 </td>
 <td width="33%">
 <a href="intro6.html"><img src="right.PNG"></a>
 </td>
```

```
</tr>
</table>
</body>
</html>
```

```
<!--
___intro6.html___
by Whitney Anne Trettien

The sixth page of the introduction, which pops up in a Thickbox.
-->

<!doctype html>
 <head>
 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
 <title>ARS COMBINATORIA, by whitney anne trettien</title>
 <link rel="stylesheet" type="text/css" href="style-intro.css" />
 </head>
 <body style="background-color:white;">
 <p>Although, methodologically, media archaeology appears to unnecessarily
(or even ahistorically) presentize history, in practice it emerges from a
strong moral imperative to respect the complexity of our past. As Siegfried
Zielinski writes, we must &quot;understand history as being present not only
when it demands to be accepted as a responsibility and a heavy burden&quot;
&mdash; that is, not only when it historicize our present situation &mdash;
&quot;but also when there is value in allowing it to develop as a special
attraction&quot; &mdash; that is, as a curious Other that simply does not fit
our historical narratives or conceptual models. In this way, the
archaeological approach forces us to defamiliarize our understanding of
cultures, technologies and knowledge itself.<sup><a
href="#">8<span>Zielinski, <i>Deep Time of the Media: Toward and Archaeology
of Hearing and Seeing by Technical Means</i>, trans. by Gloria Custance
(Cambridge, MA: The MIT Press, 2006): 3.</span></a></sup></p>
 <table width="450px" style="position:absolute;bottom:5px;">
 <tr>
 <td width="33%">
 <a href="intro5.html"><img src="left.PNG"></a>
 </td>
 <td width="34%">
 <div style="position:absolute;right:auto;left:auto;width:auto;">
 <div class="cube" style="left:10px;"></div>
 <div class="cube" style="left:20px;"></div>
 <div class="cube" style="left:30px;"></div>
 <div class="cube" style="left:40px;"></div>
 <div class="cube" style="left:50px;"></div>
 <div class="cube" style="left:60px;background-color:yellow;"></div>
 <div class="cube" style="left:70px;"></div>
 <div class="cube" style="left:80px;"></div>
 <div class="cube" style="left:90px;"></div>
 <div class="cube" style="left:100px;"></div>
 <div class="cube" style="left:110px;"></div>
 </div>
 </td>
 <td width="33%">
 <a href="intro7.html"><img src="right.PNG"></a>
 </td>
 </tr>
 </table>
 </body>
 </html>
```

```
<!--
___intro7.html___
by Whitney Anne Trettien

The seventh page of the introduction, which pops up in a Thickbox.
-->

<!doctype html>
 <head>
 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
 <title>ARS COMBINATORIA, by whitney anne trettien</title>
 <link rel="stylesheet" type="text/css" href="style-intro.css" />
 </head>
 <body style="background-color:white;">
 <p>Until recently, the term <i>remediation</i> has dominated the discourse
on media history, describing the ways in which &quot;new&quot; media absorb
and repurpose &quot;old&quot; media. As Jay David Bolter and Richard Grusin
put it, &quot;what is new about new media comes from the particular ways in
which they refashion older media and the ways in which older media refashion
themselves to answer the challenges of new media.&quot;<sup><a
href="#">9<span>Bolter and Grusin, <i>Remediation: Understanding New
Media</i> (Cambridge, MA: The MIT Press, 1999): 15.</span></a></sup>  While
Bolter and Grusin are primarily concerned with historicizing the present
&mdash; the subtitle of their landmark study of remediation is, after all,
<i>Understanding New Media</i> &mdash; many have since challenged their
historical methods, including its lack of human agency.<sup><a
href="#">10<span>Gitelman, <i>Always Already New: Media, History, and the
Data of Culture</i> (Cambridge, MA: The MIT Press, 2006): 9. On the passage
in Bolter and Grusin quoted above, Gitelman writes: &quot;However astute
their readings of the ways different media compare and contrast at a formal
level, Bolter and Grusin have trimmed out any mention of human agents, as if
media were naturally the way they are, without authors, designers, engineers,
entrepreneurs, programmers, investors, owners, or
audiences.&quot;</span></a></sup> Implicitly, the present study of
algorithmic methods of reading and writing presents a further critique, if
not of remediation itself, then of a methodology that locks media history
into a set of safe assumptions, thereby implicitly evaluating technologies in
terms of their Darwinian &quot;fitness&quot; within the environment of
capitalism and mass media. Rather than seeking the old in the new, the
present work attempts to, as Zielinski exhorts, &quot;find something new in
the old&quot;: and &quot;if we are lucky and find it, we shall have to say
goodbye to much that is familiar.&quot;<a href="#"><sup>11<span>Zielinski
3.</span></a></sup></p>
 <table width="450px" style="position:absolute;bottom:5px;">
<tr>
<td width="33%">
<a href="intro6.html"><img src="left.PNG"></a>
</td>
<td width="34%">
<div style="position:absolute;right:auto;left:auto;width:auto;">
<div class="cube" style="left:10px;"></div>
<div class="cube" style="left:20px;"></div>
<div class="cube" style="left:30px;"></div>
<div class="cube" style="left:40px;"></div>
<div class="cube" style="left:50px;"></div>
<div class="cube" style="left:60px;"></div>
<div class="cube" style="left:70px;background-color:yellow;"></div>
<div class="cube" style="left:80px;"></div>
```

```
<div class="cube" style="left:90px;"></div>
<div class="cube" style="left:100px;"></div>
<div class="cube" style="left:110px;"></div>
</div>
</td>
<td width="33%">
<a href="intro8.html"><img src="right.PNG"></a>
</td>
</tr>
</table>
</body>
</html>
```

```
<!--
___intro8.html___
by Whitney Anne Trettien

The eighth page of the introduction, which pops up in a Thickbox.
-->

<head>
 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
 <title>ARS COMBINATORIA, by whitney anne trettien</title>
 <link rel="stylesheet" type="text/css" href="style-intro.css" />
 </head>
 <body style="background-color:white;">
 <p>It is at this point that I&#39;m forced to drop the scholarly pretense of
the third person. After spending months researching combinatory reading and
writing practices &mdash; practices that are radically Other to us, so far
from (to return to Chartier) &quot;the genealogy of our own contemporary
manner of reading&quot; &mdash; I could not, in both theory and reality,
write a narrative history. The institutional conventions of scholarly
reading, writing and publication seek to familiarize and contain,
conceptualizing the production of knowledge and text itself as a process of
illumination, literally &quot;bringing to light&quot;; yet, as the
combinatory practices I was researching underscore, there is nothing
&quot;natural&quot; to these institutions. In fact, their very familiarity is
partly a byproduct of the assumptions I hoped to challenge &mdash;
assumptions that have perpetuated totalizing arguments about
&quot;print&quot; and &quot;the book.&quot; How could I defamiliarize a
history of reading and writing within such a prosaic academic literacy?</p>
 <table width="450px" style="position:absolute;bottom:5px;">
<tr>
<td width="33%">
<a href="intro7.html"><img src="left.PNG"></a>
</td>
<td width="34%">
<div style="position:absolute;right:auto;left:auto;width:auto;">
<div class="cube" style="left:10px;"></div>
<div class="cube" style="left:20px;"></div>
<div class="cube" style="left:30px;"></div>
<div class="cube" style="left:40px;"></div>
<div class="cube" style="left:50px;"></div>
<div class="cube" style="left:60px;"></div>
<div class="cube" style="left:70px;"></div>
<div class="cube" style="left:80px;background-color:yellow"></div>
<div class="cube" style="left:90px;"></div>
<div class="cube" style="left:100px;"></div>
<div class="cube" style="left:110px;"></div>
</div>
</td>
<td width="33%">
<a href="intro9.html"><img src="right.PNG"></a>
</td>
</tr>
</table>
</body>
</html>
```

```
<!--
___intro9.html___
by Whitney Anne Trettien

The ninth page of the introduction, which pops up in a Thickbox.
-->

<head>
 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
 <title>ARS COMBINATORIA, by whitney anne trettien</title>
 <link rel="stylesheet" type="text/css" href="style-intro.css" />
 </head>
 <body style="background-color:white;">
 <p>Thus instead of a linear text, I&#39;ve produced a digital mechanism
that, like the objects of my study, forces the reader to participate in the
process of making meaning. On the one hand, this medium allows me to present
a comparative history without compromising specificity or reducing the
complexity of one moment to a mere reflection of another; yet it still
strives for thematic cohesion by using our digital present quite literally as
a map for exploring programmatic epistemologies in our past. Like our current
media ecology, this map can be, in the words of many of my test users,
&quot;disorienting,&quot; a Borgesian textual labyrinth. I sympathize with
these frustrations. As students and scholars, we are not primed to
participate in reading texts as any more than &quot;critical
interpreters&quot; who absorb and repurpose language, and writing is still
presented as an act of &quot;originality.&quot; In other words, the practice
of cutting up and combining texts &mdash; that is, manipulating language
materially &mdash; is almost entirely absent from our current conceptual
model of literacy. Yet such forms of reading and writing are one facet to the
infinitely complex history of both the book and (if the recent avalanche of
literature on new media literacies is any indication) the book-to-come. By
both presenting and enacting the very mechanisms I theorize, I hope to put a
neglected past in conversation with our present while still waving
&quot;goodbye to much that is familiar.&quot;</p>
 <table width="450px" style="position:absolute;bottom:5px;">
<tr>
<td width="33%">
<a href="intro8.html"><img src="left.PNG"></a>
</td>
<td width="34%">
<div style="position:absolute;right:auto;left:auto;width:auto;">
<div class="cube" style="left:10px;"></div>
<div class="cube" style="left:20px;"></div>
<div class="cube" style="left:30px;"></div>
<div class="cube" style="left:40px;"></div>
<div class="cube" style="left:50px;"></div>
<div class="cube" style="left:60px;"></div>
<div class="cube" style="left:70px;"></div>
<div class="cube" style="left:80px;"></div>
<div class="cube" style="left:90px;background-color:yellow;"></div>
<div class="cube" style="left:100px;"></div>
<div class="cube" style="left:110px;"></div>
</div>
</td>
<td width="33%">
<a href="coda1.html"><img src="right.PNG"></a>
</td>
</tr>
```

```
</table>
</body>
</html>
```

```
<!--
___coda1.html___
by Whitney Anne Trettien

This is the first page of the coda to the introduction.
-->
<!doctype html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>ARS COMBINATORIA, by whitney anne trettien</title>
<link rel="stylesheet" type="text/css" href="style-intro.css" />
<script src="scripts/jquery-1.3.1.min.js" type="text/javascript"></script>
<script src="scripts/thickbox.js" type="text/javascript"></script>
</head>
<body>
<p>About halfway through my project &mdash; knee-deep in JQuery code, an
unfamiliar Other of a whole new order for this traditionally-trained
humanities student &mdash; I experienced a gestalt switch: that is, I no
longer &quot;knew&quot; theoretically but knew quite literally the social-,
historical- and institutional-embeddedness of media forms. There is nothing
like hours spent searching Javascript tutorials, hacker blogs and plug-in
libraries, clicking CTRL-U to peek beneath the hood of every page, to
understand that media are, in the words of Lisa Gitelman, &quot;socially
realized structures of communication, where structures include both
technological forms and their associated protocols, and where communication
is a cultural practice.&quot;<a href="#"><sup>12<span>Gitelman
7.</span></a></sup> In the same vein, it feels disingenuous &mdash; silly,
even &mdash; to wax poetical about the ease and facility, the &quot;speed and
light&quot; of digital media after cracking open the plastic casing on my
aging laptop to physically solder the power pin back to the motherboard in
order to avoid losing all of my work.<a href="#"><sup>13<span>See Matthew
Kirschenbaum on &quot;medial ideology&quot;: &quot;At the core of a medial
ideology of electronic text is the notion that in place of inscription,
mechanism, sweat of the brow (or its mechanical equivalent steam), and cramp
of the hand, there is light, reason, and energy unleashed in the electric
empyrean&quot; (Kirschenbaum 39; see 36ff).</span></a></sup> Even the shift
from the physical to the digital archive was instructive: after several days
spent in the rare book room at Yale&#39;s Beinecke Library, I ordered and
received a digital facsimile of a volvelle I was studying, only to find its
beautiful nest of interlocking paper wheels flattened to what looked like a
map or circle diagram, all of its tactility and movement lost in the
ostensibly more &quot;interactive&quot; screen.</p>
<table width="450px" style="position:absolute;bottom:5px;">
<tr>
<td width="33%">
<a href="intro9.html"><img src="left.PNG"></a>
</td>
<td width="34%">
<div style="position:absolute;right:auto;left:auto;width:auto;">
<div class="cube" style="left:10px;"></div>
<div class="cube" style="left:20px;"></div>
<div class="cube" style="left:30px;"></div>
<div class="cube" style="left:40px;"></div>
<div class="cube" style="left:50px;"></div>
<div class="cube" style="left:60px;"></div>
<div class="cube" style="left:70px;"></div>
<div class="cube" style="left:80px;"></div>
<div class="cube" style="left:90px;"></div>
```

```
<div class="cube" style="left:100px;background-color:yellow;"></div>
<div class="cube" style="left:110px;"></div>
</div>
</td>
<td width="33%">
<a href="coda2.html"><img src="right.PNG"></a>
</td>
</tr>
</table>

</body>
</html>
```

```
<!--
___coda2.html___
by Whitney Anne Trettien

This is the second page of the coda to the introduction.
-->

<!doctype html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>ARS COMBINATORIA, by whitney anne trettien</title>
<link rel="stylesheet" type="text/css" href="style-intro.css" />
<script src="scripts/jquery-1.3.1.min.js" type="text/javascript"></script>
<script src="scripts/thickbox.js" type="text/javascript"></script>
</head>
<body>
<p>To put it quite simply, the site you&#39;re reading now, like any media
form, does not &quot;do&quot; anything on its own. Rather, it was designed,
by me, to achieve certain functionalities within constraints largely outside
my control, including the standards of web design, screen widths, and the
affordances of the programming languages and markup structures I chose to
use.</p>
<p>In other words, this site does not nullify my authority as author, nor
does it seek to erase the technological restrictions, ideologies or ideals
that went into its production. Rather, like Mallarm&eacute;&#39;s Livre or
the text-generating mechanism I discuss herein, it is both a language machine
of the present and an imaginative, mediated representation of the past. Yet,
like the doughnut-shaped disk drive envisioned by Rabinow, it also returns us
to a future as-yet fulfilled &mdash; to the book as circle, as labyrinth, as
infinite hypertext.</p>
<table width="450px" style="position:absolute;bottom:5px;">
<tr>
<td width="33%">
<a href="coda1.html"><img src="left.PNG"></a>
</td>
<td width="34%">
<div style="position:absolute;right:auto;left:auto;width:auto;">
<div class="cube" style="left:10px;"></div>
<div class="cube" style="left:20px;"></div>
<div class="cube" style="left:30px;"></div>
<div class="cube" style="left:40px;"></div>
<div class="cube" style="left:50px;"></div>
<div class="cube" style="left:60px;"></div>
<div class="cube" style="left:70px;"></div>
<div class="cube" style="left:80px;"></div>
<div class="cube" style="left:90px;"></div>
<div class="cube" style="left:100px;"></div>
<div class="cube" style="left:110px;background-color:yellow"></div>
</div>
</td>
<td width="33%">
<a href="coda2.html"><img src="white.PNG"></a>
</td>
</tr>
</table>

</body>
</html>
```

```
/*
___text.php___
by Whitney Anne Trettien

This is the main text file, containing all of the thesis text except for the
introduction. It should be painfully obvious that "switch" is the only
function I know in PHP.
*/

<?php $cOption = $_GET['o'];
switch($cOption) {
case '1':
      echo '<p>Around 1650, Georg Philipp Harsd&ouml;rffer devised an
ingenious ballet.  It&#39;s simple: first, give each dancer a board inscribed
with a letter of the alphabet; then watch as new words or phrases  emerge
from dance. The very movement of the dancer&#39;s bodies will act as a
combinatory mechanism from which language springs (<a
href="bibliography.html#westerhoff"
target="_blank">Westerhoff<span>Westerhoff, Jan C. &quot;Poeta Calculans:
Harsd&ouml;rffer, Leibniz, and the <i>Mathesis Universalis</i>.&quot;
<i>Journal of the History of Ideas</i> 60.3 (1999): 449-67.</span></a>
465).</p><p>There is no evidence that Harsd&ouml;rffer ever produced such a
ballet.  Perhaps the first modern conceptual poet, Harsd&ouml;rffer&#39;s
genius lies in his ability to construct reality through  language &mdash;
that is, to use language not as a mirror for the world, but as its
fundamental building blocks.</p>';
      break;
case '2':
      echo '<p>Harsd&ouml;rffer used pieces of wood to make anagrams (<a
href="bibliography.html#schwenter"
target="_blank"><i>Delitiae</i><span>Harsd&ouml;rffer, Georg Philipp and
Daniel Schwenter. <i>Delici&aelig; Physico-Mathematic&aelig;, Oder, Mathemat.
Und Philosophische Erquickstunden...</i> N&uuml;rnberg: in Verlegung Jeremiae
D&uuml;mlers, 1651.</span></a> II.514), designed letter-dice to teach
children to build word combinations (<a
href="bibliography.html#Harsd&ouml;rffer-trichter"
target="_blank"><i>Trichter</i><span>Harsd&ouml;rffer, Georg Philipp.
<i>Poetischer Trichter; Die Teutsche Dicht- Und Reimkunst, Ohne Behuf Der
Lateinischen Sprache, in VI. Stunden Einzugiessen</i>. Hildesheim, New York:
G. Olms, 1971.</span></a> II.18; see <a href="bibliography.html#westerhoff"
target="_blank">Westerhoff<span>Westerhoff, Jan C. &quot;Poeta Calculans:
Harsd&ouml;rffer, Leibniz, and the <i>Mathesis Universalis</i>.&quot;
<i>Journal of the History of Ideas</i> 60.3 (1999): 449-67.</span></a> 464),
and  assigned numbers to letters to unlock a poem&#39;s hidden values (<a
href="bibliography.html#Harsd&ouml;rffer-trichter"
target="_blank"><i>Trichter</i><span>Harsd&ouml;rffer, Georg Philipp.
<i>Poetischer Trichter; Die Teutsche Dicht- Und Reimkunst, Ohne Behuf Der
Lateinischen Sprache, in VI. Stunden Einzugiessen</i>. Hildesheim, New York:
G. Olms, 1971.</span></a> II.26-9), earning him the title Der  Spielende, or
&#34;the Player,&#34; in the <i>Fruchtbringende Gesellschaft</i>. Each of
these games uses language not  as an abstraction, the purely rational product
of the mind, but as quite literally a material object to be manipulated and
moved, cut-up and combined.</p>';
      break;
case '3':
      echo '<blockquote><p class="blockquote">In this simple mechanization we
see how the ideas of the  combinatorial nature of the world on the one hand
and language on the other hand meet: letters are inscribed onto material
```

```
elements, so that a permutation of these out of itself generates an anagram
without requiring any ingenuity. For the  baroque poet this structural
isomorphism between language and matter explains why this method of anagram
production  works, while this fact itself serves as a further manifestation
of the very same isomorphism. Because language and matter  both follow
combinatorial principles, it is possible to mechanize the former by
inscribing it onto the latter. In the  &#34;language machine&#34; the
combinatorial features of language and the world which are generally distinct
are unified  into a single artifact. (<a href="bibliography.html#westerhoff"
target="_blank">Westerhoff<span>Westerhoff, Jan C. &quot;Poeta Calculans:
Harsd&ouml;rffer, Leibniz, and the <i>Mathesis Universalis</i>.&quot;
<i>Journal of the History of Ideas</i> 60.3 (1999): 449-67.</span></a> 464-
5)</p></blockquote>';
      break;
case '4':
      echo '<p>The <i>F&uuml;nffacher Denckring der Teutschen Sprache</i>, or
the Five-fold  Thought-ring of the German Language, is a database of German
words composed of five predicate variables:  prefixes (forty-eight values),
initial letters or diphthongs (fifty values), medial letters (twelve values),
final letters of  diphthongs (120 values) and suffixes (twenty-four values).
Instead of using a table structure, however, each variable is  inscribed
along the edge of a disc nested within each of the other discs, forming a
simple combinatory mechanism that can generate any information stored in its
circular, relational database.</p>';
      break;
case '5':
      echo '<blockquote><p class="blockquote"><a href="#">Will ich nun alle
Stammw&ouml;rter ordenlich finden / so fange ich bey  dem A de&szlig; zweyten
Ringes an / und drehe darzu das kleine a de&szlig; dritten Ringes:<span>I
want now to find all stem-words in an orderly manner, so I begin with the A
on the second Ring / and counterrotate the small a on the third
ring:</span></a><a href="#"> dann suche ich den vierten  Ring Aab / Aabb /
Aabd / &amp;c. blinde oder deutunglose W&ouml;rter / bi&szlig; auf das ch /
Aach / Aquisgranum, eine  benamte Stadt in Niderland / Aal / eines Fisches
und eines Schusters Werckzeug Namen / Aas (cadaver)&c.<span>then I seek the
fourth ring Aab / Aabb / Aabd / &c. / blind and meanlingless words / up to
the ch / Aach / Aquisgranum, named a city in the Netherlands / Aal / the name
of a fish and a tool of the cobbler / Aas (cadaver)&c.</span></a>  (<a
href="bibliography.html#schwenter"
target="_blank"><i>Delitiae</i><span>Harsd&ouml;rffer, Georg Philipp and
Daniel Schwenter. <i>Delici&aelig; Physico-Mathematic&aelig;, Oder, Mathemat.
Und Philosophische Erquickstunden...</i> N&uuml;rnberg: in Verlegung Jeremiae
D&uuml;mlers, 1651.</span></a> II. 516-7)</p></blockquote>';
      break;
case '6':
      echo '<p>By spinning the discs of the Denckring, the user can generate
up to 97,209,600  words &mdash; 300 times as many as are in the most complete
printed dictionary ever produced, the Oxford English  Dictionary.</p>';
      break;
case '7':
      echo '<p>Harsd&ouml;rffer claims that <a href="#">&#34;wird sich
verhoffentlich kein Wort in unsrer  gantzen Sprache finden / welches nicht
auf diesem Ring weisen seyn solte&#34;<span>hopefully [there is] no word
found in our entire language / which cannot be be shown on this
ring</span></a> (<a href="bibliography.html#schwenter"
target="_blank"><i>Delitiae</i><span>Harsd&ouml;rffer, Georg Philipp and
Daniel Schwenter. <i>Delici&aelig; Physico-Mathematic&aelig;, Oder, Mathemat.
Und Philosophische Erquickstunden...</i> N&uuml;rnberg: in Verlegung Jeremiae
```

```
D&uuml;mlers, 1651.</span></a> II.519), although of  course in reality the
rings are incomplete (<a href="bibliography.html#zeller">Zeller<span>Zeller,
Rosmarie. <i>Spiel Und Konversation Im Barock; Untersuchungen Zu
Harsd&ouml;rffers Gespr&auml;chsspielen.</i> New York: de Gruyter,
1974.</span></a> 167). In addition to representing <a href="#">&#34;die
gantze Teutsche Sprache auf  einem Bl&auml;tlein&#34;<span>the complete
German language on a small page</span></a> (<a
href="bibliography.html#schwenter"
target="_blank"><i>Delitiae</i><span>Harsd&ouml;rffer, Georg Philipp and
Daniel Schwenter. <i>Delici&aelig; Physico-Mathematic&aelig;, Oder, Mathemat.
Und Philosophische Erquickstunden...</i> N&uuml;rnberg: in Verlegung Jeremiae
D&uuml;mlers, 1651.</span></a> II.516), the Denckring also forms new words
that do not yet exist in German but,  because they are formed by combining
correct radicals and letters, could exist as legitimate words.</p>';
      break;
case '8':
      echo '<p>As Harsd&ouml;rffer explains, explicitly tying his device to
the linguistic  theories of Schottel,</p><blockquote><p class="blockquote"><a
href="#">Ist also dieses eine unfehlbare Richtigkeit / ein vollst&auml;ndiges
Teutsches  Wortbuch zu verfassen / und beharren wir in der Meinung /
da&szlig; alle solche zusammen gesetzte W&ouml;rter / welche ihre  Deutung
w&uuml;rken f&uuml;r gut Teutsch zul&auml;ssig / sonderlich in den
Gedichten<span>It is an infallible accuracy / to write a dictionary of all
German words / and we insist on the opinion / that all such words combined
together / which are denoted [to be] of good and proper German / are
particularly [useful] in poems</span></a><a href="#"> / ob sie gleich sonsten
nicht  gebr&auml;uchlich / wie hiervon zu lesen der umb unsere Sprache
wolverdiente Herr Schottelius in seiner Einleitung und in  seinen Lobreden
der Sprachk&uuml;nst vorgef&uuml;get.<span> / even if they are not very
common / as hereof Mr. Schottel orders our language to be read in his
introduction and chapters on the linguistic arts.</span></a> (<a
href="bibliography.html#schwenter"
target="_blank"><i>Delitiae</i><span>Harsd&ouml;rffer, Georg Philipp and
Daniel Schwenter. <i>Delici&aelig; Physico-Mathematic&aelig;, Oder, Mathemat.
Und Philosophische Erquickstunden...</i> N&uuml;rnberg: in Verlegung Jeremiae
D&uuml;mlers, 1651.</span></a> II.518).</p></blockquote>';
      break;
case '9':
      echo '<p>Because the Denckring materializes proper root-words and
mechanizes the  proper formula for combining them, any output it produces
must also be proper German. In other words, in this view  language is
fundamentally a material and mechanical phenomenon: material because it is
rooted in monosyllabic  stem-words that are themselves rooted in &#34;die
Teutschen letteren oder Buchstabe&#34; (<a href="bibliography.html#schottel-
haubtsprache" target="_blank">Schottel <i>HaubtSprache</i><span>Schottelius,
Justus Georg. <i>Ausfuehrliche Arbeit Von Der Teutschen Haubt Sprache, Worin
Enthalten Gemelter Dieser Haubt Sprache Uhrankunft, Uhraltertuhm,
Reinlichkeit, Eigenschaft, Vermoegen, Unvergleichlichkeit, Grundrichtigkeit,
Zumahl Die Sprach Kunst Und Vers Kunst Teutsch Und Guten Theils Lateinisch
Voellig Mit Eingebracht, Wie Nicht Weniger Die Verdoppelung, Ableitung, Die
Einleitung, Nahmwoerter, Authores Vom Teutschen Wesen Und Teutscher Sprache,
Von Der Verteutschung, Item Die Stammwoerter Der Teutschen Sprache Samt Der
Erklaerung Und Derogleichen Viel Merkwuerdige Sachen ... Ausgefertiget Von
Justo-Georgio Schottelio ..</i> Braunschweig: Gedrukt und verlegt durch C. F.
Zilligern, 1663.</span></a> 61.31), and  mechanical because meaning is not
computed rationally but through an absolute and abstracted algorithm. As
Newman  points out (writing more generally of Harsd&ouml;rffer&#39;s
linguistics), far from secularizing the vernacular, its physical
```

```
instantiation &#34;in textual form adds to its authority and testifies to the
reality of divinity present in the German  tongue&#34; (<a
href="bibliography.html#newman" target="_blank">Newman<span>Newman, Jane O.
<i>Pastoral Conventions: Poetry, Language, and Thought in Seventeenth-Century
Nuremberg.</i> Baltimore: Johns Hopkins University Press, 1990.</span></a>
97), bringing the reader closer to an originary <i>lingua adamica</i> that is
divinely, not merely  prescriptively, proper.</p>';
      break;
case '10':
      echo '<blockquote><p class="blockquote">It does not seem to bother
Harsd&ouml;rffer that it is a book, a  man-made, printed text, that he relies
upon for assurances about the divine capacities of the written word &mdash;
precisely the opposite. The availability of this knowledge in textual form
adds to its authority and testifies to the reality of  divinity present in
the German tongue. (<a href="bibliography.html#newman"
target="_blank">Newman<span>Newman, Jane O. <i>Pastoral Conventions: Poetry,
Language, and Thought in Seventeenth-Century  Nuremberg.</i> Baltimore: Johns
Hopkins University Press, 1990.</span></a> 97)</p></blockquote>';
      break;
case '11':
      echo '<p>On a practical level, Harsd&ouml;rffer envisions the Denckring
as part of his  poetic process. The device makes its first appearance in his
<i>Poetischer Trichter</i> or <i>Poet&#39;s Funnel</i>, a  book on different
theories and forms of verse and, as is written in the <i>Delitiae
Mathematicae et Physicae</i>, <a href="#">&#34;hat  ... seinen Gebrauch in
Erfindung der Reimw&ouml;rter / wann man die Reimsilben auf dem dritten und
vierten Ring suchet  und die Reimbuchstaben auf dem zweyten Ring darzu
drehet&#34;<span>has ... its use in the invention of rhyme-words / when one
seeks the rhymes on the third and fourth rings, and turns the rhyme-letters
of the second ring</span></a> (<a href="bibliography.html#schwenter"
target="_blank"><i>Delitiae</i><span>Harsd&ouml;rffer, Georg Philipp and
Daniel Schwenter. <i>Delici&aelig; Physico-Mathematic&aelig;, Oder, Mathemat.
Und Philosophische Erquickstunden...</i> N&uuml;rnberg: in Verlegung Jeremiae
D&uuml;mlers, 1651.</span></a> II.518). Similarly, unlike columnar tables
printed in books &mdash; indeed, if the Denckring were printed as a folio
dictionary, it would fill thousands of volumes &mdash; the  structure of
nesting rings allows the poet to identify words that match a particular rhyme
scheme, provides immediate  random access to the lexicon, and is mobile,
encouraging poets to incorporate it into poetic games and performance
pieces.</p>';
      break;
case '12':
      echo '<p>Perhaps most importantly, in using the Denckring the poet is
not locating an  item in a list, but dynamically generating meaning in the
moment: that is, the very motion of the poet&#39;s hand  assembles a whole
word out of dissembled parts, engaging the baroque spirit of <i>inventio</i>.
More than a mere storage  device, the Denckring is a writing implement,
mediating the relationship between the poet&#39;s physical body and the
generated text.</p>';
      break;
case '13':
      echo '<p>God assembled the world from thirty-two divine letters and
numbers; the poet  spins a wheel inscribed with trace elements of the
<i>lingua adamica</i> to generate 97,209,600 wor(l)ds, many of which  exist
only as possibilities. By embodying the theory of Stammw&ouml;rter, the
Denckring also embodies its contradictions,  turning the very institution
that legitimates it &mdash; that is, God, the divine &mdash; into a mere
mechanism that dumbly spits out  meaning. For if the poet has at her disposal
```

```
the material elements of creation and God&#39;s combinatory metric, what
what Book of Nature could be read that the poet couldn&#39;t already write
herself?</p>';
      break;
case '14':
      echo '<p>Combinatorial creation, anagrams, the notion of root elements
&mdash; these  ideas converge in the concept of Stammw&ouml;rter, or German
morphemes rooted in the originary language of  Adam.</p>';
      break;
case '15':
      echo '<p>One of the most well-known German linguists of the early
seventeenth century  and a proponent of Stammw&ouml;rter theory, Justus Georg
Schottel wrote a series of grammars promoting the antiquity and
expressiveness of German, calling for (like Opitz&#39;s <i>Aristarchus</i>) a
return to the pure, Adamic vernacular. The first,  <i>Teutsche
Sprachkunst</i> (1641), begins with a collection of ten &#34;Testimonia der
Gelarten von der Trefflichkeit der  deutschen Sprache,&#34; or testimonies
from the learned on the excellence of the German language. Just as the
kabbalist&#39;s Sefirot connects the ten digits to the the ten manifest
attributes of God, each of the ten testimonies in the  <i>Teutsche
Sprachkunst</i> expand upon a different quality of excellence inherent in the
German tongue, laying the  foundation for Schottel&#39;s own commentary on
German&#39;s combinatorial capacity.</p>';
      break;
case '16':
      echo '<p>A year after the publication of the <i>Teutsche
Sprachkunst</i>, Prince Ludwig  I of Anhalt-K&ouml;then invited Schottel to
join the <i>Fruchtbringende Gesellschaft</i>, where he was granted the title
<i>der  Suchende</i>, or &#34;the seeker,&#34; and the emblem of a cabbage.
Shortly after that, Schottel, who had been tutoring  Duke August the Younger
of Brunswick-L&uuml;neburg&#39;s young sons since 1638, was joined in the
Duke&#39;s household  by Sigmund von Birken, a prolific poet and friend to
Georg Philipp Harsd&ouml;rffer. In 1645 Schottel himself became the  tenth
member of Harsd&ouml;rffer&#39;s Pegnesische Blumenorden under the name
&#34;Fontano,&#34; the adjectival  version of the Latin word for
&#34;fountain&#34; or &#34;spring.&#34;</p>';
      break;
case '17':
      echo '<p>The security of royal patronage, friendships with imaginative
thinkers and  access to the most extensive library in Europe, the Duke&#39;s
Herzog August Library, helped Schottel weave together  his linguistic theory,
most fully developed in the massive <i>Ausf&uuml;hrliche Arbeit von der
teutschen Haubtsprache</i>  (1663). To historicize his native tongue,
Schottel traces the German language back to Ashkenaz, <a href="#">&#34;ein
Altvater der  Teutschen / hat mit sich die alte Celtische oder Teutsche
Sprache von Babel gebrach&#34;<span>the patriarch of the Germans / who
brought the old Celtic or German language with him from Babel</span></a> and
from Ashkenaz all the way  back to Adam (<a href="bibliography.html#schottel-
haubtsprache" target="_blank">Schottel <i>HaubtSprache</i><span>Schottelius,
Justus Georg. <i>Ausfuehrliche Arbeit Von Der Teutschen Haubt Sprache, Worin
Enthalten Gemelter Dieser Haubt Sprache Uhrankunft, Uhraltertuhm,
Reinlichkeit, Eigenschaft, Vermoegen, Unvergleichlichkeit, Grundrichtigkeit,
Zumahl Die Sprach Kunst Und Vers Kunst Teutsch Und Guten Theils Lateinisch
Voellig Mit Eingebracht, Wie Nicht Weniger Die Verdoppelung, Ableitung, Die
Einleitung, Nahmwoerter, Authores Vom Teutschen Wesen Und Teutscher Sprache,
Von Der Verteutschung, Item Die Stammwoerter Der Teutschen Sprache Samt Der
Erklaerung Und Derogleichen Viel Merkwuerdige Sachen ... Ausgefertiget Von
Justo-Georgio Schottelio ..</i> Braunschweig: Gedrukt und verlegt durch C. F.
```

```
Zilligern, 1663.</span></a> I.34). Thus, while the Confusion of Babel may
have muddied the <i>lingua  adamica</i>, it did not completely erase it: in
fact, German, Schottel claims, has its roots in the pure vernacular of the
Garden  of Eden.</p>';
      break;
case '18':
      echo '<blockquote><p class="blockquote"><a href="#">Dem allen nach wird
gewi&szlig;lich folgen da&szlig; / gleich wie  das jtzige Teutschland annoch
dasselbe Teutschland ist / welches vor etzlichen tausend Jahren gewesen / ob
es schon  jtzo `besser bebauet / herrlicher ausgeschm&uuml;kt<span>It follows
then accordingly that / just as the present Germany is the same Germany /
that existed one thousand years ago / although it is better developed and
more magnificently adorned</span></a><a href="#"> ... Also ist gleichfals
unsere jtzige Teutsche Sprache / eben dieselbe  uhralte weltweite Teutsche
Sprache<span>So is our German language currently / the same ancient,
universal German language</span></a><a href="#> ... denn wie das Land /
Teutschland bleibet / also m&uuml;ssen die Stammw&ouml;rter /  Teutsche
W&ouml;rter bleiben die denn ihre nat&uuml;rliche Eigenschaften (davon in
dieser Sprachkunst wird gehandelt) so  lange in sich / samt jhrer Deutung /
gehabt haben / so longe sie in rerum natura gewesen.<span>thus just as the
country / remains German / also must the stem-words / remain German words
that retain their natural properties (which are traded in this linguistic
theory) / and their meaning / as they have been rerum natura.</span></a> (<a
href="bibliography.html#schottel-haubtsprache" target="_blank">Schottel
<i>HaubtSprache</i><span>Schottelius, Justus Georg. <i>Ausfuehrliche Arbeit
Von Der Teutschen Haubt Sprache, Worin Enthalten Gemelter Dieser Haubt
Sprache Uhrankunft, Uhraltertuhm, Reinlichkeit, Eigenschaft, Vermoegen,
Unvergleichlichkeit, Grundrichtigkeit, Zumahl Die Sprach Kunst Und Vers Kunst
Teutsch Und Guten Theils Lateinisch Voellig Mit Eingebracht, Wie Nicht
Weniger Die Verdoppelung, Ableitung, Die Einleitung, Nahmwoerter, Authores
Vom Teutschen Wesen Und Teutscher Sprache, Von Der Verteutschung, Item Die
Stammwoerter Der Teutschen Sprache Samt Der Erklaerung Und Derogleichen Viel
Merkwuerdige Sachen ... Ausgefertiget Von Justo-Georgio Schottelio ..</i>
Braunschweig: Gedrukt und verlegt durch C. F. Zilligern, 1663.</span></a>
I.48)</p></blockquote>';
      break;
case '19':
      echo '<p><b>Rerum natura</b>, the nature of things: an allusion to
<i>De rerum  natura</i>, a poem by Lucretius circa the first century B.C. In
it, Lucretius posits an Epicurean, atomistic world of infinite  possible
combinations &mdash; of <a href="#">&#34;dissimiles ... formae glomeramen in
unum / conveniunt&#34;<span>globes dissimilar in form joined together as
one</span></a> (<a href="<a href="http://www.perseus.tufts.edu/cgi-
bin/ptext?doc=Perseus%3Atext%3A1999.02.0130;query=card%3D%2342;layout=;loc=2.
581" target="_blank">Lucretius<span>Lucretius, <U>De rerum
natura.</U></span></a> II.686-7).  For Lucretius and Epicurus before him,
each atom is a discrete unit of irreducible and self-evident meaning <a
href="#">&#34;quod  nusquam sine permitiali / discidio potis est seiungi
seque gregari, / pondus uti saxis, calor ignis, liquor aquai&#34;<span>since
on no occasion is one able to be separated from the group without a
destructive tearing apart, as mass from stone, heat from fire, or liquid from
water</span></a> (<a href="<a href="http://www.perseus.tufts.edu/cgi-
bin/ptext?doc=Perseus%3Atext%3A1999.02.0130;query=card%3D%2313;layout=;loc=1.
483" target="_blank">Lucretius<span>Lucretius, <U>De rerum
natura.</U></span></a> I.451-3).</p>';
      break;
case '20':
```

```
      echo '<p>Although not explicitly tied to the creation myth &mdash; in
fact, Lucretius  appears to explicitly reject Adamic theory (see <a
href="bibliography.html#eco-perfect" target="_blank">Eco<span>Eco, Umberto.
<U>The Search for the Perfect Language.</U> Cambridge, Mass.,: Blackwell,
1995.</span></a> 88-9) &mdash; Lucretius&#39; atomism provides the groundwork
for conceptualizing Schottel&#39;s Stammw&ouml;rter. Like atoms,
Stammw&ouml;rter are irreducible, isolatable  units of meaning, much as
linguists today might speak of morphemes. However, unlike morphemes
Stammw&ouml;rter are not  rational conceptions of the human mind, but are
inextricably interwoven with the material text. For Schottel they are like
letters, having size and weight:</p><blockquote><p class="blockquote"><a
href="#">Gleich wie aber die Teutschen letteren oder Buchstabe alle
einlautend<span>Just as the German letters or Buchstabe are all
shrunken</span></a><a href="#">  / eben also sind die ersten Surtzelen oder
die Stammw&ouml;rter der Teutschen Sprache gleichfalls einsilbig.<span> /
even also are the first root or stem-words of the German tongue likewise one
syllable</span></a> (<a href="bibliography.html#schottel-haubtsprache"
target="_blank">Schottel <i>HaubtSprache</i><span>Schottelius, Justus Georg.
<i>Ausfuehrliche Arbeit Von Der Teutschen Haubt Sprache, Worin Enthalten
Gemelter Dieser Haubt Sprache Uhrankunft, Uhraltertuhm, Reinlichkeit,
Eigenschaft, Vermoegen, Unvergleichlichkeit, Grundrichtigkeit, Zumahl Die
Sprach Kunst Und Vers Kunst Teutsch Und Guten Theils Lateinisch Voellig Mit
Eingebracht, Wie Nicht Weniger Die Verdoppelung, Ableitung, Die Einleitung,
Nahmwoerter, Authores Vom Teutschen Wesen Und Teutscher Sprache, Von Der
Verteutschung, Item Die Stammwoerter Der Teutschen Sprache Samt Der
Erklaerung Und Derogleichen Viel Merkwuerdige Sachen ... Ausgefertiget Von
Justo-Georgio Schottelio ..</i> Braunschweig: Gedrukt und verlegt durch C. F.
Zilligern, 1663.</span></a> I.61)</p></blockquote>';
      break;
case '21':
      echo '<p>Like atoms and letters, or like the primitive concepts of
Leibniz&#39;s <i>lingua  characteristica</i> (<a
href="bibliography.html#westerhoff"
target="_blank">Westerhoff<span>Westerhoff, Jan C. &quot;Poeta Calculans:
Harsd&ouml;rffer, Leibniz, and the <i>Mathesis Universalis</i>.&quot;
<i>Journal of the History of Ideas</i> 60.3 (1999): 449-67.</span></a> 459),
a root-word in Schottel&#39;s baroque Stammw&ouml;rter theory combines
(<i>gef&uuml;get</i>) with other words to form meaning,  <a
href="#">&#34;gewisser massen gebildet werden&#34;<span>forming a certain
bulk</span></a> (<a href="bibliography.html#schottel-haubtsprache"
target="_blank">Schottel <i>HaubtSprache</i><span>Schottelius, Justus Georg.
<i>Ausfuehrliche Arbeit Von Der Teutschen Haubt Sprache, Worin Enthalten
Gemelter Dieser Haubt Sprache Uhrankunft, Uhraltertuhm, Reinlichkeit,
Eigenschaft, Vermoegen, Unvergleichlichkeit, Grundrichtigkeit, Zumahl Die
Sprach Kunst Und Vers Kunst Teutsch Und Guten Theils Lateinisch Voellig Mit
Eingebracht, Wie Nicht Weniger Die Verdoppelung, Ableitung, Die Einleitung,
Nahmwoerter, Authores Vom Teutschen Wesen Und Teutscher Sprache, Von Der
Verteutschung, Item Die Stammwoerter Der Teutschen Sprache Samt Der
Erklaerung Und Derogleichen Viel Merkwuerdige Sachen ... Ausgefertiget Von
Justo-Georgio Schottelio ..</i> Braunschweig: Gedrukt und verlegt durch C. F.
Zilligern, 1663.</span></a> I.70). These combinations are possible  because
one radical &mdash; that is, one signifier &mdash; corresponds directly and
immediately to the concept it  signifies. This relationship is not one of
mirroring, in which two distinct entities reflect one another, but, to return
the  materialism inherent in Stammw&ouml;rter, of embodiment. Thus in some
sense a root word simply is the essence a  concept.</p>';
      break;
case '22':
```

```
      echo '<blockquote><p class="blockquote"><a href="#">Durch die
nat&uuml;rlich bekandte Unm&uuml;glichkeit ist es  schlecht unm&uuml;glich
eine leichtere / gr&uuml;ndlichere und wundersamere Art der Letteren oder
Buchstaben und  W&ouml;rter / als die Teutschen sind auszubringen: Sie sind
nicht allein einlaufend / die durch einen nat&uuml;rlichen  Zufall den
deh&ouml;rigen Laut veruhrsachen / sonderen ihr einstimmiger Laut ist so
wunderreich / und ihre  Zusammenstimmung so &uuml;berk&uuml;nstlich /
da&szlig; die Natur sich hierin v&oumlllig und aller dinges ausgearbeitet
hat.  Denn / ein jedes Ding / wie seine Eigenschaft und Wirkung ist / also
mu&szlig; es vermittelst unserer letteren / und krast  derer / also
zusammengef&uuml;gten Teutschen W&ouml;rter / aus eines wolredenden Munde
daher fliessen / und nicht anders als ob  es gegenw&auml;rtig da were / durch
des Zuh&oumlrers Sin und Herze dringen. Zum Exempel nehme einer nur diese
W&ouml;rter: Wasser fliessen / ges&auml;usel / sanft / stille ac wie
k&uuml;nstlich ist es / wie gleichsam wesentlich fleust  das Wasser mit
stillem Bes&auml;usel von unser Zungen?<span>By the naturally acquainted
impossibility, it is bad, impossible, to spread a lighter / deeper and more
miraculous art of letters or Buchstaben and words / than those in German:
they are not shrinking alone / but are caused by a natural Chance, a heard
sound / but the sound is more concordant than miraculous / and its together-
tune so overly artificial / that Nature herself has hereby composed total and
all things. Then / every thing / is as its property and effect / and so it
must by means of our letters / and their force / also bring the German words
together /so they flow from a well-spoken mouth / no different than if it
were present as / by the listener&#39;s mind and sinking in the heart. For
example take just one of these words: Water flows / purrs / soft / still as
it is imitated / as, as it were important, water flows with still purrs from
our tongues?</span></a> (<a href="bibliography.html#schottel-haubtsprache"
target="_blank">Schottel <i>HaubtSprache</i><span>Schottelius, Justus Georg.
<i>Ausfuehrliche Arbeit Von Der Teutschen Haubt Sprache, Worin Enthalten
Gemelter Dieser Haubt Sprache Uhrankunft, Uhraltertuhm, Reinlichkeit,
Eigenschaft, Vermoegen, Unvergleichlichkeit, Grundrichtigkeit, Zumahl Die
Sprach Kunst Und Vers Kunst Teutsch Und Guten Theils Lateinisch Voellig Mit
Eingebracht, Wie Nicht Weniger Die Verdoppelung, Ableitung, Die Einleitung,
Nahmwoerter, Authores Vom Teutschen Wesen Und Teutscher Sprache, Von Der
Verteutschung, Item Die Stammwoerter Der Teutschen Sprache Samt Der
Erklaerung Und Derogleichen Viel Merkwuerdige Sachen ... Ausgefertiget Von
Justo-Georgio Schottelio ..</i> Braunschweig: Gedrukt und verlegt durch C. F.
Zilligern, 1663.</span></a> I.59)<p></blockquote>';
      break;
case '23':
      echo '<p>Many baroque thinkers echo Schottel&#39;s admiration for the
onomatopoeic  qualities of the German tongue, taking it as proof of
German&#39;s excellence and divine origins.</p>';
      break;
case '24':
      echo '<p>In his <i>Frauenzimmer Gespr&auml;chspiele</i>,
Harsd&ouml;rffer argues that  the German language</p><blockquote><p
class="blockquote">Speaks in the languages of nature, quite perceptibly
expressing all its sounds. ... ; it roars like the lion, lows like the oxen,
snarls like the bear, bells like the stag, whinnies like the horse, hisses
like the snake ... On all those occasions in which nature gives things their
own sound, nature speaks in our own German tongue.  For this, many have
wished to assert that the first man, Adam, would not have been able to name
the birds and all the  other beasts of the fields in anything but our words,
since he expressed, in a manner conforming to their nature, each and  every
innate property and inherent sound; and thus it is not surprising that the
roots of the larger part of our words coincide  with the sacred language.
```

```
(Harsd&ouml;rffer, quoted in <a href="bibliography.html#eco-perfect"
target="_blank">Eco<span>Eco, Umberto. <i>The Search for the Perfect
Language.</i> Cambridge, Mass.,: Blackwell, 1995.</span></a>
99)</p></blockquote>';
      break;
case '25':
      echo '<p>Here Harsd&ouml;rffer brings the concept of Stammw&ouml;rter
full circle, back to the <i>lingua adamica</i>. From a finite number of
radicals, infinite meanings become possible, constructing entire linguistic
universes unknown to humans since the Confusion of Babel, perhaps even since
the Fall. The more pure these radicals are &mdash; that is, the more they
embody the divine properties of the universe, the Paths of Wisdom &mdash; the
closer their combinations come to God&#39;s creation. Thus finding and
excavating Stammw&ouml;rter in the German tongue not only legitimates the
vernacular, uplifting it from its state of corruption and neglect, but also
exalts its speakers by allowing them to enter a more direct, almost
prelapsarian communion with God.</p>';
      break;
case '26':
      echo '<p>Yet, ironically, by presenting Stammw&ouml;rter as a form of
onomatopoeia, Schottel and Harsd&ouml;rffer undermine the very theory they
are attempting to support. Once linguistic perfection is no longer anchored
in the moment of Adamic naming, any language, divinely formed or artificial,
may become more perfect simply by molding its sounds to the &#34;language of
nature.&#34; Likewise, if humans can excavate the <i>lingua adamica</i> from
an ordinary vernacular, then they possess the basic elements of creation, the
building blocks from which all meaning constructed, and hence the power to
re-construct the world in their own image &mdash; to build a new Tower of
Babel.</p>';
      break;
case '27':
      echo '<p>Although Schottel takes pains to historicize his project, to
present it as a descriptive account of the German tongue, in the end it is
both prescriptivist and interventionist, wresting the power of the divine and
placing it in the hands of man.</p>';
      break;
case '28':
      echo '<p>The first half of the seventeenth century was a period of
transition, when scholars and poets and writers and publishers were
struggling to develop new standards for the changing practices of art, and
for the emerging culture of experimentation in natural philosophy (see, for
instance, <a href="bibliography.html" target="_blank">Johns<span>Johns,
Adrian. <U>The Nature of the Book: Print and Knowledge in the Making.</U>
Chicago, IL: University of Chicago Press, 1998.</span></a>). Of course, the
baroque era was also a period of spiritual and social upheaval, as bitter
religious wars ravaged central Europe. One must expect that the linguistic
theories of Schottel and Harsd&ouml;rffer would reflect these competitions
between the mechanical and the divine, and natural and the artificial.</p>';
      break;
case '29':
      echo '<p>As Jane Newman points out in a clever deconstruction of New
Historicism, history is not simply the &#34;background, &#39;origin,&#39; and
ultimate goal of textual expression,&#34; and texts are not merely
&#34;mirrors of historical conditions &#39;on the outside&#39;&#34; (<a
href="bibliography.html#newman" target="_blank">Newman<span>Newman, Jane O.
<i>Pastoral Conventions: Poetry, Language, and Thought in Seventeenth-Century
Nuremberg.</i> Baltimore: Johns Hopkins University Press, 1990.</span></a> 7,
10). In fact very few linguistic texts of the German baroque
```

```
&#34;&#39;refer&#39; or &#39;respond&#39; to an &#39;outside&#39;&#34;
history at all (<a href="bibliography.html#newman"
target="_blank">Newman<span>Newman, Jane O. <i>Pastoral Conventions: Poetry,
Language, and Thought in Seventeenth-Century  Nuremberg.</i> Baltimore: Johns
Hopkins University Press, 1990.</span></a> 10). Rather, Newman argues, texts
are institutions, which &#34;means, above all, understanding them as
historical events in a maximally concrete sense&#34;: as &#34;a material
phenomenon that comes into existence at a given moment and testifies to that
moment&#34; by both &#34;separating itself (as text) from it and by embedding
it in a rhetoric of transcendence&#34; (<a href="bibliography.html#newman"
target="_blank">Newman<span>Newman, Jane O. <i>Pastoral Conventions: Poetry,
Language, and Thought in Seventeenth-Century  Nuremberg.</i> Baltimore: Johns
Hopkins University Press, 1990.</span></a> 20).</p>';
      break;
case '30':
      echo '<blockquote><p class="blockquote">To study the institutional
identity of a text as well as the function of textual institutions relies,
then, on analyzing both the inaugural function of the text, its power to be
foundational and to provide a historical beginning, and its monumental
function, the strategies whereby it not only testifies to that beginning but
also transcends it by serving as a fixed model for specific future (perhaps
not only textual) behavior. These strategies gain their authority by
strategically &#39;forgetting&#39; and then recoding in a
&#39;monumental&#39; textual form the historicity of the inaugural moment as
a moment beyond and outside history. (<a href="bibliography.html#newman"
target="_blank">Newman<span>Newman, Jane O. <i>Pastoral Conventions: Poetry,
Language, and Thought in Seventeenth-Century  Nuremberg.</i> Baltimore: Johns
Hopkins University Press, 1990.</span></a> 22)</p></blockquote>';
      break;
case '31':
      echo '<p>Schottel&#39;s theory is both monument and foundation &mdash;
it both historicizes the vernacular and de-centers the very Judeo-Christian
narrative that legitimizes it. In short, like Stammw&ouml;rter themselves, it
both reflects and transcends its moment of creation.</p>';
      break;
case '32':
      echo '<blockquote><p class="blockquote">The great lesson that Kabbalah
can teach contemporary interpretation is that meaning in belated texts is
always wandering meaning, even as the belated Jews were a wandering people.
Meaning wanders, like human tribulation, or like error, from text to text,
and within a text, from figure to figure. What governs this wandering, this
errancy, is defense, the beautiful necessity of defense. For not just
interpretation is defense, but meaning itself is defense, and so meaning
wanders to protect itself. In its etymology, &#39;defense&#39; refers to
&#39;things forbidden&#39; and to &#39;prohibition&#39;, and we can speculate
that poetic defense rises in close alliance with the notions of trespass and
transgression, crucial for the self-presentation of any new strong poet. (<a
href="bibliography.html#bloom" target="_blank">Bloom<span>Bloom, Harold.
<i>Kabbalah and Criticism.</i> New York: Seabury Press, 1975.</span></a>
82)</p></blockquote>';
      break;
case '33':
      echo '<p>As Gershom Scholem puts it, the <i>Sefer Yetzirah</i>, or
<i>Book of Creation</i>, is &#34;small in size but enormous in influence&#34;
(<a href="bibliography.html#scholem-kabbalah"
target="_blank">Scholem<span>Scholem, Gershom Gerhard. <i>Kabbalah.</i> New
York: Quadrangle/New York Times  Book Co, 1974.</span></a> 23). Written
sometime around the eighth century, it is the earliest extant text in the
```

```
Kabbalist tradition, laying out a cosmogeny of combination and permutation.
</p>';
      break;
case '34':
      echo '<blockquote><p class="blockquote">With 32 mystical paths of
Wisdom<br/>engraved Yah<br/>     the Lord of Hosts<br/>     the God of
Israel<br/>the living God<br/>     King of the universe<br/>El Shaddai<br/>
Merciful and Gracious<br/>     High and Exalted<br/>     Dwelling in
Eternity<br/>     Whose name is Holy &mdash; <br/>          He is lofty and
holy &mdash; <br/>And He created His universe<br/>     with three books
(Sepharim), <br/>          with text (Sepher) <br/>          with number
(Sephar) <br/>          and with communication (Sippur). (<a
href="bibliography.html#kaplan" target="_blank">Sepher Yetzirah<span>Kaplan,
Aryeh, ed. <i>Sefer Yetzirah</i>. York Beach, Me.: S. Weiser,
1997.</span></a> 1.1)</p></blockquote>';
      break;
case '35':
      echo '<p>In Kabbalah, the thirty-two paths created by God come from 1)
the ten digits, futher manifest in the Ten Sefirot, or emanations of God, and
2) the twenty-two letters of the Hebrew alphabet, which come together through
231 &#34;gates,&#34; clusters of letters that form the roots of the Hebrew
verb (<a href="bibliography.html#scholem-kabbalah"
target="_blank">Scholem<span>Scholem, Gershom Gerhard. <i>Kabbalah.</i> New
York: Quadrangle/New York Times  Book Co, 1974.</span></a> 25). &#34;He hath
formed, weighed and composed with these twenty-two letters every created
thing,&#34; the <i>Sefir Yetzirah</i> reads, &#34;and the form of everything
which shall hereafter be&#34; (<a href="bibliography.html#kaplan"
target="_blank">Sepher Yetzirah<span>Kaplan, Aryeh, ed. <i>Sefer
Yetzirah</i>. York Beach, Me.: S. Weiser, 1997.</span></a> II.2). </p>';
      break;
case '36':
      echo '<p>Aryeh Kaplan explains the basic principles of Kabbalah, which
left a deep impression on seventeenth-century European
thinkers:</p><blockquote><p class="blockquote">The letters and digits are the
basis of the most basic ingredients of creation, quality and quantity. The
qualities of any given thing can be described by words formed out of letters,
while all of its associated quantities can be expressed by numbers. (<a
href="bibliography.html#kaplan" target="_blank">Kaplan<span>Kaplan, Aryeh,
ed. <i>Sefer Yetzirah</i>. York Beach, Me.: S. Weiser, 1997.</span></a>
5)</p></blockquote>';
      break;
case '37':
      echo '<p>Much like the Christian &#34;Book of Nature,&#34; which sees
God&#39;s creation as an all-encompassing text coextensive with the Bible,
the <i>Sefir Yetzirah</i> of the Jewish Kabbalah treats the world as
fundamentally textual, a spatial environment to be &#34;read.&#34; However,
unlike the &#34;Book of Nature,&#34; Kabbalism is interested not only in
reading the world as the end-product of God&#39;s work, but in (re)producing
the mechanisms of creation to unlock new meaning. Thus for the Kabbalist
&#34;the world-process is essentially a linguistic one, based on the
unlimited combinations of letters&#34; to form new spiritual environments (<a
href="bibliography.html#scholem-kabbalah"
target="_blank">Scholem<span>Scholem, Gershom Gerhard. <i>Kabbalah.</i> New
York: Quadrangle/New York Times  Book Co, 1974.</span></a> 1974 25).</p>';
      break;
case '38':
      echo '<blockquote><p class="blockquote">We are dealing with a treasure-
hoard of the second degree, one that refers to the notations of nature, which
```

```
in their turn indicate obscurely the pure gold of things themselves. The
truth of all these marks &mdash; whether they are woven into nature itself or
whether they exist in lines on parchments and in libraries &mdash; is
everywhere the same: coeval with the institution of God. (<a
href="bibliography.html#foucault" target="_blank">Foucault<span>Foucault,
Michel. <i>The Order of Things: An Archaeology of the Human Sciences.</i> New
York:  Pantheon Books, 1970.</span></a> 34)</p></blockquote>';
      break;
case '39':
      echo '<p>Kabbalists extend the idea of manipulation, combination and
permutation to all sacred texts, employing a set of three basic hermeneutic
tools to uncover the secret meanings embedded in the rolls of the Torah. The
first, <b>Temurah</b>, involves rearranging words and sentences to generate
new texts from the base elements of the old, much like anagrams.
<b>Notarikon</b> similarly cobbles together a new text from the old, but
using only the first and last letter of a series of words. And, combining the
thirty-two letters and ten digits of creation, <b>Gematria</b> assigns
numerical values to letters which then acquire quantitative
significance.</p>';
      break;
case '40':
      echo '<p>Underlying each of these practices is the belief that letters,
written language, exist as material objects with a primarily spatial, rather
than temporal, existence. Even the voice has a physical presence in the Sefir
Yetzirah: it is &#34;impressed upon the air, and audibly modified in five
places; in the throat, in the mouth, by the tongue, through the teeth, and by
the lips&#34; (<a href="bibliography.html#kaplan" target="_blank">Sepher
Yetzirah<span>Kaplan, Aryeh, ed. <i>Sefer Yetzirah</i>. York Beach, Me.: S.
Weiser, 1997.</span></a> II.3). As a physical entity, then, Scripture is not
strictly linear but contains multiple dimensions, even extralinguistic
dimensions, unlocked by cutting and anagrammatically permuting the
text.</p>';
      break;
case '41':
      echo '<p>In many ways this work you&#39;re reading now is a
Ka(nni)bbalistic reading of its source texts &mdash; a golem, a monstrous
creation that lives dispite the inanimate and artificial nature of its
constituent parts.</p>';
      break;
case '42':
      echo '<blockquote><p class="blockquote">Meaning, whether in modern
poetry or in Kabbalah, wanders wherever anteriority threatens to take over
the whole map of misreading, or the verbal universe, if that phrase be
preferrred. Meaning swerves, enlarges oppositely, vacates, drives down so as
to rise up again, goes outside in the wan hope of getting itself more on the
inside, and at last attempts to reverse anteriority by forsaking the evasions
of mental space for those of mental time. A poem&#39;s images or Kabbalistic
hypostases are thus types of ambivalence (not of ambiguity) that cope with
the burden of anteriority. (<a href="bibliography.html#bloom"
target="_blank">Bloom<span>Bloom, Harold. <i>Kabbalah and Criticism.</i> New
York: Seabury Press, 1975.</span></a> 89)</p></blockquote>';
      break;
case '43':
      echo '<p>Pieter Brueghel the Elder&#39;s <i>Tower of Babel</i> (1563)
is a golem. Unlike the tall, toddering towers of Meister der Weltenchronik
or, later, Athanasius Kircher, Brueghel&#39;s creation is a thick mound,
slowly spiraling out of the sea, consuming the surrounding
countryside.</p><p>On the one hand, the Tower teams with human life: tiny
```

```
figures crawl through its crevices, hanging from machinery, as others observe
its construction from the hillside. Yet, larger than any biological form, it
seems so distinctly <i>in</i>human &mdash; an inanimate monster composed of
decaying parts.</p>';
      break;
case '44':
      echo '<p>Part Ziggurat and part Colosseum, Brueghel&#39;s <i>Tower</i>
embodies the concept of recursion &mdash; a fundamental principal in both the
language of humans and code, the language of computers. Each level is
composed of a series of nesting blocks that emanate from the center, forming
a set of concentric wheels that point up toward heaven.</p>';
      break;
case '45':
      echo '<p>Yet the blocks are not entirely planar. By constructing the
elements on a slight angle, the tower&#39;s blocked, circular design and
rounded Roman arches are at tension with its spiraling shape, causing it to
crumble from the inside. Indeed, although it stretches into the clouds, it is
clear the tower can never reach heaven; already its carefully-cut blocks are
crumbling to dust, almost as they are added to the wall. As the king stands
beside his eviscerated tower, inspecting the unused bricks strewn over the
hillside, the painting illuminates the stark difference between design and
implementation, between the dream of unity and the disparate reality of its
parts.</p>';
      break;
case '46':
      echo '<p>While today it is common for philosophers and linguists to
conceive of language as a system of rationally-understood syntactical
relationships between larger units of meaning, such as words or whole
sentences (see, for instance, <a href="bibliography.html"
target="_blank">Chomsky<span>Chomsky, Noam. <U>Cartesian Linguistics: A
Chapter in the History of Rationalist Thought.</U> New York: Harper and Row,
1966.</span></a>), the Cartesian revolution in linguistics did not occur
until the latter half of the seventeenth century. During the first half, the
letter still dominated linguistic theory as the primary unit of sound and
sense. </p> ';
      break;
case '4611':
      echo '<p>In Nuremberg during 1658, the Czech education reformer John
Amos Comenius first published his <i>Orbis Sensualium Pictus</i>, or
&quot;The Visible World in Pictures,&quot; an encyclopedia of images designed
to teach children the alphabet and vocabulary. Originating from the same
theories of the <i>lingua adamica</i> as Harsd&ouml;rffer&#39;s Denckring,
Comenius&#39;s textbook &mdash; most likely known to G. W. Leibniz and
Quirinus Kuhlmann as children &mdash; was enormously popular in Germany and
went into 244 editions between 1658 and 1964 (<a href="bibliography.html"
target="_blank">Crain<span>Crain, Patricia. <U>The Story of A: The
Alphabetization of America from <i>The New England Primer</i> to <i>The
Scarlet Letter</i>.</U> Stanford, CA: Stanford University Press,
2000.</span></a> 27).</p> ';
      break;
case '4622':
      echo '<p>Comenius does not depict the alphabet for mere mnemonic
purposes, but as a way of naturalizing its structure. In other words, the
<i>Orbis Pictus</i> renders the alphabet&#39;s arbitrary arrangement and
strange relationship with sound as, in fact, a reflection of the natural,
divine order of the world. As Crain writes,</p><blockquote><p
class="blockquote">The difference, in this scheme, between nature and
artifice, between what the animals do and what people do, is taxonomic;
```

```
Comenian man organizes nature. Adam&#39;s task, naming the animals, really
amounted to putting them in their place; in the Comenian Eden this amounts to
ranging the creatures in alphabetical as well as generic order. (<a
href="bibliography.html" target="_blank">Crain<span>Crain, Patricia. <U>The
Story of A: The Alphabetization of America from <i>The New England Primer</i>
to <i>The Scarlet Letter</i>.</U> Stanford, CA: Stanford University Press,
2000.</span></a> 36)</p></blockquote>';
      break;
case '4633':
      echo '<p>If, for Comenius, the alphabet represents the divine natural
order, then, learning the alphabet becomes a process of learning God&#39;s
creation. As Cohen writes, in the <i>Orbis Pictus</i> &quot;the proper
acquisition of language must logically lead to the conversion of the
world,&quot; since &quot;proper language learning would lead to brotherhood,
grammar to God&quot; (<a href="bibliography.html"
target="_blank">Cohen<span>Cohen, Murray. <U>Sensible Words: Linguistic
Practice in England, 1640-1785.</U> Baltimore: Johns Hopkins University
Press, 1977.</span></a> 19). Thus as with Harsd&ouml;rffer&#39;s Denckring,
to read a word simply is to write &mdash; and right &mdash; the universe, as
the material manipulation of a finite set (the 26 letters) unlocks the
infinite multitude of God&#39;s creation.</p>';
      break;
case '4644':
      echo '<p>Yet if, for Comenius, learning the alphabet equates with
learning the order of the universe, then any literate person becomes, in some
sense, privy to God&#39;s secrets. Describing the episteme of recursive,
symbolic reflections during the sixteenth and seventeenth centuries in
Europe, Foucault writes: &quot;To know must therefore be to interpret: to
find a way from the visible mark to that which is being said by it and which,
without that mark, would lie like unspoken speech, dormant within
things&quot; (<a href="bibliography.html#foucault"
target="_blank">Foucault<span>Foucault, Michel. <u>The Order of Things: An
Archaeology of the Human Sciences.</u> New York:  Pantheon Books,
1970.</span></a> 32). Likewise, the process of reading a text within this
system also writes its physical equivalent into existence, as the site of
literacy enables both the production and consumption of a universal
language.</p> ';
      break;
case '4655':
      echo '<blockquote><p class="blockquote">The verbal and visual tropes
that surround the alphabet cloak the fact that the unit of textual meaning
&mdash; the letter &mdash; lacks meaning itself. The alphabet&#39;s semantic
vacuum represents a threat to orthodoxy, for into this space competing
meaning systems may rush. The features of the alphabet&#39;s mask change over
time, drawn as they are from a cultural moment&#39;s fund of meaningfulness.
Where a culture goes to make sense of itself is where the alphabet too
scavenges. While one might be used to thinking in an archeological fashion
about excavating layers of meaning, the student of the alphabetic text digs
down only to reveal a null. It is the content of those layers that yields
interest. (<a href="bibliography.html" target="_blank">Crain<span>Crain,
Patricia. <U>The Story of A: The Alphabetization of America from <i>The New
England Primer</i> to <i>The Scarlet Letter</i>.</U> Stanford, CA: Stanford
University Press, 2000.</span></a> 18)</p></blockquote>';
      break;
case '47':
      echo '<blockquote><p class="blockquote">Language was envisioned as an
aggregate of discrete signs, each denoting a mental image, with in turn
mirrored a natural world of separate physical objects. As the syntactical
```

```
relations between signs could not be easily visualized, they tended to be
neglected. (<a href="bibliography.html#hudson"
target="_blank">Hudson<span>Hudson, Nicholas. <i>Writing and European
Thought, 1600-1830.</i> Cambridge ; New York, NY, USA:  Cambridge University
Press, 1994.</span></a> 43)</p></blockquote>';
      break;
case '48':
      echo '<p>Like many other baroque thinkers, Harsd&ouml;rffer identified
the material form of letters in objects of the world &mdash; or, rather,
found the world in the materiality of letters. In a visual alphabet from the
<i>Delitiae mathematicae et physicae &ndash; Der mathematischen und
philosophischen Erquickstunden, Dritter Theil</i> (1653, 44-5),
Harsd&ouml;rffer plays with signifying nature of <i>Buchstaben</i>, lending
his writing (as Juliet Fleming puts it, describing early modern graffiti)
&#34;meaning in excess both of its signified content, and of its easily
recognized aesthetic dimensions &mdash; a meaning that has to do with the
fact of its appearance in matter&#34; (<a href="bibliography.html#fleming"
target="_blank">Fleming<span>Fleming, Juliet. <i>Graffiti and the Writing
Arts of Early Modern England.</i> Philadelphia: University of  Pennsylvania
Press, 2001.</span></a> 115).</p>';
      break;
case '49':
      echo '<blockquote><p class="blockquote">If words represent reality to
our understanding, then, like existing things, they must consist of assembled
elements. Reducing language to physical properties visually organized on the
page justifies operating on words as objects. If the written and spoken
language can be touched, tabulated, and visualized, then it can be secured,
improved, perfected and rationally taught. (<a href="bibliography.html#cohen"
target="_blank">Cohen<span>Cohen, Murray. <i>Sensible Words: Linguistic
Practice in England, 1640-1785.</i> Baltimore: Johns  Hopkins University
Press, 1977.</span></a> 8)</p></blockquote>';
      break;
case '53':
      echo '<p>The connection between the Denckring and the hand is strong.
Four hands populate the edges of the mechanism, positioned as if they were
the reader&#39;s own. The hands in the upper left and lower right hold
instruments &mdash; a compass and a quill, respectively &mdash; while the
other two display symbols, a heavy ring and a small wreath.</p>';
      break;
case '54':
      echo '<p>The hands holding instruments represent writing as a material,
artifactual practice mediated by instruments, tools and technologies, linking
it visually to the emerging practice of experimental philosophy. In fact, the
<i>Delici&aelig; Physico-Mathematic&aelig;</i> (one of the texts in which the
Denckring appears) is peppered with illustrations of disembodied hands
holding tools to measure, slice, or organize matter &mdash; instruments that,
like written language, help us to understand the surrounding world by
rendering it in human terms.</p>';
      break;
case '55':
      echo '<p>The hands in the upper right and lower left corners of the
Denckring depict language as symbolic &mdash; an atemporal monument that
materializes and therefore preserves the abstract. In Western culture, a
wreath represents both eternity and death, the fate of language upon
inscription, while the hand in the top right corner gestures outward, palm
facing up, over the word &#34;unvergessen,&#34; or
&#34;unforgotten.&#34;</p>';
      break;
```

```
case '5511':
      echo '<p>Thus two hands around the Denckring illustrate the transient,
performative act of writing &mdash; the quill sits poised, as if preparing to
set pen to paper &mdash; while the other two symbolize its material product:
the written text.</p><p>As an abstracted form known to all humans, then, the
hand becomes a monument &mdash; a structure, much like Harsd&ouml;rffer&#39;s
Denckring, that stores data indefinitely. That is, until the human hand comes
to retrieve it through the gesture of writing.</p>';
      break;
case '5522':
      echo '<p>Hands are often found around volvelles in early printed books,
usually in the form of &#34;pointers.&#34; For example, a woodcut calendar
from 1466 calculates the date of easter, as well as the golden number of any
given year, by spinning a set of nesting wheels against each other. The
innermost wheel depicts an angel pointing with hand and foot at a golden
number (<a href="bibliography.html" target="_blank">Sherman<span>Sherman,
Claire Richter et al. <U>Writing on Hands: Memory and Knowledge in Early
Modern Europe.</u> Chicago, IL: University of Chicago Press, 2002.</span></a>
164-5, Cat. 41). Similarly, Giambattista della Porta&#39;s popular
cryptography manual contains several decoder dials &mdash; volvelles used to
decode encrypted texts mechanically &mdash; which feature God&#39;s pointing
finger in the center, poised to decrypt the surrounding symbols (<a
href="bibliography.html" target="_blank">Schmidt<span>Schmidt, Suzanna Karr.
&quot;Constructions both Sacred and Profane: Serpents, Angels and Pointing
Fingers in Renaissance Books with Moving Parts.&quot;</span></a> <a
href="http://www.robertsabuda.com/everythingpopup/suzannekarr.asp"
target="_blank">online</a>).</p>';
      break;
case '56':
      echo '<blockquote><p class="blockquote">There is then a history of
technology that is also the history of &#34;man,&#34; the
programmed/programming machine: the human written. The human cannot simply be
returned to the divine/oral origin; the hand is there from the start, as the
locus of retroactive redetermination. Thus, from the start, the written being
and the writing being are coincident and differential, opening and enclosing
at one and the same time interiority and exteriority, the human and the
technological, the mind and the body, speech and writing in their narrow
sense; from the start, the relationships between these and all other
differential terms exist within the possibility of protention and retention
and the differential possibilties of rhythms of emergence. (<a
href="bibliography.html#goldberg" target="_blank">Goldberg<span>Goldberg,
Jonathan. <i>Writing Matter: From the Hands of the English Renaissance.</i>
Stanford,  Calif.: Stanford University Press, 1990.</span></a>
24)</p></blockquote>';
      break;
case '5611':
      echo '<p>In fact, in the visual culture of early modern books, the
image of the hand often straddles the boundary between performance and
product. Sign languages like George Dalgarno&#39;s <i>A Manual Alphabet</i>
(1680) and John Bulwer&#39;s <i>Chirologia or the Naturall Language of the
Hand</i> (1644) link gestures with the production of speech, treating hand
motions as a form of universal language, while earlier finger-reckoning
systems, such as Luca Pacioli&#39;s, use digits of bone and sinew to
calculate digits of numerical abstraction. Finger alphabets also proliferated
during the late sixteenth century, facilitating memory through the rehearsed
gestures of individual letters and even, in Giambattista della Porta&#39;s
cryptology manual <i>De furtivis literarum notis</i> (1563), encrypting texts
through a gestural performance that links touched body parts to individual
```

```
letters (see <a href="bibliography.html"
target="_blank">Sherman<span>Sherman, Claire Richter et al. <U>Writing on
Hands: Memory and Knowledge in Early Modern Europe.</u> Chicago, IL:
University of Chicago Press, 2002.</span></a> 186).</p>';
      break;
case '5622':
      echo '<p>Even as the hand performs language, it is itself a surface for
writing during the early modern period. In thoughtful and extensive exhibit
on images of inscribed hands in early printed books, Claire Richter Sherman
points to the many ways that marks on the hand operated as a form of memory
palace, helping humans &#34;to understand, order and recall abstract
concepts&#34; through &#34;iconic metaphors, bodily mnemonics, and cognitive
maps encompassing processes of association, memory, and recollection&#34; (<a
href="bibliography.html" target="_blank">Sherman<span>Sherman, Claire Richter
et al. <U>Writing on Hands: Memory and Knowledge in Early Modern Europe.</u>
Chicago, IL: University of Chicago Press, 2002.</span></a> 13; see Cat. 56,
64).</p>';
      break;
case '5633':
      echo '<p>In Stephan Fridolin&#39;s <i>Schatzbehalter</i> (1491), a
series of woodcuts by Michael Wohlgemut (Albrecht D&uuml;rer&#39;s
instructor) depict the human hand as a database storing a set of 50 roman
numerals, each value linking back to a meditation in the text. In addition to
using the hand as a storage device, Fridolin encourages his readers to cut
out large paper hands and tag them with keywords that will help the penitent
call up associated text and images (see <a href="bibliography.html"
target="_blank">Sherman<span>Sherman, Claire Richter et al. <U>Writing on
Hands: Memory and Knowledge in Early Modern Europe.</u> Chicago, IL:
University of Chicago Press, 2002.</span></a>- 66). Thus the hand becomes a
technology of inscription that not only triggers the reader&#39;s memory, but
actually forces the reader to compose her own texts through a meditative form
of cut-ups and recombinations.</p>';
      break;
case '5644':
      echo '<p>Manicules &mdash; the &#34;pointing hands&#34; that annotate
the margins of so many books across the ages &mdash; slide along the same
sharp edge of performance and monument, motion and storage. As Clanchy points
out, these hand-drawn symbols might be seen as a form of <i>ars
memorativa</i>, &#34;a simple way of facilitating the retrieval of
information&#34; (<a href="bibliography.html"
target="_blank">Clanchy<span>Clanchy, M. T. <U>From Memory to Written Record:
England 1066-1307.</U> Cambridge, MA: Harvard University Press,
1979.</span></a> 172). Yet, as Will Sherman argues, the manicule also
&#34;has a gestural function that extends beyond its straightforwardly
indexical one&#34; (<a href="bibliography.html"
target="_blank">Sherman<span>Sherman, William H. <U>Used Books: Marking
Readers in Renaissance England.</U> Philadelphia, PA: University of
Pennsylvania Press, 1008.</span></a> 51). Operating within a dense network of
inscription, the manicule originates in a moment of reading that becomes the
act of writing a hand that points at, and thereby &#34;stores,&#34; a slice
of text. Like Harsd&ouml;rffer&#39;s Denckring, like Burroughs&#39; cut-ups,
the manicule forces us to ride the rim between reading and writing, between
textual consumption and textual production.</p>';
      break;
case '57':
      echo '<p>Harsd&ouml;rffer&#39;s Denckring collapses the boundaries
between speaking, reading and writing, presenting the totality of a language
that nonetheless does not exist outside the combinatory potential of its
```

```
mechanism. Roland Barthes asserts that the &#34;text&#34; &mdash; an
authorless, &#34;irreducible ... plural,&#34; an &#34;explosion, a
dissemination&#34; &mdash; is a relatively recent phenomenon compared to the
&#34;work,&#34; which is a authored and pedigreed piece of Literature (see <a
href="bibliography.html" target="_blank">Barthes<span>Barthes, Roland.
&quote;From Work to Text.&quot; <U>Textual Strategies.</u> Ed. J Harari.
Ithaca, NY: Cornell University Press, 1979.</span></a>), 73ñ81); but in fact
the Denckring shows that the ergodic text is an important part of baroque
culture. In Harsd&ouml;rffer&#39;s poetry generator, the author-function is
algorithmic and abstract, constructing a structure or code which the reader
then must assemble through the motion of the hand &mdash; a motion connected
with that of writing, of measuring.</p>';
      break;
case '58':
      echo '<blockquote><p class="blockquote">Oral and written are
derivative, then, from the handedness of the human. As Raymond Williams says,
there are &#34;relationships embodied in writing.&#34; For Derrida, that is
literally true; the writing being is also being written, whether at the
instinctive level or while writing at the computer. In the relationship
between writing in the general and in the specific sense &mdash; in
relationships which are not merely at the order of conceptuality, but which
are also social and historical &mdash; the field of a differential
inscription of human being comes into view. (<a
href="bibliography.html#goldberg" target="_blank">Goldberg<span>Goldberg,
Jonathan. <i>Writing Matter: From the Hands of the English Renaissance.</i>
Stanford,  Calif.: Stanford University Press, 1990.</span></a>
24)</p></blockquote>';
      break;
case '59':
      echo '<p>The sixteenth and early seventeenth centuries find central
Europe weak and disorganized. Around three hundred different princes,
prelates, counts and knights ruled over a loosely-affiliated smattering of
states, provinces, towns and fiefdoms, all grouped under the central
governance of the Holy Roman Empire. Foreign leaders such as the King of
Spain held fiefs within the Burgundian Circle, while German rulers held
foreign territories, causing them to be both directly under and outside the
authority of the Holy Roman Empire. Emperor Charles V himself, a Habsburg,
ruled the Spanish realms of Castile and Aragon and often appealed to the King
of Spain to help suppress internal divisions &mdash; an alliance which
threatened the independence of the German princes, who retaliated by
cementing their own alliances with the King of France, the Emperor&#39;s
enemy.</p>';
      break;
case '60':
      echo '<p>Religious conflict further strained the already tenuous ties
between local rulers and the Emperor. In 1555, after a series of Protestant
uprisings in the first half of the sixteenth century, Emperor Ferdinand I,
brother and successor to Charles V, signed a peace treaty with the
Schmalkaldic League, a group of Lutheran princes, dukes and rulers. The
treaty, called the Peace of Augsburg, officially (though not effectively)
ended the religious conflict by the principle of <i>cuius regio, eius
religio</i>, or &#34;whose region, his religion&#34;: that is, whatever
religion the leader of the territory believed in, whether Catholic or
Protestant, so must all subjects believe.</p>';
      break;
case '61':
      echo '<p>Although intended to curb violence, in fact, by cementing
religious divisions through a policy of mutual intolerance, the Augsburg
```

```
Peace agreement ironically destroyed any lingering sense of unity amongst the scattered states of the Holy Roman Empire and further weakened the authority of state leaders; for, as C. V. Wedgwood writes, the Lutheran princes and dukes &#34;were demanding from the Emperor what they refused to their own people&#34; (<a href="bibiography.html#wedgwood" target="_blank">Wedgwood<span>Wedgwood, C. V. <i>The Thirty Years War.</i> New York: New York Review Books, 2005.</span></a> 46). Those who refused to convert to the state&#39;s religion were forced to emigrate, leading to large populations of refugees spreading of disease and famine across an already unstable region. By 1618, tensions were so high that when Ferdinand II, another Habsburg and a staunch Catholic, became King of the largely Protestant Bohemia, the nobility threw his representatives out a window, starting a religious war that would last the next thirty years.</p>';
      break;
case '62':
      echo '<p>As Ferdinand&#39;s imperial governors were falling (quite luckily) onto a pile of manure in Prague, Martin Opitz, a young Protestant student from Lower Silesia, was traveling north along the Oder toward Frankfurt an der Oder, a mid-size trading town nestled in the heart of Brandenburg. When Opitz arrived in 1618 to study law, the state of Brandenburg, like most of the Empire, was cut deep with religious divisions. Although resistant to the Reformation in the early decades of the sixteenth century, its rulers had since &#34;jerked from Lutheranism to Calvinism and back, blazing a trail of dispossession, exile and violence&#34; that threatened to destroy what little religious harmony remained within the state (<a href="bibiography.html#wedgwood" target="_blank">Wedgwood<span>Wedgwood, C. V. <i>The Thirty Years War.</i> New York: New York Review Books, 2005.</span></a> 46).</p>';
      break;
case '63':
      echo '<blockquote><p class="blockquote">In Brandenburg the Elector declared that he would rather burn his only University than allow one Calvinist doctrine to appear in it. Nevertheless his successor became a Calvinist and introduced a pastor at Berlin, whereat the Lutheran mob broke into the newcomer&#39;s house and plundered it so effectively that he had to preach on the following Good Friday in a bright green undergarment, which was all that the rioters had left him. (<a href="bibliography.html#wedgwood" target="_blank">Wedgwood<span>Wedgwood, C. V. <i>The Thirty Years War.</i> New York: New York Review Books, 2005.</span></a> 47)</p></blockquote>';
      break;
case '64':
      echo '<p>It was in this environment that young Opitz entered the University of Frankfurt an der Oder, where he published his first essay, a polemic on behalf of the German language entitled <i>Aristarchus; sive, de contemptu linguae Teutonicae</i>.</p>';
      break;
case '65':
      echo '<p><a href="#">&#34;Quotiescunq; majores nostros Germanos, viros fortes ac invictos, cogito,&#34;<span>Whenever I think about our native German ancestors, strong men and indeed invincible</span></a> Opitz begins, <a href="#">&#34;religione quadam tacita ac horrore ingenti percellor&#34;<span>I am knocked down by a certain quiet religiousness and a powerful reverent awe</span></a>; for the German tongue is &#34;a charming tongue, a decent tongue, a serious tongue&#34; well-suited to alexandrine and Neoclassical verse. Yet, to express the artistic potential of his &#34;charming&#34; native tongue, Opitz chooses Latin, the &#34;corrupt&#34; language of Catholicism, switching occasionally to German when citing his own
```

```
original poetry. Thus reflecting the culture of the Holy Roman Empire itself,
the essay begins by presenting columns of clean, Roman typeface &mdash;</p>';
      break;
case '66':
      echo '<p>&mdash;- but then slowly becomes colonized by pockets of
Gothic font that playfully bursts across the page in expressive curls:</p>';
      break;
case '67':
      echo '<p>While the essay and even the lines of the pages attempt to
contain these Gothic outbursts through an imposed aesthetic harmony, the
forced unity only leads to greater discord as the exiled, dispossessed tongue
attempts to reclaim, even to re-colonize, the page. In fact, on this page
Latin is quite literally pushed to the edge, squashed between the enclosing
ranks of German.</p>';
      break;
case '68':
      echo '<p>Thus not only does the text reflect the kind of political
battles being fought at the time, it actually becomes a soldier in the war,
re-drawing territorial boundaries. Seen from this angle, <i>Aristarchus</i>
does not ironically reinforce the need to use Latin but actually undermines
it, forcing Latin toward the margins as German stakes a claim to its own
space.</p><p>The verses may be Neoclassical in form but, as Opitz makes
clear, they belong to German.</p>';
      break;
case '100':
      echo '<p>In a text-generating volvelle, space is a spiral, a nest, a
knot. Spinning the wheel unrolls a series of combinations, much like the
mesmerizing list of As, Cs, Ds and Gs produced by DNA sequencers. Yet the
disc itself consumes only a small portion of the page, perhaps a few inches
square. By condensing the art of combination into an interlocking mechanism,
volvelles produce the seemingly infinite from a finite space.</p><p>By
contrast, the cut-up method dismembers the text. Hands crack back the
book&#39;s spine, ripping the sinews of binding; scissors slice the page and
words fall, disfigured, from the text&#39;s corpus. Unlike the tightly-wound
spiral spinning imagined clouds of combinations, cut-ups expose the
text&#39;s spread to violence; they explode the thick brick of the book into
ashen wisps of words.</p>';
      break;
case '101':
      echo '<p>Like the four hands floating around Harsd&ouml;rffer&#39;s
Denckring &mdash; like manicules &mdash; like the reader&#39;s hand spinning
the discs &mdash; like the finger that points and clicks &mdash; the hand
will cut up the hand that recombines. The coherent abstraction of a textual
icon, dismembered from its body, reconnects through the process of an
embodied hand that moves, spinning a web of combinations.</p><p>As William
Burroughs wrote, cut-ups are experimental &mdash; &quot;experimental in the
sense of being something to do.&quot;</p>';
      break;
case '102':
      echo '<p>When Thomas Jefferson took scissors to his Bible in the Spring
of 1804 &mdash; three years into his Presidency and less than twenty years
after the federation of the United States of America &mdash; he was
profoundly dissatisfied with corrupted, anti-democratic state of the
Christian religion. &quot;In extracting the pure principles which [Jesus]
taught,&quot; he later wrote to his friend John Adams, &quot;we should have
to strip off the artificial vestments in which they have been muffled by
priests, who have travestied them into various forms, as instruments of
riches and power to them&quot; (<a href="bibliography.html"
```

```
target="_blank">Jefferson<span>Jefferson, Thomas. <U>Jefferson&#39;s Extracts
from the Gospels: <i>The Philosophy of Jesus</i> and <i>The Life and Morals
of Jesus</i>.</u> Ed. Dickinson W. Adams. Princeton, NJ: Princeton University
Press, 1983.</span></a> 352).</p>';
      break;
case '103':
      echo '<blockquote><p class="blockquote">We must reduce our volume to
the simple evangelists, select, even from them, the very words only of Jesus,
paring off the Amphibologisms into which they have been led by forgetting
often, or not understanding, what had fallen from him, by giving their own
misconceptions as his dicta, and expressing unintelligibly for others what
they had not understood themselves. There will be found remaining the most
sublime and benevolent code of morals which has ever been offered to man. I
have performed this operation for my own use, by cutting verse by verse out
of the printed book, and arranging, the matter which is evidently
[Jesus&#39;], and which is as easily distinguishable as diamonds in a
dunghill. (<a href="bibliography.html"
target="_blank">Jefferson<span>Jefferson, Thomas. <U>Jefferson&#39;s Extracts
from the Gospels: <i>The Philosophy of Jesus</i> and <i>The Life and Morals
of Jesus</i>.</u> Ed. Dickinson W. Adams. Princeton, NJ: Princeton University
Press, 1983.</span></a> 352)</p></blockquote>';
      break;
case '104':
      echo '<p>To purify the Word of God, Jefferson ordered two clean copies
of the New Testament, then proceeded to dissect each corpus, extracting only
those verses that spoke in the &quot;true&quot; voice of Jesus (<a
href="bibliography.html" target="_blank">Sheridan<span>Sheridan, Eugene R.
&quot;Introduction&quot; <U>Jefferson&#39;s Extracts from the Gospels: <i>The
Philosophy of Jesus</i> and <i>The Life and Morals of Jesus</i>.</u> Ed.
Dickinson W. Adams. Princeton, NJ: Princeton University Press,
1983.</span></a> 27) &mdash; verses, as Jefferson put it, &quot;as easily
distinguishable as diamonds in a dunghill&quot; (<a href="bibliography.html"
target="_blank">Jefferson<span>Jefferson, Thomas. <U>Jefferson&#39;s Extracts
from the Gospels: <i>The Philosophy of Jesus</i> and <i>The Life and Morals
of Jesus</i>.</u> Ed. Dickinson W. Adams. Princeton, NJ: Princeton University
Press, 1983.</span></a> 352). As he wrote, &quot;the result is an 8vo. of 46.
pages of pure and unsophisticated doctrines, such as were professed and acted
on by the <i>unlettered</i> apostles, the Apostolic fathers, and the
Christians of the 1st. century&quot; (<a href="bibliography.html"
target="_blank">Jefferson<span>Jefferson, Thomas. <U>Jefferson&#39;s Extracts
from the Gospels: <i>The Philosophy of Jesus</i> and <i>The Life and Morals
of Jesus</i>.</u> Ed. Dickinson W. Adams. Princeton, NJ: Princeton University
Press, 1983.</span></a> 352).</p><p>Jefferson titled his work <i>The
Philosophy of Jesus of Nazareth, extracted from the account of his life and
doctrines as given by Mathew, Mark, Luke, & John. being an abidgement of the
New Testament for the use of the Indians unembarrassed with matters of fact
or faith beyond the level of their comprehension.</i></p>';
      break;
case '105':
      echo '<p>Thomas Jefferson took his scissors to a set of Bibles for a
second time around 1820, nearly seventeen years after completing <i>The
Philosophy of Jesus</i>. The result was a multilingual reconstruction
entitled <i>The Life and Morals of Jesus of Nazareth Extracted textually from
the Gospels in Greek, Latin, French &amp; English.</i></p><p>Each spread in
the book contains four columns, one for each language, and includes passages
pertaining not only to Jesus&#39; teachings (as with <i>The Philosophy of
Jesus</i>) but also to his life. As Sheridan points out, whereas Jefferson
cut up and recombined his first Bible &quot;in response to his personal
```

```
religious needs and his concern with the problem of maintaining social
harmony in a republication nation,&quot; his second &quot;was strictly a
product of his private search for religious truth&quot; (<a
href="bibliography.html" target="_blank">Sheridan<span>Sheridan, Eugene R.
&quot;Introduction&quot; <U>Jefferson&#39;s Extracts from the Gospels: <i>The
Philosophy of Jesus</i> and <i>The Life and Morals of Jesus</i>.</u> Ed.
Dickinson W. Adams. Princeton, NJ: Princeton University Press,
1983.</span></a> 30).</p>';
      break;
case '106':
      echo '<p>I have used the language of disfigurement and dissection to
describe the cut-up method, as if the body of the book must remain an
inviolable, unimpregnated closure of text. The cut-up, I have implied, is a
kind of Frankenstein, a Golem, a Tower of Babel that sucks life from its
elements as soon as their attacker dislocates and recombines them into
something frighteningly, unnaturally new. </p><p>Yet it is clear that
Jefferson and the women of Little Gidding saw their scissors as swords of
peace, not pillage. His cut-ups did not, he thought, disfigure the Word but
clipped away the blemishes that had accumulated, plucking diamonds from the
dung. In short, the selective recombination of the old gave birth to a
primary, not secondary, text &mdash; a New Son brought to wash the Wor(l)d of
its sins.</p>';
      break;
case '107':
      echo '<p>For Jefferson, this new text was, in the words of Burroughs, a
poetry &quot;for everyone!&quot; (<a href="bibliography.html"
target="_blank">Burroughs<span>Burroughs, William. &quot;The Cut Up
Method.&quot; <U>The Moderns: An Anthology of New Writing in America.</U Ed.
Leroi Jones. New York: Corinth Books, 1963. 345-8.</span></a> 346).</p>';
      break;
case '108':
      echo '<blockquote><p class="blockquote">Nicholas Ferrar&#39;s response
to the increasing acrimony and division in political and religious life
[during the last sixteenth century] was not to join in the fray but to create
an alternative that he hoped would stand outside and rise above that discord.
(<a href="bibliography.html" target="_blank">Ransome<span>Ransome, Joyce.
&quot;Monotessaron: The Harmonies of Little Gidding.&quot; <U>The Seventeenth
Century</U 20.1 (April 2005): 22-52.</span></a> 22)</p></blockquote>';
      break;
case '109':
      echo '<p>The cut-up method is an accident.</p>';
      break;
case '110':
      echo '<p>In 1626 &mdash; the same year England crowned the
controversial King Charles I &mdash; Nicholas Ferrar moved his family from
London to Little Gidding, nestled just north of Cambridge. There, as
religious wars were being waged both at home and abroad, Ferrar&#39;s family
prayed, lived a strict routine of self-reliance and, what concerns us here,
constructed &quot;Harmonies&quot; &mdash; cut up and reassembled religious
texts and images.</p>';
      break;
case '111':
      echo '<p>Like sewing, weaving and other forms of text/ile production,
assembling the &quot;Harmonies&quot; was women&#39;s hand-i-work at Little
Gidding (<a href="bibliography.html" target="_blank">Dyck<span>Dyck, Paul.
&quot;&#39;So rare a use&#39;: Scissors, Reading and Devotion at Little
Gidding.&quot; <U>George Herbert Journal</U> 27.1&2 (2003/2004): 67-
81.</span></a> 68). The patriarch Nicholas directed his nieces on the form,
```

```
structure and verses to compile; then the women extracted the relevant
sections from the text, laid them out on the table, and recombined them to
form a new narrative.</p><p>Thus with appropriately-phallic &amp;quot;Knives &
Cizers&amp;quot; in hand, these nameless nieces tore into the body of the Bible
&amp;mdash; the &amp;quot;Heads&amp;quot; &amp;mdash; to breathe life into a new text. Like
strands of DNA splicing in the womb, discrete elements must disassemble
themselves before reassembling into a new organism &amp;mdash; a creature
paradoxically the same and yet wholly different from its parents.</p>';
      break;
case '112':
      echo '<p>As Dyck points out, the source texts often contained marginal
commentary, so that &amp;quot;the Little Gidding books literally sunder much of
the biblical textual markup ... from the textual apparatus that gives it
meaning&amp;quot; (Dyck &amp;quot;Marking the Bible&amp;quot; 2-3). Ensconced in a
bedding of commentary, words are woven into an intellectual tradition,
acquiring sense through abstract syntactical relationships in the same way
individual threads form an image when woven into a tapestry. Sliced from the
page, though, they become material objects, suddenly strange &amp;mdash; all form
and no content.</p>';
      break;
case '113':
      echo '<p>Although seventeenth-century women were not often given their
own pens and bid write, they were at times given their own scissors and bid
<i>cut, paste</i>. Disallowed from forming their own, they fed off the words
of others.</p>';
      break;
case '114':
      echo '<p>Biblical harmonies were widely known during the sixteenth and
seventeenth centuries. In England, for instance, Thomas Middleton, James
Bentley, Hugh Broughton, John Huid, Henry Garthwait, John Lightfoot and
William Gould all produced some variant of a &amp;quot;Harmony&amp;quot; that
attempted to reconcile the books of the Bible into one coherent narrative,
thereby participating in a long tradition of biblical exegesis that extends
beyond the scope of the present work.</p><p>What is fascinating about the
Little Gidding &amp;quot;Harmonies,&amp;quot; then, is not their drive to narrativize
(to harmonize) religious teachings, but the cut-and-paste methods they
employ.</p>';
      break;
case '115':
      echo '<p>Like Ramon Llull&amp;#39;s ars magna &amp;mdash; itself a form of
biblical exegesis &amp;mdash; each of the four Gospels in the Little Gidding
books is color-coded and labeled with a letter. Similarly, variations in type
and interspersed images keep each chunk of text separate from the
others.</p><p>Thus whereas most other Harmonies homogenize a re-ordered
narrative into a singular new printing, the Little Gidding books maintain the
unique character of each disparate element, facilitating not only a form of
writing, but a cut-and-paste form of reading. Through visual homogeneity, the
Harmony&amp;#39;s user is empowered to combine different texts and images in new
ways &amp;mdash; that is, to read against the forward-facing grain of a typical
printed book.</p>';
      break;
case '116':
      echo '<p>Yet this process does not individualize the reader (as we
might imagine) but actually links her to a long tradition of
&amp;quot;harmonizing&amp;quot; the Bible through selective reading. Although we,
conditioned by the hat tricks of Tristan Tzara and the radical literary
experiments of William Burroughs, theorize cut-ups and combinations as a way
to &amp;quot;lose the undesired past, to cut [our] way out of an old identity, if
```

```
not out of identity itself&amp;quot; (<a href="bibliography.html"
target="_blank">Harris<span>Harris, Oliver. <U>William Burroughs and the
Secret of Fascination.</u> Carbondale: Southern Illinois University Press,
2003.</span></a> 8), the Little Gidding readers used the Harmonies to
construct a desired past in the present, connecting their isolated commune to
a transhistorical community of readers. To individualize the text was to
participate in a shared religious practice. In fact, this collective form of
reading cut-ups was rooted in the routine of the commune. As Stewart points
out, the Harmonies were read aloud three times daily at Little Gidding in a
special room called the Concordance Room (<a href="bibliography.html"
target="_blank">Stewart<span>Stewart, Stanley. <U>George Herbert.</U> Boston:
Twayne Publisher, 1986.</span></a> 66).</p>';
      break;
case '117':
      echo '<p>The Bolognese naturalist Ulisse Aldrovandi (1522-1605)
collected nature, accruing over 7,000 diverse specimens throughout his
life.</p><p>Francesca Fontana, Aldrovandi&amp;#39;s <i>uxor dilectissima</i> (as
he called her), collected notes. She cut and collated pieces of
Aldrovandi&amp;#39;s manuscripts; compiled scraps from his library for his
<i>Pandechion Epistemonicon</i>; and upon his death, edited and published his
writings (<a href="bibliograph.html" target="_blank">Blair<span>Blair, Ann.
&amp;quot;Reading Strategies for Coping with Information Overload ca.1550-
1700.&amp;quot; <U>Journal of the History of Ideas</U> 64.1 (2003): 11-
28.</span></a> 26-7, <a href="bibliography.html"
target="_blank">Findlen<span>Findlen, Paula. &amp;quot;Masculine Prerogatives:
Gender, Space, and Knowledge in the Early Modern Museum.&amp;quot; <U>The
Architecture of Science.</U> Ed. Peter Louis Galison, Emily Ann Thompson.
Cambridge, MA: MIT Press, 1999.</span></a> 44-5). By remediating and remixing
Aldrovandi&amp;#39;s objects, Fontana rendered his cabinet of curiosities &amp;mdash;
his thesaurus, his treasure-hoard &amp;mdash; as text, then quite literally
positioned these scraps within the pages of the existing world of
print.</p><p>While Aldrovandi brought the world to Bologna, his wife,
wielding scissors, brought his words to the world.</p>';
      break;
case '118':
      echo '<blockquote><p class="blockquote">As much as this arrangement [of
the Harmonies] makes possible a reading of one combined account, it also
foregrounds the overlap, or the imbrication, of the texts, drawing attention
not to monolithic unity, but to harmonic difference; the book suggests a
unity that is necessarily expressed exactly through difference. The
arrangement suggests distinct kinds of text and thus of reading: a liturgical
and immediate story of Christ that invites a receptive response, and a
scholarly and mediated story of Christ that exposes the multiple versions
that constitute it, and that invites a critical awareness of the relations
between the versions. While naming the differences between these kinds of
reading is helpful, one must also note how tightly the two relate:
receptivity and inquiry are mutually constituting conditions of the devoted
reader. (<a href="bibliography.html" target="_blank">Dyck<span>Dyck, Paul.
&amp;quot;&amp;#39;So rare a use&amp;#39;: Scissors, Reading and Devotion at Little
Gidding.&amp;quot; <U>George Herbert Journal</U> 27.1&amp;2 (2003/2004): 67-
81.</span></a> 70)</p></blockquote>';
      break;
case '119':
      echo '<p>It is perhaps not surprising that Nicholas Ferrar, founder of
the Little Gidding community, was good friends with the visual poet George
Herbert. Throughout his poetry, Herbert cuts, copies and recombines disparate
verses and phrases to create not a set of individual, self-sufficient poems,
```

```
but a network of verse that, like the Harmonies, is dependent on patterns,
iconography and echoes of the Scripture.</p>';
      break;
case '11911':

      echo '<p>As Stewart writes,</p></p><blockquote><p class="blockquote">it
is a mistake to overlook the way in which these lines [from The Temple]
&quot;bring in&quot; others in the collection. As Herbert conceives the
interpretation of Scripture, so he presents his own lines as part of a
composite expression, which in turn reflects a continuing process of the
speaker&#39;s struggle to grasp the hidden meaning (&quot;Such are thy
secrets&quot;) of God&#39;s Word and, so, his intentions for one particular
&quot;Christian.&quot; This process transforms shards of experience that seem
unrelated and confused into a new and refreshing recognition or
&quot;story.&quot; (<a href="bibliography.html"
target="_blank">Stewart<span>Stewart, Stanley. <U>George Herbert.</U> Boston:
Twayne Publisher, 1986.</span></a> 67)</p></blockquote>';
      break;
case '120':
      echo '<p>In other words, both the Little Gidding &quot;Harmonies&quot;
and Herbert&#39;s pattern explore the sometimes slippery juncture between
written language and its material medium, reflecting a faithful
interpretation of scripture through typography and the spatial layout of the
page. Or, to borrow Katherine Hayles&#39; term, both the Harmonies and
Herbert&#39;s poems operate through material metaphor, &quot;foreground[ing]
the traffic between words and physical artifacts&quot; (Hayles 22).</p>';
      break;
case '12011':
      echo '<blockquote><p class="blockquote">When a literary work
interrogates the inscription technology that produces it, it mobilizes
reflexive loops between its imaginative world and the material apparatus
embodying that creation as a physical presence. Not all literary works make
this move, of course, but even for those that do not, my claim is that <i>the
physical form of the literary artifact always affects what the words (and
other semiotic components) mean</i>. Literary works that strengthen,
foreground, an thematize the connections between themselves as material
artifacts and the imaginative realm of verbal/semiotic signifiers they
instantiate open a window on the larger connections that unite literature as
a verbal art to its material forms. (<a href="bibliography.html"
target="_blank">Hayles<span>Hayles, N. Katherine. <u>Writing Machines.</u>
Cambridge, Mass.: MIT Press, 2002.</span></a> 25)</p></blockquote>';
      break;
case '121':
      echo '<p>In 1634, the women of Little Gidding compiled a Harmony for
King Charles I from several copies of the New Testament, two copies of Henry
Garthwait&#39;s gospel harmony <i>Montoessaron</i> (1634), pages from a folio
Bible and an assembly of biblical illustrations (<a href="bibliography.html"
target="_blank">Dyck  and Williams<span>Dyck, Paul and Stuart Williams.
&quot;Toward an Electronic Edition of an Early Modern Assembled
Book.&quot;<U>CHWP</U> A.44 (July 2008).</span></a> <a
href="http://www.chass.utoronto.ca/epc/chwp/CHC2007/Dyck_Williams/Dyck_Willia
ms.htm" target="_blank">online</a>). As John Ferrar recorded after his
brother Nicholas&#39;s death, King Charles I considered the book to be
&quot;a rare jewel&quot; (<a href="bibliography.html" target="_blank">Dyck
and Williams<span>Dyck, Paul and Stuart Williams. &quot;Toward an Electronic
Edition of an Early Modern Assembled Book.&quot;<U>CHWP</U> A.44 (July
2008).</span></a> <a
href="http://www.chass.utoronto.ca/epc/chwp/CHC2007/Dyck_Williams/Dyck_Willia
```

ms.htm" target="_blank">online). Thus the same King who sparked anger by marrying a Catholic princess; who would soon be embroiled in a bitter civil war with the Long Parliament; and who, within twenty years of receiving the Little Gidding Harmony, would lose his head to the fervently anti-Catholic Oliver Cromwell found solace in a text reassembled from the extracted parts of books both Protestant and Catholic. From discord, the Little Gidding women constructed a Harmony. ';
	break;
case '122':
	echo '<blockquote> N.F. having first spent Some time in the contrivance of the Work (wch was comonly an hour every Day) and having given his Nieces directions How & in what Manner they should do it, They with their Cizers cut out of each Evangelist such & such Verses, & layd them together, to make & perfect such & such a Head, or chapter, which when they had first roughly done, then with their Knives & Cizers they neatly fitted each Verse So cutt out, to be pasted downe upon sheets of Paper, & So artifically they performed this new-found-out-way, as it were a new kind of Printing: For all that saw the Bookes when they were done, tooke them to be printed on ye ordinary Way, So finely were ye verses joyned together and with great Presses for that purpose pressed down upon ye white sheets of paper. (quoted in DyckDyck, Paul. "'So rare a use': Scissors, Reading and Devotion at Little Gidding." <U>George Herbert Journal</U> 27.1&2 (2003/2004): 67-81. 69) </blockquote>';
	break;
case '123':
	echo ' After the Civil War, Sir John Gibson (1606-1665) — a wealthy Royalist who had been unwaveringly loyal to King Charles I, even guarding York during its siege in 1644 — found himself imprisoned at Durham Castle by 1653, unable to repay his debts. There, with his own life in tatters, he began cutting and pasting a multimedia, multidimensional commonplace book. ';
	break;
case '124':
	echo ' Adam Smyth describes Sir John Gibson's manuscript as "breathless, chaotic, fragmented," suggesting "little sense of coherence; even less of a linear order" (SmythSmyth, Adam. "'Rend and teare in peeces': Textual Fragmentation in Seventeenth-Century England." <U>The Seventeenth Century</U> 19.1 (april 2004): 36-52. 38). Handwritten sententiae surround emblematic images; pre-Christian poems and devotional texts mingle; and cuttings are interleaved throughout the book (SmythSmyth, Adam. "'Rend and teare in peeces': Textual Fragmentation in Seventeenth-Century England." <U>The Seventeenth Century</U> 19.1 (april 2004): 36-52. 37). Sometimes Gibson remixes the printed medium in simple ways, as when he appropriates a printed border to frame a verse (see SmythSmyth, Adam. "'Rend and teare in peeces': Textual Fragmentation in Seventeenth-Century England." <U>The Seventeenth Century</U> 19.1 (april 2004): 36-52. 42); other times, the remediated layers of copied verse, printed images and commentary radically reinterpret the source materials, as in (Smyth argues) the collage of texts related to Charles I's execution (see SmythSmyth, Adam. "'Rend and teare in peeces': Textual Fragmentation in Seventeenth-Century England." <U>The Seventeenth Century</U> 19.1 (april 2004): 36-52. 46). And, like Harsdörffer, Kuhlmann, and many other German figures during the

```
same period, Gibson shows a deep interest in anagrams, punctuating his cut-up
collages with wordplays and anagrammatic juxtapositions.</p>';
      break;
case '126':
      echo '<p>It is true what you&#39;re thinking &mdash; that cut-ups take
us rather far afield from <i>ars combinatoria</i> proper. Text generation
through permutation as represented in Harsd&ouml;rffer&#39;s Denckring is
systematic and computational; cutting and pasting, while combinatory in a
general sense, is more of an <i>ars compilationis</i>. In other words,
whereas the former selectively breeds one organism at a time from an existing
genome, the latter dissects organs from already-formed potentials, sewing
them back together to form something wholly new.</p><p>That is: <i>ars
combinatoria</i> is precise and conservative, operating within a closed
network, whereas <i>ars compilationis</i> is open, incessant and unwieldy, as
the scholar carves back the shoots and brambles that constantly threaten to
engulf her work.</p>';
      break;
case '127':
      echo '<p>Yet as writing practices, both <i>ars combinatoria</i> and
cut-ups operate through selection, through picking out discrete elements
&mdash; as in <i>legere</i>, to collect, gather, choose, pick out,
read.</p>';
      break;
case '128':
      echo '<p>Both <i>ars combinatoria</i> and what we might call <i>ars
compilationis</i> &mdash; perhaps better situated in the growing histories on
commonplacing, notetaking and marginalia (see, for example, <a
href="bibliography.html" target="_blank">Sherman<span>Sherman, William H.
<U>Used Books: Marking Readers in Renaissance England.</U> Philadelphia, PA:
University of Pennsylvania Press, 1008.</span></a>) &mdash; are literacies:
strategies for consuming, producing and using texts in a language-rich media
environment.</p>';
      break;
case '129':
      echo '<p>A growing literature is exploring how we read and write across
the dense network of twenty-first-century communications technologies (see <a
href="bibliography.html" target="_blank">Coiro et al.<span>Coiro, Julie and
Michele Knobel, Colin Lankshear, and Donald J. Leu, eds. <U>Handbook of
Research on New Literacies.</U> Lawrence Erlbaum Associates/Taylor & Francis
Group, 2008.</span></a>). Perhaps not surprisingly, our &quot;new&quot; media
literacies echo rather &quot;old&quot; strategies for manipulating
information. Just as Sir John Gibson cut and copied passages, notes and
references into notebook, we tumble and tweet the flood of net ephemera that
washes over our screens, cutting and pasting found materials into webforms
housed on servers we&#39;ll likely never see. In some ways, the structures we
use, whether a printed commonplace book or a WordPress account, constrain
how, when and where we compile our data; yet, just as Caramuel and
Harsd&ouml;rffer radically altered Ramist word tables into circular databases
to suit their programmatic epistemologies, we tweak Blogger&#39;s codes,
write opensource plug-ins and create our own tags, our own folksonomies.
</p>';
      break;
case '12911':
      echo '<p>Cutting, copying, pasting and combining are literacies:
strategies for consuming, producing, and using texts in a language-rich media
environment. It perhaps not surprising that we find deep similarities in the
literacies demanded of readers at the turn of the seventeenth century and
```

```
those at the turn of the twenty-first. Both eras were and are learning to
organize and understand vast amounts of information.</p>';
      break;
case '130':
      echo '<blockquote><p class="blockquote">All writing is in fact cut-ups.
A collage of words read heard overheard. What else? Use of scissors renders
the process explicit and subject to extension and variation. (<a
href="bibliography.html" target="_blank">Burroughs<span>Burroughs, William.
&quot;The Cut Up Method.&quot; <U>The Moderns: An Anthology of New Writing in
America.</U Ed. Leroi Jones. New York: Corinth Books, 1963. 345-8.</span></a>
347)</p></blockquote>';
      break;
case '131':
      echo '<blockquote><p class="blockquote">ALL WRITING IS IN FACT CUT-UPS
OF GAMES AND ECONOMIC BEHAVIOR OVERHEARD? WHAT ELSE? (<a
href="bibliography.html" target="_blank">Burroughs<span>Burroughs, William.
&quot;The Cut Up Method.&quot; <U>The Moderns: An Anthology of New Writing in
America.</U Ed. Leroi Jones. New York: Corinth Books, 1963. 345-8.</span></a>
347)</p></blockquote>';
      break;
case '13111':
      echo '<p>Perhaps the most well-known cut-up artist of the twentieth-
century is William S. Burroughs, who enacted Gysin&#39;s methodologies on his
own oeuvre. Slicing and folding his way through the Word Hoard, a trunk
stuffed with his typewritten work, Burroughs produced a set of three cut-up
novels: <i>The Soft Machine</i> (1961), <i>The Ticket That Exploded</i>
(1962) and <i>Nova Express</i> (1964).</p>';
      break;
case '13122':
      echo '<p>Through the physical manipulation of the page, Burroughs&#39;
cut-ups argue that &quot;the Word is literally a virus&quot; inhabiting its
&quot;human host,&quot; serving &quot;no internal function other than to
replicate itself&quot; (<a href="bibliography.html"
target="_blank">Burroughs<span>Burroughs, William S. <U>The Adding
Machine.</U> Arcade Publishing, 1993.</span></a> 48). This is, Burroughs
emphasizes, &quot;not an allegorical comparison&quot; (<a
href="bibliography.html" target="_blank">Burroughs<span>Burroughs, William S.
<U>The Adding Machine.</U> Arcade Publishing, 1993.</span></a> 59): language
physically inhabits the body, forcing us to experience the world through an
ongoing, neverending internal monologue (see <a href="bibliography.html"
target="_blank">Land<span>Land, Christopher. &quot;Apomorphine Silence:
Cutting-up Burroughs&#39; Theory of Language and Control.&quot;
<U>Ephemera</U> 5.3 (2005):450-470.</span></a> 453-456). </p>';
      break;
case '132':
      echo '<blockquote><p class="blockquote">Quite simply, Burroughs&#39;
cut-up project of the 1960s began as a way to systematize the drive to lose
the undesired past, to cut his way out of an old identity, if not out of
identity itself. ... The ambition to make a complete break, to cut off
history and dispossess the self, to kick the habit of what Walter Benjamin
once called &#39;that most terrible drug &mdash; ourselves &mdash; which we
take in solitude&#39;, this ambition demands discontinuity as a historical as
well as a formal principle. ... The disjunctive and metamorphic
juxtapositions that mark his cut-up texts of the 1960s undo the illusion of
fixed identity, and this sabotage coincided, fully and quite deliberately,
with the introduction of a radical discontinuity in the narrative of his own
biographical and literary history. (<a href="bibliography.html"
target="_blank">Harris<span>Harris, Oliver. <U>William Burroughs and the
```

```
Secret of Fascination.</u> Carbondale: Southern Illinois University Press,
2003.</span></a> 8-9)</p></blockquote>';
      break;
case '133':
      echo '<blockquote><p class="blockquote">Later Burroughs sought a self-
abolishing structure, and tried to defeat our codes of continuity, cultural
and temporal, by shuffling his prose into random order. &quot;Writers until
the cut-up method was made explicit,&quot; he says, &quot;had no way to
produce the accident of spontaneity.&quot; But it seems that in the logic of
the situation we shall find such accidents happy only when we see in them
some allusion, direct or ironical, to our inherited notions of linguistic and
narrative structure ... If Burroughs is a satirist, and he is, then that also
presupposes a past significantly altered. (<a href="bibliography.html"
target="_blank">Kermode<span>Kermode, Frank. <U>The Sense of an Ending:
Studies in the Theory of Fiction.</U> New York: Oxford University Press,
1967.</span></a> 117-8)</p></blockquote>';
      break;
case '134':
      echo '<p>The story has been written so many times; why struggle for new
words?</p><blockquote><p class="blockquote">At the surrealist rally in the
1920s, Tristan Tzara the man from nowhere proposed to create a poem on the
spot by pulling words out of a hat. A riot ensued wrecked the theatre.
Andr&eacute; Breton expelled Tristan Tzara from the movement and grounded the
cut up on the Freudian couch. (<a href="bibliography.html"
target="_blank">Burroughs<span>Burroughs, William. &quot;The Cut Up
Method.&quot; <U>The Moderns: An Anthology of New Writing in America.</U Ed.
Leroi Jones. New York: Corinth Books, 1963. 345-8.</span></a>
345)</p></blockquote>';
      break;
case '135':
      echo '<p>Here&#39;s how Gysin told the story in 1979:</p><blockquote><p
class="blockquote">I always refrained from ever even reminding Tzara of this
painful incidence in the Closerie des Lilas, where a traditional versifier
named Saint-Pol-Roux was being feted as the Prince of Poets. The loosely
amalgamated avant-garde of the time organized a commando night raid on the
celebrated cafe and was routed by the riot squad. Breton loudly proclaimed
that Tzara was a fink: he called the cops. When the lights went out, Breton
heard the voice of Tzara calling the commissariat on the telephone,
distinctly. The commandos chose to believe him. They went in together: they
came out apart. Dada was dead. (Gysin in <a href="bibliography.html"
target="_blank">Zurbrugg<span>Zurbrugg, Nicholas, ed. <U>Art, Performance,
Media: 31 Interviews.</U> Twin Cities: University of Minnesota Press,
2004.</span></a> 190)</p></blockquote>';
      break;
case '136':
      echo '<blockquote><p class="blockquote">Tristan Tzara and I used to
bump into one another sometimes in the late &#39;50s around about midnight,
for a standup steak and a beer at the circular zinc counter of the old Royal
Saint Germain, now transmogrified into the monstrous Le Drugstore, where no
poets meet who can help it. Every time we met, Tzara would whine, &quot;Would
you be kind enough to tell me just why your young friends insist on going
back over the ground we covered in 1920?&quot; What could I say, except,
&quot;Perhaps they feel you did not cover it thoroughly enough.&quot; (Gysin
in <a href="bibliography.html" target="_blank">Zurbrugg<span>Zurbrugg,
Nicholas, ed. <U>Art, Performance, Media: 31 Interviews.</U> Twin Cities:
University of Minnesota Press, 2004.</span></a> 190)</p></blockquote>';
      break;
case '137':
```

```
      echo '<p>There is yet another beginning to this
story.</p><blockquote><p class="blockquote">One day in late September 1959,
Brion Gysin was in his hotel room, mounting some drawings, slicing through
the boards with his Stanley knife and simultaneously slicing through the pile
of old New York Herald Tribunes he was using to protect his table. When he
finished, he noticed that where a strip of a page was cut away, the newsprint
on the next page lined up and could be read across, combining stories from
different pages, often with hilarious results. Some of the combinations
amused him so much that his neighbors in the hotel knocked on his door,
thinking that his laughter was a hysteria attack. (<a
href="bibliography.html" target="_blank">Miles<span>Miles, Barry. <U>William
Burroughs: el hombre invisible.</U> New York: Hyperion, 1993.</span></a>
111)</p></blockquote>';
      break;
case '138':
      echo '<p>In the early 1970s, Tom Stoppard writes this action into the
first scene of <i>Travesties</i>, a play starring James Joyce, Lenin, Henry
Carr and Tristan Tzara. Together, they sit in a library &quot;occupied with
books, papers, pencils . . .&quot;:</p><blockquote><p
class="blockquote">TZARA is writing as the play begins. On his table are a
hat and a large pair of scissors. TZARA finishes writing, then takes up the
scissors and cuts the paper, word by word, into his hat. When all the words
are in the hat he shakes the hat and empties it on the table. He rapidly
separates the bits of paper into random lines, turning a few over, etc., and
then reads the result in a loud voice. (<a href="bibliography.html"
target="_blank">Stoppard<span>Stoppard, Tom. <U>Travesties.</U> New York:
Grove Press, 1975.</span></a> 1-2)</p></blockquote>';
      break;
case '139':
      echo '<p>The hush of the library, with its categorized tomes tucked
carefully on their shelves, shatters with the sound of Tzara&#39;s voice,
chanting gibberish from a loose pile of paper scraps. Tzara&#39;s act of cut-
and-combine is more than a dissembling of the paper archive: it&#39;s a
recombination, compiled from material scraps like Golem, like the Tower of
Babel. Its very existence as such exposes the cut-up and combinatory nature
of all around it, including what is presumed, in the wagging finger of the
librarian, to be sacred.</p><p>No longer a natural or self-evident system,
the archive is exposed for the randomness of its rules and its arbitrary
organization. If the Biblical cut-ups of Jefferson or the Little Gidding
community gave birth to a new, more perfect text, Tzara&#39;s hat trick is a
form of linguistic death &mdash; an exquisite corpse.</p>';
      break;
case '140':
      echo '<blockquote><p class="blockquote">Like Dali&#39;s, Tzara&#39;s
hats are of felt, with their softly sloping crowns creased along their
summits to produce the parallel lips of the fashionable fedora. A labial
crease. A genital smile. (<a href="bibliography.html"
target="_blank">Krauss<span>Krauss, Rosalind. <U>The Optical Unconscious.</U>
Cambridge, MA: MIT Press, 1993.</span></a> 162)</p></blockquote><p>The women
of Little Gidding splice together texts like DNA in the womb; Tzara reaches
into a hat shaped like female genitals to extract lines of letters.</p><p>Can
we gender the act of recombination? Do women, so often deprived of the pen,
make their mark with the cut-ups of <i>ars combinatoria</i>?</p><p>And if so,
what does Tzara&#39;s distinctly male act &mdash; described as the moment
that gave birth to Dada &mdash; signal for a movement so distinctly
homosocial and masculine (see <a href="bibliography.html"
target="_blank">Hopkins<span>Hopkins, David. <U>Dada&#39;s Boys: Masculinity
```

```
after Duchamp.</U> New Haven, CT: Yale University Press,
2007.</span></a>)?</p>';
      break;
case '141':
      echo '<p>Like his friend Tzara before him, Brion Gysin &mdash; William
Burroughs&#39; colleague and, in Burroughs&#39; words, &quot;the only man
that I&#39;ve ever respected&quot; (Burroughs in <a href="bibliography.html"
target="_blank">Zurbrugg<span>Zurbrugg, Nicholas, ed. <U>Art, Performance,
Media: 31 Interviews.</U> Twin Cities: University of Minnesota Press,
2004.</span></a> 86) &mdash; experimented with cut-ups, slicing not only
pages but magnetic tape and film.</p><p>Many of his poems operate through
permutation, as in: &quot;Come to Free the Words,&quot; &quot;Too Free Come
the Words,&quot; &quot;The Words Come Too Free&quot; (quoted in <a
href="bibliography.html" target="_blank">Zurbrugg<span>Zurbrugg, Nicholas,
ed. <U>Art, Performance, Media: 31 Interviews.</U> Twin Cities: University of
Minnesota Press, 2004.</span></a> 191); others juxtapose sound, text and
image in a spontaneous, multimodal performance. He was, as he later described
it, &quot;attempting to show that these techniques could be decanted back and
forth through different media&quot; (Gysin in <a href="bibliography.html"
target="_blank">Zurbrugg<span>Zurbrugg, Nicholas, ed. <U>Art, Performance,
Media: 31 Interviews.</U> Twin Cities: University of Minnesota Press,
2004.</span></a> 192).</p>';
      break;
case '142':
      echo '<blockquote><p class="blockquote">There was a whole muggy area of
doubt as to what to call this monstrosity I helped bring about. Because of
the machines involved, the electronics, I would have called it &quot;machine
poetry,&quot; but everyone shied away from that. I felt good about creating
through the machines, and they did not. I wanted to make language work in a
new way, to surprise its secrets by using it as the material one passed
through the available electronics to amplify the voices of poetry. ... I
wanted to get as far away as possible from &quot;inspiration.&quot; I wanted
expiration instead, to breathe out rather than in. (Gysin quoted in <a
href="bibliography.html" target="_blank">Zurbrugg<span>Zurbrugg, Nicholas,
ed. <U>Art, Performance, Media: 31 Interviews.</U> Twin Cities: University of
Minnesota Press, 2004.</span></a> 194)</p></blockquote>';
      break;
case '143':
      echo '<p>In 1960, mathematician Ian Sommerville and Brion Gysin
collaborated on automating the process of permuting the phrase &quot;I AM
THAT I AM.&quot; Using a computer, Gysin and Sommerville programmed a simple
algorithm that would print out all 120 permutations of the phrase organized
into four text blocks (see <a href="bibliography.html"
target="_blank">Funkhouser<span>Funkhouser, Chris. <U>Prehistoric Digital
Poetry: An Archaeology of Forms, 1959-1995.</U> Tuscaloosa: University of
Alabama Press, 2007.</span></a> 39, <a href="bibliography.html"
target="_blank">Kuri<span>Kuri, Jos&eacute; F&eacute;rez, ed. <U>Brion Gysin:
Tuning in to the Multimedia Age.</U> London: Thames & Hudson,
2003.</span></a> 96-7).</p>';
      break;
case '144':
      echo '<p>Florian Cramer links computer programming and the cut-up
experiments of Gysin, Burroughs and Sommerville to mysticism and magic, all
of which rely on the formal execution of linguistic commands. &quot;Spoken by
the author on the tape recording,&quot; Cramer writes, Gysin&#39;s
permutations of &#39;IN THE BEGINNING WAS THE WORD&#39; &quot;were not solely
mathematical computations, but also incantations&quot; (<a
href="bibliography.html" target="_target">Cramer<span>Cramer, Florian.
```

```
<U>Words Made Flesh: Code, Culture, Imagination.</U> Piet Zwart Institute,
2005.</span></a> 17). Thus just as ritual provides a structure to order the
seemingly random chaos of the world, programming ritualizes language itself,
creating a controlled and yet infinitely expandable environment for
contemplating our experience as linguistically embodied selves.</p>';
      break;
case '146':
      echo '<p>Importantly and helpfully, Cramer&#39;s cultural history of
computing embeds material technologies in an immaterial context, teasing out
the subtle connections between software and the imagination. Yet the practice
of cut-ups and <i>ars combinatoria</i> is more than the manipulation of
symbols through matter; it is a complete materializing of language
itself.</p>';
      break;
case '147':
      echo '<p>By embodying itself as a running mental monologue, language
&mdash; particularly after the advent of written text &mdash; frames
existence as a linear narrative for Burroughs. Land
explains:</p><blockquote><p class="blockquote">Burroughs is suggesting that
the word-virus operates as an external &#39;other&#39; that colonizes the
body, forcing it to sub-vocalize and thereby reproducing itself. It is this
internal monologue, all but impossible to shut off and expressly non-human,
which produces an all-too-human sense of identity and self-continuity;
generating a linear, narrative time along which experience is distributed and
through which identity is assured. (<a href="bibliography.html"
target="_blank">Land<span>Land, Christopher. &quot;Apomorphine Silence:
Cutting-up Burroughs&#39; Theory of Language and Control.&quot;
<U>Ephemera</U> 5.3 (2005):450-470.</span></a> 455)</p></blockquote>';
      break;
case '14711':
      echo '<p>Which is to say: cut-ups cut-up control. Cut-ups slice through
the flat sheet of a linear reality to give the page new dimensions, new
depths. In doing so, cut-ups also free the poet &mdash; a necessarily
linguistic being always/already inhabited by the word virus &mdash; from the
parasite of narrative language, helping her to develop new linguistic sense
perceptions.</p>';
      break;
case '14722':
      echo '<p>Yet written language can never be wholly freed from its
material state. Ironically, just as Burroughs disembodies the word virus, it
attaches itself again to the page, clinging to the space where pools of ink
sink into the fibers of the paper. In short, cut-ups can only extract
obsessive bookishness from the human body by drawing attention to embodied
state of any book, any text, which only exists through the page
itself.</p><p>Thus being an active reader frees the writer to write; while
merely&quot;writing&quot; &mdash; that is, manipulating the text as inscribed
matter &mdash; allows the reader access to the text as a creation.</p>';
      break;
case '148':
      echo '<p>Thus just as Harsd&ouml;rffer mechanizes a materialist view of
language in his Denckring, so Burroughs&#39; theory of cut-ups treats
language as always/already embodied, refusing to lock it in an immaterial
mind.</p><p>From this angle, Burroughs&#39; linguistic theory &mdash; like
the history of <i>ars combinatoria</i> in general &mdash; might be read as a
reaction against contemporary movements to scientize the study of language.
As the cognitive revolution began locating linguistic meaning in syntactic
relationships and mental processes, cut-up artists refused to dematerialize
and disembody the word. For them, the relationship between human and text was
```

```
not that of process and product, but a palimpsest of media forms that
inscribed themselves onto and through the body.</p>';
      break;
case '14811':
      echo '<blockquote><p class="blockquote">We began to find out a whole
lot of things about the real nature of words and writing. What are words and
what are they doing? The cut-up method treats words as the painter treats his
paint, raw material with rules and reasons of its own . . . if you want to
challenge and change fate . . . cut up words, make them a new world. (Gysin,
quoted in <a href="bibliography.html" target="_blank">Miles<span>Miles,
Barry. <U>William Burroughs: el hombre invisible.</U> New York: Hyperion,
1993.</span></a> 130)</p></blockquote>';
      break;
case '14822':
      echo '<blockquote><p class="blockquote">My ambition was to destroy the
assumed natural links of language, that in the end are but expressions of
Power, the favourite weapon of control or even the essence of control.
(Gysin, quoted in <a href="bibliography.html" target="_blank">Kuri<span>Kuri,
Jos&eacute; F&eacute;rez, ed. <U>Brion Gysin: Tuning in to the Multimedia
Age.</U> London: Thames & Hudson, 2003.</span></a> 164)</p></blockquote>';
      break;
case '14833':
      echo '<blockquote><p class="blockquote">The permutated poems set the
words spinning of on their own; echoing out as the words of a potent phrase
are permutated into an expanding ripple of meanings which they did not seem
to be capable of when they were struck and then stuck into that phrase.
(Gysin in <a href="bibliography.html" target="_blank">Kuri<span>Kuri,
Jos&eacute; F&eacute;rez, ed. <U>Brion Gysin: Tuning in to the Multimedia
Age.</U> London: Thames & Hudson, 2003.</span></a> 154)</p></blockquote>';
      break;
case '149':
      echo '<p>Of course, it would be disingenuous not to point out
Burroughs&#39; later boredom with the whole project. As he said of cut-ups in
a 1983 interview with Nicholas Zurbrugg,</p><blockquote><p
class="blockquote">Well, I haven&#39;t done any of those in years, but they
were interesting. They explored the whole matter of synchronicity. You can
record in the street, and you can take something that you&#39;ve recorded and
play it back in the street, and you observe all sorts of synchronicities. It
just makes you aware of certain things going on all the time anyway.
(Burroughs in <a href="bibliography.html"
target="_blank">Zurbrugg<span>Zurbrugg, Nicholas, ed. <U>Art, Performance,
Media: 31 Interviews.</U> Twin Cities: University of Minnesota Press,
2004.</span></a> 70)</p></blockquote><p>And thus the cutting edge becomes
&quot;interesting&quot; &mdash; simply something to do.</p>';
      break;
case '150':
      echo '<p>As Funkhouser points out, Gysin and Sommerville&#39;s
experiment was not the first to use computers to automatically generate lines
of verse. In 1959, Theo Lutz used a Zuse Z22 computer to &quot;take over the
laborious production of stochastic texts,&quot; substituting the effort of,
for instance, throwing dice with a &quot;program-controlled data
processor&quot; to generate random numbers (<a href="bibliography.html"
target="_blank">Lutz<span>Lutz, Theo. &quot;Stochastische Texte.&quot;
<U>augenblick</u> 4 (1959): 3-9.</span></a> <a href="http://www.stuttgarter-
schule.de/lutz_schule_en.htm">online</a>). Here&#39;s how it
works:</p><blockquote><p class="blockquote">A new number is formed from an
initial number by an arithmetic operation, and from this number digits are
taken by intersection, which are then considered to be a random number. The
```

```
number generated by this operation is the initial number to determine the
next random number. By continuing this process, a sequence of numbers is
obtained. (<a href="bibliography.html" target="_blank">Lutz<span>Lutz, Theo.
&amp;quot;Stochastische Texte.&amp;quot; <U>augenblick</u> 4 (1959): 3-9.</span></a>
<a href="http://www.stuttgarter-
schule.de/lutz_schule_en.htm">online</a>)</p></blockquote>';
      break;
case '151':
      echo '<blockquote><p class="blockquote">The machine stores a certain
number of subjects, predicates, logical operators, logical constants and the
word &amp;quot;IST&amp;quot; (engl.: &amp;quot;IS&amp;quot;), coded as binary numbers. Using
the first random number the machine forms the address (i.e. the position
number in the store) of a subject by adding a constant which the machine now
has at its disposal. In the one-following memory cell, the program locates a
code number which it evaluates as gender of the subject in question, e.g. 0=
masculine, 1= feminine and 2 = neutral. The machine then determines a logical
operator using a new random number and coordinates this with the gender of
the subject, using the located code number. At this stage a print-out is done
for the first time e.g. the teleprinter prints:</p><p>NICHT JEDER BLICK<br
/>(engl.: NOT EVERY LOOK) (<a href="bibliography.html"
target="_blank">Lutz<span>Lutz, Theo. &amp;quot;Stochastische Texte.&amp;quot;
<U>augenblick</u> 4 (1959): 3-9.</span></a> <a href="http://www.stuttgarter-
schule.de/lutz_schule_en.htm">online</a>)</blockquote>';
      break;
case '152':
      echo '<p>To computationally compose a poem, Lutz took Franz Kafka&amp;#39;s
<i>The Castle</i> as his source text, cutting the text into sixteen titles
and subjects that were then stored in a database. Much like the roll of a
dice, the numbers randomly generated by the program called the titles and
subjects from the database, then organized them into a syntax according to
predetermined logical constraints. The result is, as Funkhouser describes it,
full of &amp;quot;discursive leaps and quirky, unusual semantic connections&amp;quot;
that force the reader to &amp;quot;connect and interpret abstractions in the poem
... and derive meaning from the verbal associations while reading the text in
and against its context&amp;quot; (<a href="bibliography.html"
target="_blank">Funkhouser<span>Funkhouser, Chris. <U>Prehistoric Digital
Poetry: An Archaeology of Forms, 1959-1995.</U> Tuscaloosa: University of
Alabama Press, 2007.</span></a> 37-8).</p>';
      break;
case '153':
      echo '<blockquote><p class="blockquote">Lutz&amp;#39;s selection of words,
combined with his programming method, enables a speculative, self-reflexive,
unconventional style of expression; the programming method consists of about
fifty commands and could theoretically generate more than four million
different sentences. (<a href="bibliography.html"
target="_blank">Funkhouser<span>Funkhouser, Chris. <U>Prehistoric Digital
Poetry: An Archaeology of Forms, 1959-1995.</U> Tuscaloosa: University of
Alabama Press, 2007.</span></a> 38)</p></blockquote>';
      break;
case '154':
      echo '<p>More recently, the cut-up method has been re-packaged as
Flarf, a form of poetry composed by (among other things) plucking snatches of
text from Google searches to combine into a poem.</p><p>Flarf is <a
href="http://www.brooklynrail.org/2009/02/books/flarf-from-glory-days-to-
glory-hole" target="_blank">energetic</a>. Flarf is <a
href="http://possumego.blogspot.com/2009/01/flarf-and-shit-and-fake-
shit.html" target="_blank">shit</a>. Flarf is <a
href="http://jacketmagazine.com/29/hoy-flarf.html" target="_blank">naive</a>.
```

```
Flarf is <a href="http://lime-tree.blogspot.com/2009/02/dale-smith-deals-
death-blow-to-flarf.html" target="_blank">dead</a>.</p><p>And perhaps the
only objectively true statement: Flarf is controversial.</p>';
      break;
case '155':
      echo '<p>In an April 2006 edition of Jacket magazine, Dan Hoy chides
Flarfists for not recognizing that &quot;there is always a power determining
orthodoxy in any &#39;random&#39; generator&quot; &mdash; particularly in a
corporate entity like Google. Hoy also accuses Flarf of ignoring its Dadaist
predecessors (<a href="bibliography.html" target="_blank">Hoy<span>Hoy, Dan.
&quot;The Virtual Dependency of the Post-Avant and the Problematics of Flarf:
What Happens when Poets Spend Too Much Time Fucking Around on the
Internet.&quot; <U>Jacket</U> 29 (April 2006): online.</span></a> <a
href="http://jacketmagazine.com/29/hoy-flarf.html"
target="_blank">online</a>).</p>';
      break;
case '156':
      echo '<p>I&#39;m not interested in debating the relative merits of
Flarf. For our purposes, it&#39;s enough to recognize how Flarf has
reinvigorated debate around the issue of randomness and chance in poetry
&mdash; and reintroduced the question of code from a new angle. Thus far,
most of the poems or volvelles or cut-ups I have discussed emerge directly
from some form of code practice &mdash; whether a procedure such as the
exquisite corpse, a paper mechanism or a text-generating Perl poem. Yet, as
Hoy argues, Google&#39;s obscure, closely-guarded search algorithms may
influence the underlying ideology of a poetic practice without an artist even
knowing how or why. What writer with a computer and access to the internet
hasn&#39;t used a search engine to skim across the surface of the web&#39;s
&quot;bottomless ocean &mdash; all potentiality&quot; (<b>Sokei-an</b>)?
Whose &quot;formless mind&quot; has been left untouched by the new tools that
surround us?</p>';
      break;
case '157':
      echo '<p>Just as Tzaras&#39;s hat trick forces Stoppard to rethink the
archive; just as Harsd&ouml;rffer&#39;s Denckring opens a new relationship
between poetry and the vernacular; just as Jefferson cuts up his Bible to
reconstitute religion, so Flarf invites to consider how algorithmic media
infiltrate the poetic mechanism. Maybe this is the ideological redemption
Flarf&#39;s critics keep hoping to find behind its patent absurdism; but most
likely not. Like other acts of <i>ars combinatoria</i>, Flarf&#39;s selective
cut-and-recombine methods tell us more about the information age in which we
live than it does about poetry. When overload impedes imagination, spin the
wheel, search the web.</p>';
      break;
case '158':
      echo '<p>In <a
href="http://pangrammaticon.blogspot.com/2006/02/against-generative-grammar-
of-flarf.html" target="_blank">a comment appended to Thomas
Basb&oslash;ll&#39;s response</a>, Anne Boyer writes that &quot;Hoy misses
the mark &mdash; the progenitor of Flarf isn&#39;t Cage, or Queneau, but
Baroness von Else Freytag Loringhoven.&quot; Although written out of most
histories of modernism, Freytag-Loringhoven was a well-known female avant-
garde poet and subject of a recent feminist re-reading of Dada by Amelia
Jones.</p><p>If Dada began with Tzara pulling cut-up words from a yonic hat,
what does it mean that the most hated cut-up poetry movement of the twenty-
first century identifies with the Baroness, rather than Cage or
Queneau?</p>';
      break;
```

```
case '159':
      echo '<blockquote><p class="blockquote">To use and/or promote a search
engine without question is an implicit acceptance of it as an arbiter of
relevance. This is like compiling a list of what&#39;s going on in the world
from only the U.S. network TV news shows, without acknowledging the biases
inherent in their selection processes. Google is not the zeitgeist, nor is it
an indifferent and all-inclusive database of it. Google, as a generator, is a
corporate algorithm that ranks webpage relevancy (from a limited cross-
section &mdash; i.e. what it bothers to index &mdash; of an already limited
segment of the zeitgeist &mdash; i.e. Internet users and web content) based
on its own idiosyncratic definition of &#39;relevancy&#39; and, in tandem
with corporate web designers, manipulation of the results. (<a
href="bibliography.html" target="_blank">Hoy<span>Hoy, Dan. &quot;The Virtual
Dependency of the Post-Avant and the Problematics of Flarf: What Happens when
Poets Spend Too Much Time Fucking Around on the Internet.&quot; <U>Jacket</U>
29 (April 2006): online.</span></a> <a
href="http://jacketmagazine.com/29/hoy-flarf.html"
target="_blank">online</a>)</p></blockquote>';
      break;
case '161':
      echo '<blockquote><p class="blockquote">In the beginning was the word.
Everything seems to be wrong with what was produced from those beginnings, so
let&#39;s rub out the word and start afresh ... if the whole thing began with
the word, well then, if we don&#39;t like what was produced, and we
don&#39;t, let&#39;s get right to the root of the matter and radically alter
it. (Gysin, quoted in <a href="bibliography.html"
target="_blank">Kuri<span>Kuri, Jos&eacute; F&eacute;rez, ed. <U>Brion Gysin:
Tuning in to the Multimedia Age.</U> London: Thames & Hudson,
2003.</span></a> 130)</p>';
      break;
case '162':
      echo '<p><b>To rub out</b>: a verb phrase meaning both to erase,
eradicate, remove the word. By permuting the phrase to the point of absurdity
in his programmed permutation poem &#39;RUB OUT THE WORD&#39;, Gysin extracts
the (as Burroughs puts it) &quot;word virus&quot; from the human, rendering
meaning as nonsense, and written language as no more than set of arbitrary
symbols.</p><p>Yet rub out also carries a sense of creation, construction
&mdash; as in to &quot;rub the text out of a slab of marble,&quot; removing
the negative space around letters to bring a text to the surface. Thus to rub
out the word is both to raze it and to raise it, wiping the surface clean
even as one constructs a new reality.</p>';
      break;
case '163':
      echo '<p>Thus Gysin permutes The Word to break outside an all-
encompassing monotheistic system &mdash; indeed, to break outside of Meaning
itself &mdash; but ends up reinstating another form of mystical creation:
producing the infinite from the finite, or (to return to Leibniz&#39;s binary
system) 1 from 0.</p><p>In this way, like Llull&#39;s machines of conversion,
the material manipulation of symbols spins, and spins, and spins until
language takes flight, no longer anchored to any totalizing linguistic system
but becomes pure sound &mdash; a pure, pre-Babel utterance. Writing is not a
process of making meaning, then, but of destroying meaning; and reading is
precisely what we typically conceive of as writing &mdash; an attempt to find
sense in the chards.</p>';
      break;
case '164':
      echo '<p>Perhaps mystic Quirinus Kuhlmann and Brion Gysin share more
than either might have imagined.</p>';
```

```
      break;
 case '170':
      echo '<p>As Lutz&#39;s work shows, even the earliest use of electronic
digital computers treated writing as, to borrow a term from Kenneth
Goldsmith, a fundamentally &quot;uncreative&quot; act of cutting up and
recombining existing texts.. Indeed, the very act of programming &mdash; of
building a model that (at least in much digital poetry) calls from and
manipulates data stored in a database &mdash; engages writing as a form of
algorithmic generation (see <a href="bibliography.html"
target="_blank">Kirschenbaum<span>Kirschenbaum, Matthew. &quot;Hello
Worlds.&quot; <U>The Chronicle Review.</u> January 23, 2009:
online.</span></a>). Likewise, from their earliest conception digital poems
questioned the traditional role of the reader. As Charles O. Hartman writes,
as the reader becomes more important, poets begin to &quot;think of moving
him or her away from that end-of-the-road box in the diagrams and back into
the process somewhere,&quot; thereby making &quot;the reader&#39;s
constructive role in the poem more conscious&quot; (<a
href="bibliography.html" target="_blank">Hartman<span>Hartman, Charles O.
<U>Virtual Muse: Experiments in Computer Poetry</U>. Hanover, N.H.: Weleyan
University Press, 1996.</span></a> 105).</p>';
      break;
 case '171':
      echo '<blockquote><p class="blockquote">The phenomenology of reading
that emerges from them shifts away from the sense of reading and writing as
the only two directions of a highway into and out of the soul, and adds a
very tangible third person plural. Constraints of keyboard, mouse, and screen
are experienced as resistances from a machine intelligence governed by a
social network whose horizon is out of sight, and these resistances are felt
as both gateways and interfaces between body and machine. Text no longer
projects as a bounded material object; it is more like a landscape of which
only part is visible due to the limits of the gaze. Text unrolls down the
screen in a manner not unlike the road or track unrolling before the gaze in
a computer simulation of driving, and that is based on the framing of
landscape by a windshield, so we might be tempted to wonder if the experience
of reading has more affinity with driving than before. (<a
href="bibliography.html" target="_blank">Middleton<span>Middleton, Peter.
<U>Distant Reading: Performance, Readership, and Consumption in Contemporary
Poetry.</u> Tuscaloosa: University of Alabama Press, 2005.</span></a> 145-
6)</p></blockquote>';
      break;
 case '172':
      echo '<p>Funkhouser has identified a number of other early experiments
in digital poetry: Auto-Beatnik, a computer programmed to spit out beatnik-y
verse (1962); Jackson Mac Low&#39;s programmed film reader which randomly
permuted messages (1969); Angel Carmona&#39;s &quot;V2 Poems&quot; and Alan
Sondheim&#39;s TI59 calculator poems (both 1970s). As should be evident, many
of the early works were generative and combinatory. Thus, just as Lull&#39;s
arguments exist in the nexus between his combinatory Figures and their
constants, the poetry in these experiments lies in the interaction between a
designed program and a word list that, together, produce what appears to be a
traditional poem.</p>';
      break;
 case '173':
      echo '<p>Here, it&#39;s worth pausing over one oft-cited early computer
program: TRAVESTY, a text generator written in Pascal and developed in 1984
by literary critic Hugh Kenner and computer scientist Joseph O&#39;Rourke.
Designed as an algorithm for analyzing, manipulating, and (re)producing text,
TRAVESTY takes a source text, divides it into character strings (or letters)
```

```
of length <i>n</i>, then produces a statistical frequency table for any given
set of <i>n</i> characters. This table is  used to produce another text with
the same statistical properties. Or, to use a noncomputational metaphor,
TRAVESTY eats, digests and regurgitates any source text provided by its
user.</p>';
      break;
 case '174':
      echo '<p>When <i>n</i> = 1, the output text is almost unreadable, each
individual letter having been scrambled. However, when <i>n</i> = 9, the
longest string possible in the program, the output text appears almost
entirely the same as its input. Here&#39;s an example from Hartman before and
after an <i>n</i> = 9 travesty:</p><blockquote><p class="blockquote">Dead
flies cause the ointment of the apothecary to send forth a stinking savor: so
doth a little folly him that is in reputation for wisdom and honour.<br/>Dead
flies cause the ointment of the rule: folly is set in great dignity, and the
end of his mout is foolishness: and the end of his talk is mischievous
madness. (<a href="bibliography.html" target="_blank">Hartman<span>Hartman,
Charles O. <U>Virtual Muse: Experiments in Computer Poetry</U>. Hanover,
N.H.: Weleyan University Press, 1996.</span></a> 55)</p></blockquote>';
      break;
 case '175':
      echo '<blockquote><p class="blockquote">Travesty [an early text-
generating computer program] offers ... the wickedness of exploding revered
literary scripture into babble. We can reduce Dr. Johnson to inarticulate
imbecility, make Shakespeare talk very thickly through his hat, or exhibit
Francis Bacon laying waste his own edifice of logic amid the pratfalls of
<i>n</i> = 9. Yet the other side of the coin is a kind of awe. Here is
language creating itself out of nothing, out of mere statistical noise. (<a
href="bibliography.html" target="_blank">Hartman<span>Hartman, Charles O.
<U>Virtual Muse: Experiments in Computer Poetry</U>. Hanover, N.H.: Weleyan
University Press, 1996.</span></a> 57)</p></blockquote>';
      break;
 case '176':
      echo '<p>By conceptualizing writing as a process of analysis and
manipulation, TRAVESTY reduces meaning to a user-determined statistic, the
numerical primitive long-sought by Leibniz. At the same time, the
program&#39;s output continually slides along the rim between sense and
nonsense, between language-as-meaning and language-as-data; in this way, much
like the cut-ups of Tristan Tzara, TRAVESTY forces the always already
literate reader to question the text&#39;s &quot;authenticity.&quot; Thus
TRAVESTY represents a return to an unfulfilled promise of the seventeenth-
century programmatic epistemology even as it participates in the postmodern
politics of twentieth-century procedural art, finding wholeness from
plurality while exploding any sense of the &quot;natural&quot; in
&quot;natural language.&quot;</p>';
      break;
 case '177':
      echo '<p>At about <i>n</i> = 3, the early text-generating computer
program TRAVESTY produces phrases like: &quot;the thy hedge, afte se the
whatter&quot; &mdash; in other words, nonsense, but, as Hartman puts it,
&quot;clearly <i>English</i> nonsense&quot; (<a href="bibliography.html"
target="_blank">Hartman<span>Hartman, Charles O. <U>Virtual Muse: Experiments
in Computer Poetry</U>. Hanover, N.H.: Weleyan University Press,
1996.</span></a> 56). According any standard lexicon (including the <i>Oxford
English Dictionary</i>), &#39;whatter&#39; is not a word; yet it follows
standard English rules for word formation &mdash; the &#39;h&#39; follows the
&#39;w&#39;, &#39;er&#39; is a common ending &mdash; and so seems like it
<i>could</i> be.</p><p>Thus, much like Leibniz&#39;s alphabet or
```

```
Harsd&ouml;rffer&#39;s Denckring, the TRAVESTY program treats written
language as a set of rule-driven primitives and, in doing so, transforms a
closed lexicon of isolated words into an open, infinitely-regenerative
algorithm of linguistic possibilities.</p>';
      break;
 case '178':
      echo '<p>Hartman subsequently used TRAVESTY to compose poems. He began
by writing several of his own verses, then feeding them to the program using
8 different <i>n</i> values (all but <i>n</i> = 1). These eight outputs he
organized into a loose structure, then manipulated according to his own
intuitive sense of the most &quot;superior&quot; output (see <a
href="bibliography.html" target="_blank">Hartman<span>Hartman, Charles O.
<U>Virtual Muse: Experiments in Computer Poetry</U>. Hanover, N.H.: Weleyan
University Press, 1996.</span></a> 62-4). The result is his &quot;Monologues
of Soul and Body.&quot;</p>';
      break;
 case '179':
      echo '<blockquote><p class="blockquote">Thinking up ways to &#39;do
computer poetry&#39; makes us look at language and poetry from an unfamiliar
angle. This seems appropriate if one of poetry&#39;s functions is to make us
aware, with a fresh intensity, of our relation to the language that
constitutes so much of human life &mdash; or if you like, of how language
constitutes so much of our relation to the world. How do words mean when we
put them into new contexts? Under what conditions does the meaning web tear
apart? What meanings can words make (or can we make of them) when we disturb
their normal relation to each other? These are questions that any poet can be
seen as asking; and from this point of view, someone experimenting with
computer poetry is continuing an age-old project. (<a
href="bibliography.html" target="_blank">Hartman<span>Hartman, Charles O.
<U>Virtual Muse: Experiments in Computer Poetry</U>. Hanover, N.H.: Weleyan
University Press, 1996.</span></a> 104)</p></blockquote>';
      break;
 case '180':
      echo '<p>As procedural works, early digital poems are often
historicized &mdash; if they are historicized at all &mdash; against the
modernist movements of the twentieth century, particularly dada, Fluxus, the
Oulipo and cut-up methods. Thus for Glazier, the &quot;post-typographic and
nonlinear disunion&quot; of digital poetry emerges from &quot;Futurism,
Gertrude Stein, and Dada,&quot; among others (<a href="bibliography.html"
target="_blank">Glazier<span>Glazier, Loss Peque      &ntilde;o. <U>Digital
Poetics: The Making of E-Poetries.</U> Tuscaloosa, AL: University of Alabama
Press, 2002.</span></a> 35); for McGann, Mallarm&eacute;&#39;s <i>Un coup de
d&eacute;s</i> comes to &quot;forecas[t] key terms by which we now
characterize digital media&quot; (<a href="bibliography.html"
target="_blank">McGann<span>McGann, Jerome J. <U>Radiant Textuality:
Literature After the World Wide Web.</U> New York: Palgrave, 2001.</span></a>
210); and electronic literatures are theorized against their &quot;print
precursors&quot; (<a href="bibliography.html"
target="_blank">Sloane<span>Sloane, Sarah. <U>Digital Fictions: Storytelling
in a Material World.</U> New York: Greenwood Publishing Group,
2000.</span></a> 44).</p>';
      break;
 case '181':
      echo '<p>Yet  modeling digital poetry on modernist movements presents
an interesting dilemma &mdash; what Sandy Baldwin calls the &quot;paradox of
innovation.&quot; For under this historical regime, &quot;the &#39;new&#39;-
ness of literary innovation occurs against the background of a tradition that
novelty ends up reinforcing&quot; (<a href="bibliography.html"
```

```
target="_blank">Baldwin<span>Baldwin, Sandy. &quot;A Poem is a Machine to
Think With: Digital Poetry and the Paradox of Innovation.&quot; Review of
Loss Peque&ntilde;o Glazier, <U>Digital Poetics: The Making of E-
Poetries.</U> <U>Postmodern Culture</U> 13:2 (January 2003).
Online.</span></a> <a href="http://www.iath.virginia.edu/pmc/text-
only/issue.103/13.2baldwin.txt" target="_blank">online</a>). In other words,
situating digital poetry within an always already avant-garde politics
immediately renders it stale, uninteresting &mdash; merely an electronic
iteration of procedural poems already produced on paper. In this model, the
only thing &quot;new&quot; about digital poetry is the technology
itself.</p>';
      break;
 case '182':
      echo '<p>On the flip side, digital poetry is sometimes imagined as a
radical rupture from past practices precisely <i>because</i> of its
technological newness. Thus Alan Sondheim writes that &quot;for thousands of
years, writers have, again in general, taking their tools &mdash; taken
writing itself &mdash; for granted. Even Sterne and Carroll work within
traditional means. The computer and the Internet, however, have opened up a
whole (and indefinable) world of possibilities&quot; (<a
href="bibliography.html" target="_blank">Sondheim<span>Sondheim, Alan.
&quot;Introduction: Codework.&quot; <U>ABR</U> 22.6 (September/October
2001).</span></a> 1).</p>';
      break;
 case '183':
      echo '<p>Yet by focusing on technological rather than aesthetic
&quot;newness,&quot; Sondheim goes on to describes these &quot;vast uncharted
domains ... of new and future literatures&quot; (<a href="bibliography.html"
target="_blank">Sondheim<span>Sondheim, Alan. &quot;Introduction:
Codework.&quot; <U>ABR</U> 22.6 (September/October 2001).</span></a> 2)
&mdash; what he calls <i>codework</i> &mdash; in terms that could easily
apply to Harsd&ouml;rffer&#39;s Denckring or Caramuel&#39;s Maria Stella .
Just as the paradox of innovation destroys the avant-garde, so does an
obsession with newness reduce all literature to an outcome of its
technological functionalities &mdash; in many cases, functionalities
ironically found across multiple platforms.</p>';
      break;
 case '184':
      echo '<p>Importantly, my critique is not of these histories themselves,
but of their deployment. Certainly dada deeply influenced many digital
artists and programmars; likewise it would be facile to posit too deep of a
connection between, say, Caramuel&#39;s Maria Stella and the TRAVESTY
program. Rather, presenting digital poetry as radically new or even
procedurally modernist overemphasizes media technologies at the expense of
what Erkki Huhtamo, following Ernst Robert Curtius, calls <i>media topoi</i>,
or &quot;cyclically recurring elements and motives underlying and guiding the
development of media culture&quot; and, in particular, literacy (<a
href="bibliography.html" target="_blank">Huhtamo<span>Huhtamo, Erkki.
&quot;From kaleidoscomaniac to cybernerd: Notes toward an archaeology of the
media.&quot; <U>Electronic Culture: Technology and Visual Representation.</U>
Ed. by Timothy Druckrey. London: Aperture Foundation, 1996.</span></a>
301).</p><p>What if, instead of linking &quot;digital poems&quot; to
modernist &quot;procedural poems,&quot; we broadened our investigation to
combinatory reading and writing practices across multiple material and
aesthetic platforms?</p>';
      break;
 case '185':
```

```
      echo '<p>To understand Harsd&ouml;rffer&#39;s Denckring in relation to
an electronically-mediated work, we must take a radically anti-evolutionary
approach to poetry. Literatures do not grow and then die; nor do they compete
among each other in a race for the best environmental fit. The process of
historicizing a set of practices demands that we explore their underlying
epistemologies, both poetic and philosophical, as they have been embedded in
very different media machines. </p>';
      break;
 case '186':
      echo '<p>In short, this machine acts as what Hayles calls a
&quot;technotext&quot;: it constantly &quot;interrogates the inscription
technology that produces it&quot; and, in doing so, &quot;mobilizes reflexive
loops between its imaginative world and the material apparatus embodying that
creation as a physical presence&quot; (Hayles 25). Hayles distinguishes her
term from Espen Aarseth&#39;s widely-used &quot;cybertext&quot; &mdash;
defined by Aarseth as a &quot;machine for the production of variety of
expression&quot; (Aarseth 3) &mdash; by noting that it overemphasizes user
interaction, &quot;connot[ing] a functional and semiotic approach that
emphasizes a computational perspective&quot; (Hayles 29). By contrast,
technotexts engages its own materiality in a variety of ways, both reader-
and author-centered.</p>';
      break;
case '190':
      echo '<blockquote><p class="blockquote">A mechanism is faulty not for
being too artificial to account for living matter, but for not being
mechanical enough, for not being adquately machined. Our mechanisms are in
fact organized into parts that are not in themselves machines, while the
organism is infinitely machined, a machine whose every part or piece is a
machine, but only &quot;transformed by different folds that it
receives.&quot; (<a href="bibliography.html"
target="_blank">Deleuze<span>Deleuze, Gilles. <U>The Fold: Leibniz and the
Baroque.</u> Tran. by Tom Conley. Minneapolis: University of Minnesota Press,
1993.</span></a> 8)</p></blockquote>';
      break;
 case '200':
      echo '<p>As the final battles of what would become known as the Thirty
Years War were being waged in central Europe, Gottfried Wilhelm Leibniz was
born in Leipzig, a Saxony trading town in the northeast corner of the Holy
Roman Empire. His father Friedrich, a professor of moral philosophy, died
when Leibniz was only six, leaving a large library of books. At the urging of
a neighbor, Leibniz&#39;s mother opened her husband&#39;s library to the
curious young boy, who spent his formative years wandering through its
diverse collections (see <a href="bibliography.html"
target="_blank">Antognazza<span>Antognazza, Maria Rose. <U>Leibniz: An
Intellectual Biography.</U>New York, NY: Cambridge University Press,
2009.</span></a> 33-7).</p>';
      break;
case '201':
      echo '<p>In his father&#39;s library, young Leibniz encountered German
baroque poetry, moral philosophy, legal books, and perhaps even a &quot;mass
of unbound material deriving from the father of Friedrich Leubnitz&#39;s
second wife, the bookseller and publisher, Bartholom&auml;us Voigt&quot; (<a
href="bibliography.html" target="_blank">Antognazza<span>Antognazza, Maria
Rose. <U>Leibniz: An Intellectual Biography.</U>New York, NY: Cambridge
University Press, 2009.</span></a> 34). Thus even as his formal education
drilled him the strict German semi-Ramist tradition of logic, his informal
wandering introduced him to an eclectic mix of religious and aesthetic belief
systems, later influencing his notion of a <i>mathesis universalis</i>
```

```
&mdash; a method of generating, with mathematical certainty, the answer to
any philosophical question.</p>';
      break;
case '202':
      echo '<p>One book in particular had a deep influence on young Leibniz:
Daniel Schwenter and G. P. Harsd&ouml;rffer&#39;s <i>Delici&aelig; Physico-
Mathematiac&aelig;</i> (1636-1653), an imaginative collection of mathematical
and poetic games and inventions, including the first known design for a
fountain pen as well as a text-generating volvelle, the Denckring (<a
href="bibliography.html" target="_blank">Antognazza<span>Antognazza, Maria
Rose. <U>Leibniz: An Intellectual Biography.</U>New York, NY: Cambridge
University Press, 2009.</span></a> 62-3, <a href="bibliography.html"
target="_blank">Reed<span>Reed, Eugene E. &quot;Leibniz, Wieland, and the
Combinatory Principle.&quot; <U>The Modern Language Review</U> 56.4 (1961):
529-37.</span></a> 529).</p><p>In fact, Leibniz&#39;s &quot;alphabet of human
thoughts&quot; contains traces of Stammw&ouml;rter theory, while his
<i>mathesis universalis</i> &mdash; an algorithm for permuting his alphabet
into universal knowledge &mdash; works much like the Denckring. Both operate
through a computational imaginary that condenses meaning into discrete
variables before exploding it exponentially, mechanically.</p>';
      break;
case '203':
      echo '<p>In 1666, at the age of twenty, Leibniz defended his
<i>Dissertatio de Arte Combinatoria</i>. Although now difficult to find and
little studied &mdash; in a typical dismissal, Reed states it &quot;contained
nothing worthy of the Leibnizian genius&quot; (<a href="bibliography.html"
target="_blank">Reed<span>Reed, Eugene E. &quot;Leibniz, Wieland, and the
Combinatory Principle.&quot; <U>The Modern Language Review</U> 56.4 (1961):
529-37.</span></a> 529) &mdash; the work was published  immediately and
became surprisingly successful for a student dissertation (<a
href="bibliography.html" target="_blank">Antognazza<span>Antognazza, Maria
Rose. <U>Leibniz: An Intellectual Biography.</U>New York, NY: Cambridge
University Press, 2009.</span></a> 62).</p><p>In it, Leibniz explores Ramon
Llull&#39;s <i>ars magna</i>, or his combinatorial art &mdash; a subject
tackled by many thinkers of the period, including Johann Heinrich Alsted and
Athanasius Kircher. Like Llull himself, Leibniz explores the art of
combination not simply as mechanical means for permuting discrete elements,
but as system for logical discovery.</p>';
      break;
case '204':
      echo '<blockquote><p class="blockquote">It was fresh from the reading
of Harsd&ouml;rffer&#39;s work that Leibniz proceeded to demonstrate the
enormous potential of combination inherent in but a single poetic line -- an
application simple and obvious to the product of an age seeking new
combinations within the framework of the classical tradition. (<a
href="bibliography.html" target="_blank">Reed<span>Reed, Eugene E.
&quot;Leibniz, Wieland, and the Combinatory Principle.&quot; <U>The Modern
Language Review</U> 56.4 (1961): 529-37.</span></a> 529)</p></blockquote>';
      break;
case '205':
      echo '<p>For young Leibniz, knowledge emerges from an &quot;alphabet of
human thoughts&quot; &mdash; that is, a set of primitives that constitute the
most basic units of philosophical concepts. Combining these primitives
through a logically-valid algorithm will, by necessity, produce true
arguments, thus contributing to a universal science.</p><p>As Leibniz later
argued, &quot;a kind of alphabet of human thoughts can be worked out,&quot;
and &quot;everything can be discovered and judged by a comparison of the
letters of this alphabet and an analysis of the words made from them&quot;
```

```
(<a href="bibliography.html" target="_blank">Leibniz<span>Leibniz, G. W.
&quot;On the General Characteristic.&quot; In <U>Philosophical Papers and
Letters, 2nd ed.</U> Trans. by Leroy E. Loemker. Boston: D. Holland
Publishing Company, 1956.</span></a> 222).</p>';
      break;
case '206':
      echo '<blockquote><p class="blockquote">Leibniz&#39;s real proposal is
not for a device that relies on success in working out the correspondences of
this language, but for one that brings its syntax into effect prior even to
the discovery of its semantics. Even in the absence of a ready-made lexicon
of primitive concepts, the first stunning effect of Leibniz&#39;s clarity-
machine would be to give an a priori proof for its own possibility. (<a
href="bibliography.html" target="_blank">Selcer<span>Selcer, Daniel.
&quot;The Uninterrupted Ocean: Leibniz and the Encyclopedic
Imagination.&quot; <U>Representations</U> 98.1 (2007): 25-50.</span></a>
26)</p></blockquote>';
      break;
case '207':
      echo '<p>Like the baroque theory of Stammw&ouml;rter &mdash; still
popular when Leibniz was writing his Dissertatio &mdash; Leibniz&#39;s
alphabet first reduces what is known before producing what is true, setting
up a closed system that a priori proves itself (see <a
href="bibliography.html" target="_blank">Selcer<span>Selcer, Daniel.
&quot;The Uninterrupted Ocean: Leibniz and the Encyclopedic
Imagination.&quot; <U>Representations</U> 98.1 (2007): 25-50.</span></a> 26).
Thus just as any word or line of poetry produced by Harsd&ouml;rffer&#39;s
Denckring always/already mirrors the <i>lingua adamica</i>, the originary and
perfect vernacular, so too does any knowledge produced through Leibniz&#39;s
alphabet of thoughts automatically mirror some universal truth. Both poetry
and philosophy become not product but process &mdash; a system, mechanized in
a set of nesting paper wheels, whose very existence instantiates its
combinatory possibilities.</p>';
      break;
case '208':
      echo '<p>It is useful to frame Leibniz&#39;s project not only through
his baroque contemporaries and forebears, but also through contrasting
philosophical movements across Europe.</p><p>A year after Leibniz published
his <i>Dissertatio</i>, the Englishman John Wilkins published his famous
<i>Essay Towards a Real Character and Philosophical Language</i>, which lays
out his plan for a philosophical language, immutable in both orthography and
pronunciation. Wilkins&#39; language (like that of Dalgarno and other
seventeenth-century thinkers) relies on a Ramist taxonomy that hierarchizes
the world into a system of branching nodes that move from the general to the
specific, accreting characteristics through distinctions. Thus a
&quot;Tiger&quot; is a viviparous, cloven-footed, rapacious cat-kind beast
with shorter legs but a larger body than other cats, characters by its spots
(<a href="bibliography.html" target="_blank">Wilkins<span>Wilkins, John.
<U>An Essay towards a Real Character and a Philosophical Language.</U>
London: 1668.</span></a> 159).</p>';
      break;
case '209':
      echo '<p>By contrast, Leibniz&#39;s alphabet of human thoughts is
radically reductive. Rather than pursue the static, cumulative logic of
tables and taxonomies, Leibniz advocates stripping down philosophy to its
most basic concepts, then devising a system for combining these primitives to
&quot;discover&quot; (i.e., generate) all truth. In this way, the philosopher
becomes an engineer, designing machines, and philosophy itself becomes the
exponential power of <i>ars combinatoria</i>.</p>';
```

```
      break;
case '210':
      echo '<blockquote><p class="blockquote">This encyclopedia was to be
both a comprehensive and generative text, its categorical structure not
merely recording scientific, intellectual, historical and literary
achievements but functioning as a philosophical machine for the production
and organization of knowledge. (<a href="bibliography.html"
target="_blank">Selcer<span>Selcer, Daniel. &quot;The Uninterrupted Ocean:
Leibniz and the Encyclopedic Imagination.&quot; <U>Representations</U> 98.1
(2007): 25-50.</span></a> 29)</p></blockquote>';
      break;
case '211':
      echo '<p>In short, whereas the Ramist tables and taxonomies of a late
seventeenth-century thinker like John Wilkins enclose the world within a
(somewhat arbitrary) structure, Leibniz&#39;s program writes a structure into
the world. The former orders knowledge; the latter generates it.</p><p>The
former reflects reality; the latter <i>is</i> reality.</p>';
      break;
case '212':
      echo '<blockquote><p class="blockquote">By the time he had completed
his formal education, Leibniz&#39;s vision had been born. He had decided what
to do with his life. Born into a fragmented world lacerated by religious,
political, and intellectual crises, &quot;Wilhelmus Pacidius&quot;, alia
Gottfried Wilhelm Leibniz, was going to put the pieces together to achieve a
universal synthesis for the glory of God and the happiness of mankind. This
synthesis would be designed to restore unity in multiplicity, unveiling the
universal harmony which, despite apparently unbridgeable divisions, governed
reality at both the metaphysical and the epistemological level[.] (<a
href="bibliography.html" target="_blank">Antognazza<span>Antognazza, Maria
Rose. <U>Leibniz: An Intellectual Biography.</U>New York, NY: Cambridge
University Press, 2009.</span></a> 66-7)</p></blockquote>';
      break;
case '213':
      echo '<p>This &quot;alphabet of human thoughts&quot; remained an
undercurrent in Leibniz&#39;s philosophy throughout his life, manifesting
itself in a number of different plans: his dream for a networked encyclopedia
in which, through linking, every entry was a microcosm of the human macrocosm
(see <a href="bibliography.html" target="_blank">Selcer<span>Selcer, Daniel.
&quot;The Uninterrupted Ocean: Leibniz and the Encyclopedic
Imagination.&quot; <U>Representations</U> 98.1 (2007): 25-50.</span></a> 29);
his lingua characteristica, or notation system for concepts &quot;whose signs
or characters serve the same purpose that arithmetical signs serve for
numbers&quot; (<a href="bibliography.html"
target="_blank">Leibniz<span>Leibniz, G. W. &quot;On the General
Characteristic.&quot; In <U>Philosophical Papers and Letters, 2nd ed.</U>
Trans. by Leroy E. Loemker. Boston: D. Holland Publishing Company,
1956.</span></a> 222); even his notion of &quot;monads&quot; &mdash;
discrete, irreducible primitives that nonetheless reflect the infinity of the
spiritual cosmos. More specifically, Leibniz develops his
&quot;alphabet&quot; through an account of a <i>mathesis universalis</i>, a
universal system for storing and generating knowledge.</p>';
      break;
case '214':
      echo '<blockquote><p class="blockquote">Leibniz, who had subjected
(and, if necessary, corrected) all mathematical signs to Gutenberg&#39;s
place-value logic, recognized in Zero the nothingness preceding creation, and
in One, divine creation itself. No wonder, then, that his binary system was
said to have been able to describe the whole of being. (<a
```

```
href="bibliography.html" target="_blank">Kittler<span>Kittler, Friedrich A.
&quot;The Perspective of Print.&quot; <U>Configurations</U> 10.1 (Winter
2002): 37-50.</span></a> 48)</p></blockquote>';
      break;
case '215':
      echo '<blockquote><p class="blockquote">[T]here are two crucial notions
that determine Leibniz&#39;s account of the <i>mathesis universalis</i>. The
first is the idea of the combinatorial nature of his system of notation: the
primitive concepts in the <i>lingua characteristica</i>, which provide the
foundation for the whole system, are conceived as being the basis of a
combinatorial process of the development of this language. The second idea is
that the whole system is open to a mechanical treatment, which is guaranteed
by the arithmetical nature of the calculus ratiocinator. The whole process of
generation, transformation, and decision in this language could be in
principle performed by a machine. (<a href="bibliography.html#westerhoff"
target="_blank">Westerhoff<span>Westerhoff, Jan C. &quot;Poeta Calculans:
Harsd&ouml;rffer, Leibniz, and the <i>Mathesis Universalis</i>.&quot;
<i>Journal of the History of Ideas</i> 60.3 (1999): 449-67.</span></a> 451;
for Leibniz&#39;s own account of his thinking on the subject, see &quot;On
the General Characteristic&quot; (1679))</p></blockquote>';
      break;
case '216':
      echo '<p>Moreover, Leibniz explicitly describes his &quot;alphabet of
human thoughts,&quot; his <i>mathesis universalis</i>, as a new technology
that extends the human organism.</p><blockquote><p class="blockquote">Once
the characteristic numbers for most concepts have been set up, however, the
human race will have a new kind of instrument which will increase the power
of the mind much more than optical lenses strengthen the eyes and which will
be as far superior to microscopes or telescopes as reason is superior to
sign. The magnetic needle has brought no more help to sailors than this
lodestar will bring to those who navigate the sea of experiments. (<a
href="bibliography.html" target="_blank">Leibniz<span>Leibniz, G. W. &quot;On
the General Characteristic.&quot; In <U>Philosophical Papers and Letters, 2nd
ed.</U> Trans. by Leroy E. Loemker. Boston: D. Holland Publishing Company,
1956.</span></a> 224)</p></blockquote>';
      break;
case '219':
      echo '<p>Of course, in addition seeking a <i>mathesis universalis</i>
encompassing the sum total of human knowledge, Leibniz also invented a binary
system that represents each number as a six-character string of 1s and 0s.
Instead of proceeding in units of ten, then &mdash; from 10 to 20, 20 to 30,
and so on &mdash; Leibniz&#39;s binary system moves by multiples of two, with
0 == 000000, 1 == 000001, 2 == 000010, 3 == 000011. In this way, each string
is calculated by permuting the previous string. </p>';
      break;
case '220':
      echo '<p>For Leibniz, his binary system presents more than mathematical
possibilities: in fact, it embodies the world. As he wrote in a
letter,</p><blockquote><p class="blockquote">For 0 can symbolize the void
which preceded the creation of heaven and earth .... At the beginning of the
first day 1 existed, that is to say God. At the beginning of the second day,
heaven and earth, being created on the first [that is, the end of the first
day]. Finally at the beginning of the seventh day everything already existed.
This is why the last [day] is the most perfect and the Sabbath, for all is
created and complete. Thus 7 is written as 111 without 0. And it is only in
this way of writing by 0&#39;s and 1&#39;s that we see the perfection of the
seventh [day] which is considered holy. And it is even more remarkable that
its character has some relation to the Trinity. (quoted in <a
```

```
href="bibliography.html" target="_blank">Swetz<span>Swetz, F. J.
&amp;quot;Leibniz, the Yijing, and the Religious Conversion of the Chinese.&amp;quot;
<U>Mathematics Magazine</U> 76.4 (2003): 276--291.</span></a> 285-
6)</p></blockquote>';
      break;
case '221':
      echo '<p>Of course, the binary system is now embedded in digital
computing through dyadic data-carrying signals that pulse as 1 (present) or 0
(absent).</b></p>';
      break;
case '222':
      echo '<p>Leibniz invented a calculating machine to rival Blaise
Pascal&amp;#39;s &amp;quot;Pascaline,&amp;quot; invented in 1645, the year before
Leibniz&amp;#39;s birth. Capable not only of adding and subtracting,
Leibniz&amp;#39;s calculator was also intended to multiply, divide, and extract
root numbers (<a href="bibliography.html"
target="_blank">Antognazza<span>Antognazza, Maria Rose. <U>Leibniz: An
Intellectual Biography.</U>New York, NY: Cambridge University Press,
2009.</span></a> 144), and is now seen as a precursor to the modern computer.
Thus Leibniz not only imagined a <i>calculus ratiocinator</i> but, in at
least one sense, constructed one.   </p>';
      break;
case '223':
      echo '<p>Although Leibniz did not believe his binary system was the
universal characteristic he was seeking (<a href="bibliography.html"
target="_blank">Cook and Rosemont<span>Cook, Daniel J. and Henry Rosemont,
Jr. &amp;quot;The Pre-established Harmony between Leibniz and Chinese
Thought.&amp;quot; <U>Journal of the History of Ideas</U 42.2 (April-June 1981):
253-267.</span></a> 264), he often returns to numbers (specifically, Arabic
numerals) as a model for scientific inquiry. He writes: &amp;quot;if we could
explain by them abstract ideas, that would be the most scientific method ...
of unimaginable usefulness (quoted in <a href="bibliography.html"
target="_blank">Swetz<span>Swetz, F. J. &amp;quot;Leibniz, the Yijing, and the
Religious Conversion of the Chinese.&amp;quot; <U>Mathematics Magazine</U> 76.4
(2003): 276--291.</span></a> 283).</p>';
      break;
case '224':
      echo '<blockquote><p class="blockquote">But there can be no doubt that
the general art of combinations or characteristics contains much greater
things than algebra has given, for by its use all our thoughts can be
pictures and as it were, fixed, abridged, and ordered; pictured to others in
teaching them, fixed for ourselves in order to remember them; abridged so
that they may be reduced to a few; ordered so that all of them can be present
in our thinking. ... I believe that when you examine the matter more
seriously, you will agree that this general characteristic will be of
unbelievable value, since a spoken and written language can also be developed
with its aid which can be learned in a few days and will be adequate to
express everything that occurs in everyday practice, and of astonishing value
in criticism and discovery, after the model of numeral characters. (<a
href="bibliography.html" target="_blank">Leibniz to von
Tschirnhaus<span>Leibniz, G. W. &amp;quot;Letter to Walter von Tschirnhaus.&amp;quot;
In <U>Philosophical Papers and Letters, 2nd ed.</U> Trans. by Leroy E.
Loemker. Boston: D. Holland Publishing Company, 1956.</span></a>, May 1678;
193)</p></blockquote>';
      break;
case '225':
      echo '<p>When, then, the French Jesuit Joachim Bouvet wrote to Leibniz
of the Chinese <i>I Ching</i> &amp;mdash; an ancient divination system of 64
```

```
hexagrams composed of six lines each &mdash; both thinkers immediately saw
its connection to Leibniz&#39;s binary system (28 February 1698). Both use a
base-2 system (1s and 0s for Leibniz, broken and solid lines in the <i>I
Ching</i>); both permute these two values across 6 different slots; and both
connect these permuted primitives to a broad cosmology that encompasses the
creation and ordering of the universe (see <a href="bibliography.html"
target="_blank">Antognazza<span>Antognazza, Maria Rose. <U>Leibniz: An
Intellectual Biography.</U>New York, NY: Cambridge University Press,
2009.</span></a> 433-6, <a href="bibliography.html"
target="blank">Cammann<span>Cammann, Schuyler. &Chinese Hexagrams, Trigrams,
and the Binary System. <U>Proceedings fo the American Philosophical
Society</U> 135.4 (December 1991): 576-589.</span></a>, <a
href="bibliography.html" target="_blank">Cook and Rosemont<span>Cook, Daniel
J. and Henry Rosemont, Jr. &quot;The Pre-established Harmony between Leibniz
and Chinese Thought.&quot; <U>Journal of the History of Ideas</U 42.2 (April-
June 1981): 253-267.</span></a>, <a href="bibliography.html"
target="_blank">Ryan<span>Ryan, James A. &quot;Leibniz&#39; Binary System and
Shao Yong&#39;s Yijing.&quot; <U>Philosophy East & West</U> 46
(1996).</span></a>, <a href="bibliography.html"
target="_blank">Swetz<span>Swetz, F. J. &quot;Leibniz, the Yijing, and the
Religious Conversion of the Chinese.&quot; <U>Mathematics Magazine</U> 76.4
(2003): 276--291.</span></a>, <a href="bibliography.html"
target="_blank">Swiderski<span>Swiderski, Richard M. &quot;Bouvet and
Leibniz: A Scholarly Correspondence.&quot; <U>Eighteenth-Century Studies</U>
14.2 (Winter 1980-1): 135-50.</span></a> for further discussion). </p>';
      break;
case '226':
      echo '<p>In a subsequent letter to Leibniz, written on November 1,
1701, Bouchet included a diagram containing all 64 hexagrams laid out in a
circle. As in Harsd&ouml;rffer&#39;s text-generating volvelles, or the wheels
of Ramon Llull&#39;s <i>ars magna</i>, or even the units of the Kabbalah
&mdash; another mystical system which fascinated Leibniz (see <a
href="bibliography.html" target="_blank">Coudert<span>Coudert, Alison.
<U>Leibniz and the Kabbalah.</U> New York: Springer, 1995.</span></a>)
&mdash; the diagram represents the infinite through the systematic
permutation and combination of discrete elements.</p><p>Therefore, much as
Leibniz&#39;s dyads embody a programmatic cosmology, so the<i> I Ching</i>
uses minimalist abstractions to produce an entire world &mdash; both yin and
yang, in all their combinations.</p>';
      break;
case '227':
      echo '<blockquote><p class="blockquote">What is amazing in this
reckoning is that this arithmetic by 0 and 1 is found to contain the mystery
of the lines of an ancient King and philosopher named Fuxi, who is believed
to have lived more than 4000 years ago, and whom the Chinese regard as the
founder of their empire and their sciences. There are several linear figures
attributed to him, all of which come back to this arithmetic, but it is
sufficient to give here the <i>Figure of the Eight Cova</i>, as it is called,
which is said to be fundamental, and to join to them the explanation which is
obvious, provided that one notices, firstly, that a whole line &mdash; means
unity, or 1, and secondly, that a broken line &mdash; means zero, or 0. (<a
href="bibliography.html" target="_blank">Leibniz<span>Leibniz, G. W.
&quot;Explanation of Binary Arithmetic.&quot;</span></a> <a
href="http://www.leibniz-translations.com/binary.htm"
target="_blank">online</a>)</p></blockquote>';
      break;
case '228':
```

```
      echo '<p>More than the obvious similarity between Leibniz&#39;s binary
system and the hexagrams of the <i>I Ching</i>, their historical significance
excited Joachim Bouvet. As a Jesuit missionary living in China, Bouvet read
any cultural syncretism as a confirmation of his ecumenical belief system.
For instance, he speculated that Chinese characters were &quot;the writing
used among the learned before the Flood,&quot; much like Schottel&#39;s
theory of Stammw&ouml;rtter (quoted in <a href="bibliography.html"
target="_blank">Swiderski<span>Swiderski, Richard M. &quot;Bouvet and
Leibniz: A Scholarly Correspondence.&quot; <U>Eighteenth-Century Studies</U>
14.2 (Winter 1980-1): 135-50.</span></a> 138), and sought Christian symbols
in the pre-Christian people, places and narratives of the Chinese (<a
href="bibliography.html" target="_blank">Ryan<span>Ryan, James A.
&quot;Leibniz&#39; Binary System and Shao Yong&#39;s Yijing.&quot;
<U>Philosophy East & West</U> 46 (1996).</span></a> 61, <a
href="bibliography.html" target="_blank">Swiderski<span>Swiderski, Richard M.
&quot;Bouvet and Leibniz: A Scholarly Correspondence.&quot; <U>Eighteenth-
Century Studies</U> 14.2 (Winter 1980-1): 135-50.</span></a> 149). Thus, for
Bouvet, the fact that one of the foremost European thinkers had independently
devised a system that was (in his mind) nearly identical to the <i>I
Ching</i> only proved Christianity to be an all-encompassing Truth
undergirding all knowledge at all times, even before the coming of Christ.
</p>';
      break;
case '229':
      echo '<p>In this way, the syncronicity between  Leibniz&#39;s binary
system and the <i>I Ching</i> transformed both from culturally- and socially-
embedded media &mdash; the former, a product of the scientific discourse
congealing at the end of the seventeenth century in Europe; the latter, an
ancient form of communication between individuals, their present and their
past &mdash; into machines of conversion, re-fitting the Chinese hexagrams to
mechanical Christian worldview. Like the systems themselves, the Leibniz-
Bouvet correspondence isolates discrete cultural systems from each other,
levels them into abstract variables, and recombines them to write a new text,
a new narrative of Chinese and Christian history.</p><p>In short, Bouvet
seeks a new Lullian wheel &mdash; an <i>ars combinatoria</i> to
programmatically convert the Chinese through the mechanical permutation of
symbols.</p>';
      break;
case '230':
      echo '<p>As with baroque Stammw&ouml;rtter theory (which inspired young
Leibniz), Leibniz&#39;s binary system searches for origins as a path to the
future. In other words, it seeks to disappear knowledge into universal
characters, even as the seemingly transparent algorithms and abstractions it
uses can never escape their own ideology. </p>';
      break;
case '231':
      echo '<p>Since his death, Leibniz has been transformed from polymathic,
baroque thinker with an almost mystical worldview to, as Cammann describes
him, one &quot;too rational-minded to pay any attention to astrology or other
aspects of the occult&quot; (<a href="bibliography.html"
target="blank">Cammann<span>Cammann, Schuyler. &Chinese Hexagrams, Trigrams,
and the Binary System. <U>Proceedings fo the American Philosophical
Society</U> 135.4 (December 1991): 576-589.</span></a> 588).  Even Florian
Cramer positions Leibniz &quot;opposite to Kuhlmann&#39;s Kabbalist
cosmology,&quot; writing that he &quot;on the grounds of rational science
transforms Lullism into symbolic logic and an early mechanical calculation
device&quot; (<a href="bibliography.html"
target="_target">Cramer<span>Cramer, Florian. <U>Words Made Flesh: Code,
```

```
Culture, Imagination.</U> Piet Zwart Institute, 2005.</span></a> 48). Yet
from his search for a <i>mathesis universalis</i>, to his &quot;alphabet of
human thoughts,&quot; to his binary cosmology and its relationship to the
Chinese <i>I Ching</i>, Leibniz shows a deep sympathy systems that seem now
to us &quot;non-scientific&quot; or &quot;non-rational.&quot; In fact, he
bases many of his epistemological claims on an <i>ars combinatoria</i> rooted
in the very Lullism and Kabbalah Cramer sees as antithetical. The divine and
the programmatic, or the aesthetic and the rational, do not always undermine
each other but, in Leibniz&#39;s world, converge.</p>';
      break;
case '232':
      echo '<p>Proteus poetry becomes a major source for much of
Leibniz&#39;s work in the <i>Dissertatio</i>, for obvious reasons. By
reducing language (i.e. meaning) to a set of primitives that can combine to
generate new meaning, proteus verse is a model for Leibniz&#39;s
&quot;alphabet of human thoughts.&quot; </p>';
      break;
case '233':
      echo '<blockquote><p class="blockquote">It is not always the case that
the variations of a proteus verse have the same meaning, nor is a proteus
verse always defined in this way. Leibniz, nevertheless, considers it part of
the meaning of &quot;proteus verse&quot; that its variations have the same
sense. Regardless of the way the elements of a proteus verse are recombined,
the meaning will be left unchanged. This is particularly interesting because
this feature of a synthesis of elements is mirrored in one of Leibniz&#39;s
ideas on analysis. In constructing the <i>lingua characteristica</i> as a
part of the <i>mathesis universalis</i>, Leibniz considers it as necessary to
analyze concepts into their most simple components, the primitive concepts or
&quot;alphabet of human thoughts." (<a href="bibliography.html#westerhoff"
target="_blank">Westerhoff<span>Westerhoff, Jan C. &quot;Poeta Calculans:
Harsd&ouml;rffer, Leibniz, and the <i>Mathesis Universalis</i>.&quot;
<i>Journal of the History of Ideas</i> 60.3 (1999): 449-67.</span></a>
458)</p></blockquote>';
      break;
case '234':
      echo '<p>Leibniz&#39;s <i>Dissertatio de arte combinatoria</i> is
deeply influenced by Llull&#39;s <i>ars</i>, such that it could be described
as a fundamentally Lullian work. As in the <i>ars</i>, Leibniz breaks down
triads of constants, or &quot;wholes,&quot; into a set of three dyads, such
that two types of variation are possible from every whole composed of
situated parts (see <a href="bibliography.html"
target="_blank">Leibniz<span>Leibniz, G. W. &quot;Dissertation on the Art of
Combination.&quot; In <U>Philosophical Papers and Letters, 2nd ed.</U> Trans.
by Leroy E. Loemker. Boston: D. Holland Publishing Company, 1956.</span></a>
77). And, as in the Lullian Art, Leibniz seeks to use this relationship
between wholes and parts, in which combination provides the mechanism for
traversing all different possibilities, as a way of explaining the
world.</p>';
      break;
case '235':
      echo '<blockquote><p class="blockquote">That is what Leibniz explains
in an extraordinary piece of writing: a flexible or an elastic body still has
cohering parts that form a fold, such that they are not separated into parts
of parts but are rather divided to infinity in smaller and smaller folds that
always retain a certain cohesion. Thus a continuous labyrinth is not a line
dissolving into independent points, as flowing sand might dissolve into
grains, but resembles a sheet of paper divided into infinite folds or
separated into bending movements, each one determined by the consistent or
```

```
conspiring surroundings. (<a href="bibliography.html"
target="_blank">Deleuze<span>Deleuze, Gilles. <U>The Fold: Leibniz and the
Baroque.</u> Tran. by Tom Conley. Minneapolis: University of Minnesota Press,
1993.</span></a> 6)</p></blockquote>';
      break;
case '250':
      echo '<p>In 1651, five years after Leibniz was born in Saxony, Quirinus
Kuhlmann was born several hundred miles to the southeast in Brauslau
(Wroclaw), Lower Silesia. Also like Leibniz, Kuhlmann&#39;s father died when
he was very young, leaving his pious Lutheran mother to raise the sickly,
stuttering young Quirinus alone.</p>';
      break;
case '251':
      echo '<p>Growing up in Lower Silesia, one of the largest influences on
Kuhlmann&#39;s early life was Martin Opitz (1597-1639), a celebrated baroque
poet of G. P Harsd&ouml;rffer&#39;s generation, also born in Lower Silesia.
As Beare writes, &quot;the students of the two Breslau schools recited Opitz
poems and panegyrics to Opitz in the market place&quot; as part of a yearly
celebration that attracted a parade of well-known poets, including Andreas
Gryphius and Daniel Caspar (<a href="bibliography.html"
target="_blank">Beare<span>Beare, Robert L. &quot;Quirinus Kuhlmann: The
Religious Apprenticeship.&quot; <U>PMLA</U> 68.4 (1953): 828-62.</span></a>
834). By age seventeen, Kuhlmann had published his first book of verse,
<i>Unsterbliche Sterblichkeit</i> (1668).</p>';
      break;
case '252':
      echo '<p>In 1669, a year after publishing his first (and highly
acclaimed) book of poems, illness struck Quirinus Kuhlmann. Later, he claimed
to have had prophetic visions during this period &mdash; images of Christ,
the Devil, and two angels in the guise of flashing lights which flared in his
left peripheral vision for many years (see <a href="bibliography.html"
target="_blank">Beare<span>Beare, Robert L. &quot;Quirinus Kuhlmann: The
Religious Apprenticeship.&quot; <U>PMLA</U> 68.4 (1953): 828-62.</span></a>
838). Famously, he continued to excite similar visions by covering the walls
of his room with reflective papers.</p>';
      break;
case '253':
      echo '<p>From an early age, Athanasius Kircher&#39;s work fascinated
Quirinus Kuhlmann &mdash; particularly his treatise on Ramon Llull&#39;s
<i>ars magna</i>, or art of combinations. Like Leibniz,
Kuhlmann</p><blockquote><p class="blockquote">planned, using this system, to
write a series of encyclopedic studies, some in several volumes, covering
nineteen fields of knowledge, the sum total of human knowledge at the time.
Nothing came of this plan, but his failure to carry it out is lamented by
later writers, although they distrusted his reliance upon divine impulse. (<a
href="bibliography.html" target="_blank">Beare<span>Beare, Robert L.
&quot;Quirinus Kuhlmann: The Religious Apprenticeship.&quot; <U>PMLA</U> 68.4
(1953): 828-62.</span></a> 840)</p></blockquote>';
      break;
case '254':
      echo '<p>Like Leibniz, Kuhlmann also encountered Llull through the work
of Georg Philipp Harsd&ouml;rffer and was very likely familiar with his
Denckring, a mechanism for generating correct German words, rhymes and
poems.</p>';
      break;
case '255':
      echo '<p>Perhaps Quirinus Kuhlmann&#39;s most fascinating work is in
his <i>Himmlische Libes-k&uuml;sse, </i>or Heavenly Love-kisses, published
```

```
two years after his prophetic visions of Christ. Within this collection of
fifty religious-themed sonnets, Kuhlmann includes one permutation poem: XLI,
the 41st, entitled &quot;Vom Wechsel menschlicher Sachen,&quot; or &quot;On
the Variations of Human Matters.&quot;</p><p>The first twelve lines are
composed of sixteen monosyllabic words each. The first and last words in
these lines, as well as the last couplet in the sonnet, must remain stable to
preserve the rhyme scheme; the middle words, totaling 169, are
permutable.</p>';
      break;
case '256':
      echo '<p>In his commentary on the sonnet, Kuhlmann stops counting its
permutations at fifty. However, as Florian Cramer points out, the number of
possible sonnets generated by &quot;On the Variations of Human Matters&quot;
is not 50 but 3.399*10<sup>117</sup> (<a href="bibliography.html"
target="_target">Cramer<span>Cramer, Florian. <U>Words Made Flesh: Code,
Culture, Imagination.</U> Piet Zwart Institute, 2005.</span></a> 47). </p>';
      break;
case '257':
      echo '<blockquote><p class="blockquote">It functions as a world
machine, permuting, both in the meaning of its words and in its combinatory
mechanics, an inventory of the micro- and macrocosm.  Its own couplet, and
its afterword, claims that to grasp the principle of the permutation of
things means to have wisdom of the world.  This alludes to the both to the
use of Solomon&#39;s proverbs in the poem and adaptions of Solomon&#39;s Song
Celestial in the volume in which it appeared. Through this intertextuality,
the poem renders itself a Solomonic machine. It is a computational reverse
engineering of Solomon&#39;s wisdom, considering the proverbs as they are
written in the Bible the fragmentary output of an occult machine. (<a
href="bibliography.html" target="_target">Cramer<span>Cramer, Florian.
<U>Words Made Flesh: Code, Culture, Imagination.</U> Piet Zwart Institute,
2005.</span></a> 47)</p></blockquote>';
      break;
case '258':
      echo '<p>Quirinus Kuhlmann&#39;s permutable sonnet participates in the
mechanical, programmatic epistemology of the German baroque, so well
articulated by his contemporary Leibniz. That is: it reduces its verse to a
finite set of primitives, and algorithmically permutes those primitives to
generate new texts, new poems, and thereby new meaning. For Kuhlmann, to know
is to accrete and accumulate, not to reduce and refine.</p>';
      break;
case '259':
      echo '<p>Yet, unlike Harsd&ouml;rffer&#39;s Denckring, Kuhlmann&#39;s
sonnet XLI is legible, in both the obvious and etymological sense of the
word. I can reproduce a line from his poem on the screen for you, because
that line exists, regardless of how one might manipulate it after the moment
of reading.</p><p>What would I reproduce of the Denckring? Or the Maria
Stella? Or August de Campos&#39; &quot;SOS&quot;? What is the text one has
&quot;read&quot; &mdash; and what has been already &quot;written&quot;?</p>';
      break;
case '260':
      echo '<p>In short, Quirinus Kuhlmann does not instruct the reader to
physically cut up his poem, as Burroughs and Gysin do; nor has he already cut
it up, as the women of Little Gidding have done to the Bible. For Kuhlmann,
the manipulation &mdash; the handling of the text &mdash; is mental, not
material &mdash; locked in the mind, not on the page. And the Word is not
matter, but spirit, spinning off into the void. </p>';
      break;
case '261':
```

```
      echo '<p>History now monumentalizes Leibniz as the great seventeenth-
century rationalist, too mathematically-inclined &amp;quot;to pay any attention
to astrology or other aspects of the occult&amp;quot; (<a
href="bibliography.html" target="blank">Cammann<span>Cammann, Schuyler.
&Chinese Hexagrams, Trigrams, and the Binary System. <U>Proceedings fo the
American Philosophical Society</U> 135.4 (December 1991): 576-589.</span></a>
588), while brushing away Quirinus Kuhlmann as a delusional and possibly
mentally deranged mystic. (Worse still, Kuhlmann&amp;#39;s  role in seventeenth-
century German poetry is often altogether forgotten; see, for instance, <a
href="bibliography.html" target="_blank">Beare<span>Beare, Robert L.
&amp;quot;Quirinus Kuhlmann: The Religious Apprenticeship.&amp;quot; <U>PMLA</U> 68.4
(1953): 828-62.</span></a> 828.)</p><p>Yet, born only five years and several
hundred miles apart, both thinkers sought <i>ars combinatoria</i> as a
computer for organizing &amp;mdash; and, in doing so, generating &amp;mdash;
knowledge in a society newly overwhelmed with printed texts.</p>';
      break;
case '262':
      echo '<blockquote><p class="blockquote">In 1674, three years after his
permutational sonnet, Quirinus Kuhlmann published his correspondence with
Athanasius Kircher in a book <i>epistolae duae</i>. It documents an early
debate about automatically generated art and its cognitive limitations. In
his letters, Kuhlmann rejects a purely technical application of Llull&amp;#39;s
combinatorics, claiming that &amp;quot;knowing Lullus does not mean to have
knowledge in the <i>alphabetum</i> of his Ars in order to build syllogisms
with it, but to grasp the true power from the universal book of nature that
is hidden underneath it, and apply it to everything.&amp;quot; (<a
href="bibliography.html" target="_target">Cramer<span>Cramer, Florian.
<U>Words Made Flesh: Code, Culture, Imagination.</U> Piet Zwart Institute,
2005.</span></a> 105)</p></blockquote>';
      break;
case '263':
      echo '<p>Quirinus Kuhlmann was not the first to write what has become
known as proteus poetry. Optatianus Porfyrius, writing in the fourth century
AD, composed twenty stanzas by permuting the elements of one (see <a
href="bibliography.html" target="_blank">Hallyn<span>Hallyn, Fernand, and
Roxanne Lapidus. &amp;quot;&amp;#39;A Light-Weight Artifice&amp;#39;: Experimental Poetry
in the 17th Century.&amp;quot; <U>SubStance</U> 22.2/3, Issue 71/72: Special
Issue: EpistÈmocritique (1993): 289-305.</span></a> 292, <a
href="bibliography.html" target="_target">Cramer<span>Cramer, Florian.
<U>Words Made Flesh: Code, Culture, Imagination.</U> Piet Zwart Institute,
2005.</span></a> 44), and there is a long history of <i>centos</i> &amp;mdash;
from the Latin word <i>cento</i>, an overcoat stitched together from patches
of cloth &amp;mdash; which combine phrases from difference source texts to
produce Latin verse.</p><p>In 1561, Julius Caesar Scalinger, looking to the
Latin and Greek traditions of permutation poetry, created his own combinatory
verse by permuting elements of the line <i>Perfide sperasti divos falleri
Proteu</i>. Published in his <i>Poetices Libri Septem</i> (II.30), he called
his poem &amp;quot;Proteus&amp;quot; after the shape-shifting sea god (<a
href="bibliography.html" target="_target">Cramer<span>Cramer, Florian.
<U>Words Made Flesh: Code, Culture, Imagination.</U> Piet Zwart Institute,
2005.</span></a> 44, <a href="bibliography.html"
target="_blank">Higgins<span>Higgins, Dick. <U>Pattern Poetry: Guide to an
Unknown Literature.</U> Albany: State University of New York Press,
1987.</span></a> 183).</p>';
      break;
case '264':
      echo '<p>In 1626, Thomas Lansius composed a couplet of monosyllabic
words, each one permutable except <i>bona</i> and
```

```
<i>mala</i>:</p><blockquote><p class="blockquote"><i>Les, Rex, Grex, Res,
Spes, Jus, Tus, Sal, Sol, (bona) Lux, Laus:<br/>Mars, Mors, Sors, LIs Vis,
Styx, Pus, Nox, Fex, (mala) Crux, Fraus.</i></p></blockquote><p>Lansius&#39;
poem &mdash; containing 39,916 possible variations (<a
href="bibliography.html" target="_blank">Bernoulli<span>Bernoulli, Jacob.
<U>The art of conjecturing, together with Letter to a friend on sets in court
tennis.</U> Trans. by Edith Dudley Sylla. Baltimore, MD: Johns Hopkins
University Press, 2006.</span></a>197) &mdash; is reprinted in Johann
Heinrich Alsted&#39;s famous encyclopedia, influenced by the Lullian Art (<a
href="bibliography.html" target="_target">Cramer<span>Cramer, Florian.
<U>Words Made Flesh: Code, Culture, Imagination.</U> Piet Zwart Institute,
2005.</span></a> 45).</p>';
      break;
case '265':
      echo '<p>In 1616, Bernard Bauhuis published a book of Latin epigrams,
<i>Jesu epigrammatum selectorum libri</i>, containing two permutational
lines, the first having (as Bauhuis writes) &quot;as many faces, in fact, as
there are stars in the heavens (since it can be transformed one thousand and
twenty-two times, while its meaning is retained and the rules of epic poetry
are observed)&quot; (Bauhuis, quoted in <a href="bibliography.html"
target="_blank">Hallyn<span>Hallyn, Fernand, and Roxanne Lapidus.
&quot;&#39;A Light-Weight Artifice&#39;: Experimental Poetry in the 17th
Century.&quot; <U>SubStance</U> 22.2/3, Issue 71/72: Special Issue:
EpistÈmocritique (1993): 289-305.</span></a> 290).</p><blockquote><p
class="blockquote"><i>Tot tibi sunt dotes, Virgo, quot sidera
caelo.</i></p></blockquote><p>As Hallyn describes, the next year Erycius
Puteanus (Henri Van de Putte) published the <i>Pietatis Thaumata</i>, in
which he writes out all 1,022 variations, filling thirty-six pages (<a
href="bibliography.html" target="_blank">Hallyn<span>Hallyn, Fernand, and
Roxanne Lapidus. &quot;&#39;A Light-Weight Artifice&#39;: Experimental Poetry
in the 17th Century.&quot; <U>SubStance</U> 22.2/3, Issue 71/72: Special
Issue: EpistÈmocritique (1993): 289-305.</span></a> 291).</p>';
      break;
case '266':
      echo '<blockquote><p class="blockquote">Puteanus does not fail to play
up the mythological theme of Proteus. He  recalls that because of  the
multiplicity  of Proteus&#39;s changing appearances, Heraclides identifies
him with the original Chaos, after which Providence separated the elements
and assembled the distinct forms of the world.  But the symbolism of Proteus
is itself protean: Nature itself, in its variety, is a Proteus, as is Christ,
as well as Bauhusius, who is equally a new  Pythagoras and an Orpheus. (<a
href="bibliography.html" target="_blank">Hallyn<span>Hallyn, Fernand, and
Roxanne Lapidus. &quot;&#39;A Light-Weight Artifice&#39;: Experimental Poetry
in the 17th Century.&quot; <U>SubStance</U> 22.2/3, Issue 71/72: Special
Issue: EpistÈmocritique (1993): 289-305.</span></a> 293)</p></blockquote>';
      break;
case '267':
      echo '<p>Many others generated proteic verse during the seventeenth
century, including Georg Keppisius (<a href="bibliography.html"
target="_blank">Hallyn<span>Hallyn, Fernand, and Roxanne Lapidus.
&quot;&#39;A Light-Weight Artifice&#39;: Experimental Poetry in the 17th
Century.&quot; <U>SubStance</U> 22.2/3, Issue 71/72: Special Issue:
EpistÈmocritique (1993): 289-305.</span></a> 297), Pietro Francesco Passerini
(<a href="bibliography.html" target="_blank">Higgins<span>Higgins, Dick.
<U>Pattern Poetry: Guide to an Unknown Literature.</U> Albany: State
University of New York Press, 1987.</span></a> 183), Carolus of Goldsten and
Johan-Philipp Ebel (<a href="bibliography.html"
target="_blank">Hallyn<span>Hallyn, Fernand, and Roxanne Lapidus.
```

```
&quot;&#39;A Light-Weight Artifice&#39;: Experimental Poetry in the 17th
Century.&quot; <U>SubStance</U> 22.2/3, Issue 71/72: Special Issue:
EpistÈmocritique (1993): 289-305.</span></a> 298). Georg Philipp
Harsd&ouml;rffer also experimented with permutational verse.</p>';
      break;
case '300':
      echo '<p>For the thirteenth-century Majorcan monk Ramon Llull, as for
Quirinus Kuhlmann, the art of combination began with a
vision.</p><blockquote><p class="blockquote">Ramon went up a certain mountain
not far from his home, in order to contemplate God in greater tranquility.
When he had been there scarcely a full week, it happened that one day while
he was gazing intently heavenward the Lord suddenly illuminated his mind,
giving him the form and method for writing the aforementioned book against
the errors of the unbelievers.</p><p class="blockquote">Giving thanks to the
Almighty, he came down from the mountain and returned at once to the above-
mentioned abbey, where he began to plan and write the book in question,
calling it at first the <i>Ars major</i>, and later on the <i>Ars
generalis</i>. (from the <i>Vita coaetanea</i>, in <a
href="bibliography.html" target="_blank">Llull<span>Llull, Ramon. <u>Selected
Works of Ramon Llull.</u> Vol. 1. Ed. and trans. by Anthony Bonner.
Princeton, NJ: Princeton University Press, 1985.</span></a>
22)</p></blockquote>';
      break;
case '301':
      echo '<p>Just as Harsd&ouml;rffer&#39;s Denckring and Leibniz&#39;s
<i>mathesis universalis</i> reflect the political milieu of the Thirty  Years
War, Ramon Llull&#39;s <i>ars</i> is, in some sense, a microcosm of Majorcan
culture during the thirteenth century, representing a fundamental paradox
&mdash;</p><blockquote><p class="blockquote">&mdash; that of a figure coming
from a small island in the western Mediterranean and from what nowadays is
considered a minority culture, developing one of the most universalist of
systems, one which he presented to popes, kings, sultans, and universities in
Spain, France, Italy, and North Africa during his lifetime, and which brought
him extraordinary fame throughout Europe in the Renaissance. (<a
href="bibliography.html" target="_blank">Bonner<span>Bonner, Anthony.
&quot;Introduction.&quot; <u>Selected Works of Ramon Llull.</u> Vol. 1. Ed.
and trans. by Anthony Bonner. Princeton, NJ: Princeton University Press,
1985.</span></a> 3)</p></blockquote>';
      break;
case '302':
      echo '<p>As Bonner points out, during the thirteenth century an
imperialist push from Catalonia into Greece and Turkey &quot;gave the
Catalans a sense of their own identity&quot; that was &quot;tied up not only
with a feeling that they were carrying out a divine destiny, but also with a
feeling for their language as an expression of that destiny&quot; (<a
href="bibliography.html" target="_blank">Bonner<span>Bonner, Anthony.
&quot;Introduction.&quot; <u>Selected Works of Ramon Llull.</u> Vol. 1. Ed.
and trans. by Anthony Bonner. Princeton, NJ: Princeton University Press,
1985.</span></a> 7-8). Thus Ramon Llull &mdash; born in Palma, Majorca
&mdash; was writing his ars in Catalan during a time in which his native
tongue represented a divine right &mdash; in fact, when it itself was a kind
of machine for conversion.</p>';
      break;
case '303':
      echo '<p>The combinatory system of what would become known as the
Lullian Art, or the <i>ars magna</i> of Ramon Llull, is rooted in his earlier
manuscripts, particularly the <i>Ars demonstrativa</i> (1283). In it, Llull
takes five letters: A, S, T, V, and X, each representing some Figure (in
```

```
addition to other Figures regarding, for instance, the Principles of Law,
which we&#39;ll put aside for the sake of simplicity). Y and Z represent,
respectively, truth and falsehood, while the letters B through R are copied
as a principle of each figure A, S, T, V and X, constructing binary
combinations such as AB, AC, AD; SB, SC, SD; and so on. Thus through the
constrained permutation of all twenty-three letters of the medieval alphabet,
Llull demonstrates how to &quot;know and love A,&quot; or God (<a
href="bibliography.html" target="_blank">Llull<span>Llull, Ramon. <u>Selected
Works of Ramon Llull.</u> Vol. 1. Ed. and trans. by Anthony Bonner.
Princeton, NJ: Princeton University Press, 1985.</span></a> <i>Ars
demonstrativa</i> 415).</p>';
      break;
case '304':
      echo '<p>The Figure S, representing the intelligent soul, takes on a
special role as, in the words of Anthony Bonner, &quot;a connective or
relational figure&quot; that &quot;works externally, connecting [the
Art&#39;s] components to the outside world, to the &#39;artist&#39; or to
those to whom the Art is directed&quot; (<a href="bibliography.html"
target="_blank">Bonner<span>Bonner, Anthony. &quot;Introduction.&quot;
<u>Selected Works of Ramon Llull.</u> Vol. 1. Ed. and trans. by Anthony
Bonner. Princeton, NJ: Princeton University Press, 1985.</span></a> 309).
Figure T, then, acts as the mechanism that allows S to relate to the external
world. As Llull writes, &quot;just as the artisan fashions the artificial
form by means of his tools, so S does what it does in this Art by means of T,
introducing it objectively into its powers, and introducing it into the
compartments of the figures, insofar was it can be admitted into them&quot;
(<a href="bibliography.html" target="_blank">Llull<span>Llull, Ramon.
<u>Selected Works of Ramon Llull.</u> Vol. 1. Ed. and trans. by Anthony
Bonner. Princeton, NJ: Princeton University Press, 1985.</span></a> <i>Ars
demonstrativa</i> 327). </p>';
      break;
case '305':
      echo '<blockquote><p class="blockquote">The roles of the figures can
perhaps be clarified by a very loose analogy with a computer: if A, V, X, Y,
Z, the Elemental Figure, and the three Figures of Theology, Philosophy, and
Law contain the basic data, Figure T provides the processing unit, and Figure
S constitutes the terminal or control unit assuring correct access to the
data and processing. (<a href="bibliography.html"
target="_blank">Bonner<span>Bonner, Anthony. &quot;Introduction.&quot;
<u>Selected Works of Ramon Llull.</u> Vol. 1. Ed. and trans. by Anthony
Bonner. Princeton, NJ: Princeton University Press, 1985.</span></a>
310)</p></blockquote>';
      break;
case '306':
      echo '<p>In <i>Ars demonstrativa</i>, Llull goes on to develop other
Figures that specify particular relationships between different elements.
Only the Ninth Figure, however, concerns us. </p><blockquote><p
class="blockquote">It is composed of six revolving circles along with a wheel
situated in the middle in which there is a triangle representing the five
triangles of T. It is made up of circles, each inside the other, like weights
inserted one in another, with a pin in the middle, which keeps the circles in
place. ... This figure contains and includes all the other figures of this
Art, as is explained above; moreover, all the compartments of the other
figures can be formed by revolving the circles of this figure in the right
way[.] (<a href="bibliography.html" target="_blank">Llull<span>Llull, Ramon.
<u>Selected Works of Ramon Llull.</u> Vol. 1. Ed. and trans. by Anthony
Bonner. Princeton, NJ: Princeton University Press, 1985.</span></a> <i>Ars
demonstrativa</i> 333-34)</p></blockquote>';
```

```
      break;
case '307':
      echo '<p>This figure, the Ninth in Llull&#39;s <i>Ars
demonstrativa</i>, represents an early attempt to mechanize the ars. Here,
instead of tracing Llull&#39;s logic through dispersed rules and tables of
combinations, the user may operate a single machine whose laws simply are the
physical relationships between the spinning paper discs. In this way, reading
a set of interlocking letters writes a representation of the world, just as
writing their combinations becomes an act of exploration, a voyage, a
reading. </p>';
      break;
case '308':
      echo '<p>The system outlined in the <i>Ars demonstrativa</i> proved too
complicated and was derided by the intellectual community in Paris, where
Llull was teaching. After returning to Montpellier, he set to work reducing
his system to only four figures, thereby (as later described in his dictated
autobiography) &quot;eliminating &mdash; or rather disguising, because of the
weakness of human intellect which he had witness in Paris &mdash; twelve of
the sixteen figures that had formerly appeared in his Art&quot; (from the
<i>Vita coaetanea</i>, in <a href="bibliography.html"
target="_blank">Llull<span>Llull, Ramon. <u>Selected Works of Ramon
Llull.</u> Vol. 1. Ed. and trans. by Anthony Bonner. Princeton, NJ: Princeton
University Press, 1985.</span></a> <i>Ars demonstrativa</i>  22). The vision
passed to him from God proved too complicated for humans; so he simplified it
to an <i>Ars inventiva veritatis</i>, leading to his two most influential
works: the <i>Ars brevis</i> (1308) and the <i>Ars generalis ultima</i>
(1305-1308).</p>';
      break;
case '309':
      echo '<p>In the <i>Ars brevis</i>, Ramon Llull reduces the twelve
figures of his earlier system to four, and the twenty-three letters to nine,
B through K.</p><p>The First Figure, A (or God), takes letters B through K
&mdash; representative of the nine Divine Dignities (or attributes), such as
goodness, greatness, eternity, and so on &mdash; and inscribes them around
the edge of a circle &quot;to show that any subject can become a predicate
and vice versa, as when one says, &#39;goodness is great,&#39; &#39;greatness
is good,&#39; and so on&quot; (<a href="bibliography.html"
target="_blank">Llull<span>Llull, Ramon. <u>Selected Works of Ramon
Llull.</u> Vol. 1. Ed. and trans. by Anthony Bonner. Princeton, NJ: Princeton
University Press, 1985.</span></a> <i>Ars demonstrativa</i>  <i>Ars
brevis</i> 582). As Llull argues, since &quot;everything that exists is
implicit in the principles of this figure,&quot; then &quot;whatever exists
is reducible to the above-mentioned principles&quot; (<a
href="bibliography.html" target="_blank">Llull<span>Llull, Ramon. <u>Selected
Works of Ramon Llull.</u> Vol. 1. Ed. and trans. by Anthony Bonner.
Princeton, NJ: Princeton University Press, 1985.</span></a> <i>Ars brevis</i>
583).</p>';
      break;
case '310':
      echo '<p>The Second Figure consists of relationships between different
elements, represented a set of three nesting triangles: the first showing
difference, concordance and contrariety; the second showing the beginning,
middle and end of all things; and the third showing the relationships of
majority, equality and minority. As in the <i>Ars demonstrativa</i>, this
Figure (&quot;T&quot;) serves the First Figure as its constraint or
&quot;tool.&quot; As Lull writes: &quot;by joining this figure to the first,
the intellect acquires knowledge. And because this figure is general,
therefore the intellect becomes general&quot; (<a href="bibliography.html"
```

```
target="_blank">Llull<span>Llull, Ramon. <u>Selected Works of Ramon
Llull.</u> Vol. 1. Ed. and trans. by Anthony Bonner. Princeton, NJ: Princeton
University Press, 1985.</span></a> <i>Ars brevis</i> 585). </p>';
      break;
case '311':
      echo '<p>The Third Figure is a table of the 36 possible non-repeating
combinations of the two sets of B through K represented in the First and
Second Figure, and &quot;is meant to show that any principle can be
attributed to any of the others&quot; such &quot;that the intellect may know
each principle in terms of all the others, and be enabled to deduce many
arguments from a single proposition&quot; (<a href="bibliography.html"
target="_blank">Llull<span>Llull, Ramon. <u>Selected Works of Ramon
Llull.</u> Vol. 1. Ed. and trans. by Anthony Bonner. Princeton, NJ: Princeton
University Press, 1985.</span></a> <i>Ars brevis</i> 586). Thus &#39;goodness
is great&#39;, &#39;goodness is enduring&#39;, &#39;goodness is
powerful&#39;, &#39;goodness is different&#39;, &#39;goodness is
concordant&#39;, and so on. On this Figure, Llull sets the condition that one
dyad can never be contrary to another, so that all accord in their
conclusion.</p>';
      break;
case '312':
      echo '<p>The fourth and final figure of Ramon Llull&#39;s ars is his
combinatory mechanism: a set of three nesting discs, each inscribed with the
letters B through K, such that rotating them against each other generates a
three-letter code &mdash; say, B C D &mdash; and therefore six dyads or (as
Llull terms them) &quot;conditions&quot; that relate, for instance, B to C
and B to D, then C to B and C to D, then D to B and D to C.</p><p>After
acquiring these six conditions, &quot;the intellect acquires another six, by
turning the smallest circle, putting its E where its D was, opposite the C on
the middle circle,&quot; thereby forming the code B C E and a new set of E-
related conditions (<a href="bibliography.html"
target="_blank">Llull<span>Llull, Ramon. <u>Selected Works of Ramon
Llull.</u> Vol. 1. Ed. and trans. by Anthony Bonner. Princeton, NJ: Princeton
University Press, 1985.</span></a> <i>Ars brevis</i> 601). In this way,
spinning the discs spits out all possible combinations of the divine
attributes in relation to the order of the world.</p>';
      break;
case '313':
      echo '<blockquote><p class="blockquote">From what we have said above,
the artist can formulate questions concerning the Hundred Forms and solve
them in accordance with how the questions are variously treated and derived
... And thus the intellect learns in what way it can be most general in
formulating many questions and solving them by the method indicated in the
Evacuation of the Third Figure and in the Multiplication of the Fourth
Figure. As a result, who could possibly enumerate all the questions and
solutions that could be thus formulated? (<a href="bibliography.html"
target="_blank">Llull<span>Llull, Ramon. <u>Selected Works of Ramon
Llull.</u> Vol. 1. Ed. and trans. by Anthony Bonner. Princeton, NJ: Princeton
University Press, 1985.</span></a> <i>Ars brevis</i> 643)</p></blockquote>';
      break;
case '314':
      echo '<p>As Bonner points out, &#39;ars&#39; is the &quot;usual
scholastic translation of the Greek &tau;&epsilon;&chi;&nu;&eta;, meaning a
&quot;technique&quot; &mdash; &quot;not a body of doctrine, but a
system&quot; or &quot;structure&quot; (<a href="bibliography.html"
target="_blank">Bonner<span>Bonner, Anthony. &quot;Introduction.&quot;
<u>Selected Works of Ramon Llull.</u> Vol. 1. Ed. and trans. by Anthony
Bonner. Princeton, NJ: Princeton University Press, 1985.</span></a> 62). Of
```

```
course, &tau;&epsilon;&chi;&nu;&eta; is also the root of the English word
&#39;technology&#39;. More than a system or structure, even, Ramon Llull
develops a new technology: a tool for managing and, in turn, automatically
manipulating the web of interlocking concepts evident in the external
world.</p>';
      break;
case '315':
      echo '<p>Today, our relationship to Llull&#39;s <i>Ars magna</i> is
always already refracted through Renaissance personalities such as Bernard de
Lavinheta, Heinrich Cornelius Agrippa von Nettesheim and Giordano Bruno, who
transformed the Lullian Art into a form of alchemical Hermeticism, and Llull
himself into a Kabbalist mystic. As Rossi writes, &quot;the interest in the
cabbala, in universal and artificial writing, in the discovery of the primary
constituent principles of all possible knowledge, in the art of memory, and
the continual appeal to logic understood as the &#39;key&#39; capable of
opening the secrets of reality: all these themes appeared inextricably
connected with the revival of Lullism in the Renaissance&quot; (<a
href="bibliography.html" target="_blank">Rossi<span>Rossi, Paolo. <U>Logic
and the Art of Memory: The Quest for a Universal Language.</U> Trans. by
Stephen Clucas. Continuum International Publishing Group, 2006.</span></a>
41).</p>';
      break;
case '316':
      echo '<p>It is not our goal here to trace Llull&#39;s journey the
through fifteenth, sixteenth and seventeenth centuries; those paths have
already been trod (see, for example, <a href="bibliography.html"
target="_blank">Rossi<span>Rossi, Paolo. <U>Logic and the Art of Memory: The
Quest for a Universal Language.</U> Trans. by Stephen Clucas. Continuum
International Publishing Group, 2006.</span></a>, <a href="bibliography.html"
target="_blank">Yates<span>Yates, Frances. <U>The Art of Memory.</U> Chicago,
IL: University of Chicago Press, 1966.</span></a>). Suffice to say the very
same figures who imagined universal languages and encyclopedias to both store
and generate knowledge &mdash; including Heinrich Cornelius Agrippa von
Nettesheim and Johann Heinrich Alsted &mdash; also expressed an interest in
the Lullian Art, as did those who experimented with permutational poetry,
such as Bernard de Lavinheta. In doing so, Renaissance Lullists shrouded  a
thirteenth-century machine for converting Muslims &mdash; that is, for
bringing them to the light &mdash; with dark shades of mysticism; and a
technology for cultural translation became a mechanism for generation,
spitting out not reflections of the world but text, the language of babble.
</p>';
      break;
case '317':
      echo '<blockquote><p class="blockquote">Lullist combinatorics changed
its character in the 17th century from a theological device to a generative
classificatory system of knowledge. Before Diderot and d&#39;Alembert and
their revolutionary reinvention of the encyclopedia in the late 18th century,
knowledge in encyclopedias was not structured arbitrarily by the alphabet,
but in a systematic order of things according to their place in a cosmology.
Lullist combinatorics allowed the generation of complex hierarchical systems
for knowledge classification through the exhaustive combination of
categories. (<a href="bibliography.html" target="_target">Cramer<span>Cramer,
Florian. <U>Words Made Flesh: Code, Culture, Imagination.</U> Piet Zwart
Institute, 2005.</span></a> 39)</p></blockquote>';
      break;
case '318':
      echo '<blockquote><p class="blockquote">These apocryphal writings, and
other commentaries on, or epitomes of, Llull&#39;s Art, tend to abandon
```

```
Llull&#39;s formal, exemplarist approach to natural language in favor of a
mechanistic and generative model of discourse more compatible with
conventional rhetorical or dialectical precepts. Rather than establish a
system of formal logic based on explicit ontological categories, they seek to
perfect the discursive machinery of the Lullian program of inquiry. (<a
href="bibliography.html" target="_blank">Johnston<span>Johnston, Mark D.
&quot;The Reception of the Lullian Art, 1450-1530.&quot; <U>Sixteenth Century
Journal</U> 12.1 (1981): 31-48.</span></a> 47-8)</p></blockquote>';
      break;
case '319':
      echo '<p>It is worth pausing over at least one seventeenth-century book
on the Lullian Art: Athanasius Kircher&#39;s <i>Ars magna sciendi sive
combinatoria</i>, published in 1669, three years after Leibniz&#39;s
dissertation on <i>ars combinatoria</i>. Kircher &mdash; a student of Chinese
characters, cryptography and hieroglyphics, as well as the author of a
polygraphy machine &mdash; set about reforming Llull&#39;s work by linking
each letter of Llull&#39;s alphabet to an icon, much as Comenius&#39;
<i>Orbis Sensualium Pictus</i> (1658) connects letters and language to icons
of the world. In this way, the Lullian Art not only decodes reflections but
itself reflects, generating, combining and recombining a seemingly infinite
series of representations of the world. </p>';
      break;
case '320':
      echo '<p>The connection between the Lullian Art and Kabbalah is strong
&mdash; in fact, as Yates argues, &quot;there was a Cabalist element in
Lullism from the start&quot; (<a href="bibliography.html"
target="_blank">Yates<span>Yates, Frances. <U>The Art of Memory.</U> Chicago,
IL: University of Chicago Press, 1966.</span></a> 188). Kabbalah was
influential force in thirteenth-century Spain, and Llull&#39;s system of nine
Divine Dignities reflects both Aristotelian categories and the ten names of
God in the Sephiroth. A century after Llull&#39;s death, Pico della Mirandola
explicitly relates the Lullian Art to Kabbalah (see <a
href="bibliography.html" target="_blank">Yates<span>Yates, Frances. <U>The
Art of Memory.</U> Chicago, IL: University of Chicago Press, 1966.</span></a>
188).</p>';
      break;
case '321':
      echo '<p>During the Renaissance, however, Llull&#39;s reputation as a
Kabbalist rested on the <i>De auditu cabbalistico</i> (1518), a harmony of
the Lullian Art and Kabbalah then thought to be authored by Llull himself.
Scholars have since attributed it to Pietro Mainardi (1456-1529); but, during
the sixteenth and seventeenth centuries, it was included as an authentic
Lullian text in one of the more widely printed anthologies of Llull&#39;s
work, Lazarus Zetzner&#39;s <i>Opera ea quae ad adinventam ab ipso Artem
universalem</i>.</p>';
      break;
case '322':
      echo '<blockquote><p class="blockquote">The figures of his Art, on
which its concepts are set out in the letter notation, are not static but
revolving. One of the figures consists of concentric circles, marked with the
letter notations standing for the concepts, and when these wheels revolve,
combinations of the concepts are obtained. In another revolving figure,
triangles within a circle pick up related concepts. These are simple devices,
but revolutionary in their attempt to represent movement in the psyche. (<a
href="bibliography.html" target="_blank">Yates<span>Yates, Frances. <U>The
Art of Memory.</U> Chicago, IL: University of Chicago Press, 1966.</span></a>
176)</p></blockquote>';
      break;
```

```
case '323':
      echo '<p>As Frances Yates argues, the Lullian Art was also an art of
memory, using a system of places and constants to store and thereby help its
users memorize the order of the universe. In one manuscript of the <i>Ars
demonstrativa</i>, the Figure S is represented as a hand &mdash; a common
figure in the classical art of memory &mdash; with Llull&#39;s alphabet
inscribed along the fingers. </p>';
      break;
case '400':
      echo '<p>Manuscripts of Lucretius&#39;s <i>De rerum natura</i> also
contain circular diagrams that perhaps later inspired Peter Abelard (1079-
1142) to write one of the earliest surviving examples of a wheel poem (see <a
href="bibliography.html" target="_blank">Higgins<span>Higgins, Dick.
<U>Pattern Poetry: Guide to an Unknown Literature.</U> Albany: State
University of New York Press, 1987.</span></a> 28-9). After Abelard (who,
importantly, was predated by Odo of Paris, said to have written a circular
poem in the ninth century), circular labyrinths saw a revival in Europe
during the sixteenth and seventeenth centuries, sparked in part by the work
of the Italian Fortunius Licetus and Albert Molnar, a Hungarian. Both
Fortunius and Molnar compiled volumes of visual poems, including Latin
<i>carmina figurata</i> and ancient Greek <i>technopaegnia</i>, or poems
showing &quot;technical virtuosity.&quot; Now, of course, we apply the term
<i>technopaegnia</i> to any figure or &quot;pattern&quot; poem that exploits
its shape.</p>';
      break;
case '401':
      echo '<p>Every poem has, in the words of Dick Higgins, a &quot;formal
matrix&quot; in which text and its physical space interact to make meaning
(<a href="bibliography.html" target="_blank">Higgins<span>Higgins, Dick.
<U>Pattern Poetry: Guide to an Unknown Literature.</U> Albany: State
University of New York Press, 1987.</span></a> 206). For traditional verse
form, we tend to think of the poem as lines laid flat in list, each edge
neatly trimmed into a rectangle, a page, a book in miniature. Even
nontraditional works, such as Mallarm&eacute;&#39;s <i><a href="#">Un coup de
d&eacute;s Jamais N&#39;Abolira Le Hasard<span>A Throw of the Dice will Never
Abolish Chance</span></a></i> (1897), are experimental in so much as they
play with the white spaces of paper and the folded, bound form of the
book.</p><p>What, though, of the poem as a circle? Lines of text bent into
infinite loops &mdash; textual labyrinths, with no beginning and no
end?</p>';
      break;
case '402':
      echo '<blockquote><p class="blockquote">Most short poems for instance
involve a significant degree of iconicity: we see the poem as a visual whole
before we read it. Perceived optically as a complete unit the page is
qualified to such an extent that it ceases to function as an arbitrary
receptacle, or surface, for the maximum number of words it can contain
(functioning thereby as a random-sized unit in a larger construct), becoming
instead the frame, landscape, atmosphere within which the poem&#39;s own
unity is enacted and reacted upon. Page and type function as the two
ingredients in a verbal sculpture. (<a href="bibliography.html"
target="_blank">Caffrey and Nichol<span>Caffrey, Steve and bpNichol.
<U>Rational Geomancy: The Kids of the Book-machine: the collected research
reports of the Toronto Research Group, 1973-1982.</u> Vancouver: Talonbooks,
1992.</span></a> 61)</p></blockquote>';
      break;
case '403':
```

```
      echo'<p>In 1663, the Spanish Cistercian monk Juan Caramuel (1606-1682)
published his <i>Primus calamus ob oculos ponens metametricam ...</i>, known
simply as the <i>Metametrica</i>. It is an unusual and fantastic work of
linguistic aesthetics, containing pattern poems and sound poems, anagrams and
permutational poetry, mathematical language games and a plan for a universal
language based on the work of Caramuel&#39;s contemporary and fellow Jesuit,
Athanasius Kircher. Most interesting for our purposes are his circular
labyrinths.</p>';
      break;
case '404':
      echo '<p>Three circular labyrinths in Caramuel&#39;s <i>Metametrica</i>
&mdash; the &quot;Maria Stella,&quot; the &quot;Iesus Sol,&quot; and the
&quot;Coelum&quot; &mdash; use the same printed template: a disc embedded in
a set of six nested wheels, the innermost containing twelve phrases, with the
number of words around each subsequent circle increasing by twelve. Moving
from the innermost to outermost circle, the reader builds a line of verse,
such as (from the &quot;Maria Stella&quot;) <i>Parens / Te matrem / Nomina /
fontes / Lumina / quot stellas / Crux quot amara dedit</i>.</p><p>Thus the
page presents a database of isolated words and phrases organized spatially
around a set of circles; only the combinatory actions of the reader construct
a poem out of its 9,644,117,432,715,608 possible lines.</p>';
      break;
case '405':
      echo '<p>Caramuel&#39;s <i>Metametrica</i> also contains two other
kinds of circular labyrinths. The first is a set of four wheels of two discs
each, as well as phrase in the center, laid against each other in a diamond.
Between all four discs is a center circle divided into four quadrants, each
containing a phrase. By rotating the wheels against each other, the reader
can generate lines of text.</p><p>The second is an illustration for a text-
generation machine: a cylinder in four parts, each inscribed with a list of
words. Turning the cylinders against each other creates lines of verse, such
as <i>Lucifer terris referat colores</i> and <i>Lucidus coclo reparat
triumphos.</i> In nine labyrinthine diagrams, Caramuel abstractly depicts all
possible line combinations.</p>';
      break;
case '406':
      echo '<p>Although Caramuel describes his circular labyrinths in terms
of the <i>ars magna</i> of &quot;Raymundus,&quot; Ramon Llull, in fact none
of his poems physically rotate. Yet they are an important bridge between the
permutational poems of Quirinus Kuhlmann (or indeed of Caramuel himself) and
the spinning volvelles of G. P. Harsd&ouml;rffer; for unlike proteus verse,
Caramuel&#39;s distichs do not exist without the intervention of the reader.
By exploiting the spatial dimensions of the circle, infinite loops, Caramuel
imagines a mechanism for storing, retrieiving and combining textual
information.</p>';
      break;
case '408':
      echo '<p>Caramuel was not the only seventeenth-century poet to produce
circular poems. For instance, in 1666 Duke Christian Albrecht (1641-94)
produced a wheel poem that radiates lines of text out from a center point
like the spokes of a wheel (see <a href="bibliography.html"
target="_blank">Higgins<span>Higgins, Dick. <U>Pattern Poetry: Guide to an
Unknown Literature.</U> Albany: State University of New York Press,
1987.</span></a> 44). Star- and sun-shaped pattern poems, popular during the
seventeenth-century, engage with the space of the page in a similar manner
(see, for example, Hermannus de Santa Barbara&#39;s acrostic sun-shaped poem
(1687), pictured on <a href="bibliography.html"
target="_blank">Higgins<span>Higgins, Dick. <U>Pattern Poetry: Guide to an
```

```
Unknown Literature.</U> Albany: State University of New York Press,
1987.</span></a> 47). Most interesting for our purposes are the combinatory,
Kabbalistic diagrams that recur in the work of pattern poets, conceptualizing
cosmological relationships through bits of text inscribed around the edge of
nested circular diagrams. </p>';
      break;
case '409':
      echo '<p>Athanasius Kircher&#39;s &quot;Kabbalistic tree&quot; is now
the most well-known of the mystical, combinatory diagrams. Appearing in his
<i>Oedipus aegypticus</i> of 1652, the diagram takes the shape of a sunflower
of nesting rings radiating outward, aesthetically influenced by the Lullist
and Lullist-derived works of the generations before him. The center
illustrates the Tetragrammaton, the ineffable four-letter name of God; the
smallest rings illustrate His twelve-letter name; then His forty-two-letter
name; and finally His seventy-two names, iterated through the seventy-two
nations of humanity. As Stolzenberg points out, in this diagram &quot;Kircher
places the claim that all the nations of the world possess a divinely-
inspired four-letter name of God on par with the classic argument about the
wonder-working, five-letter name of Christ and its identity with the
Tetragrammaton&quot; (<a href="bibliography.html"
target="_blank">Stolzenberg<span>Stolzenberg, Daniel. &quot;Four Trees, Some
Amulets, and the Seventy-two Names of God: Kircher Reveals the
Kabbalah.&quot; <U>Athanasius Kircher: The Last Man Who Knew Everything.</U>
Ed. by Paula Findlen. New York: Routledge, 2004. 149.</span></a> 156-7; see
Stolzenberg for a more thorough description and contextualization).</p>';
      break;
case '410':
      echo '<p>Although deeply embedded in a combinatory, Kabbalistic
understanding language as a reflection of the world&#39;s order, Athanasius
Kircher&#39;s circular depiction of the seventy-two names of God is not
itself explicitly mechanical, since the wheels function as diagrams, not
volvelles. For his most famous text-generating mechanism, we must turn to a
later work, his <i>Polygraphia nova et universalis</i>, published in 1663,
the same year as Caramuel&#39;s <i>Metametrica</i>.</p>';
      break;
case '411':
      echo '<p>Although published three years before Leibniz&#39;s
<i>Dissertatio de Arte Combinatoria</i>, Kircher&#39;s <i>Polygraphia</i> is
structured using what might anachronistically be called Leibniz&#39;s model
for a <i>mathesis universalis</i>: that is, it begins with Section I,
&quot;The Reduction of All Languages to One,&quot; which encodes words in a
two-bit symbol; then moves to Section II, &quot;The Extension of One Language
to All,&quot; which decodes text to produce entire encyclopedias from a
single poem; and ends with &quot;a Techno-logica; or, a universal
Steganographic Secret operating by combinations of things&quot; (see <a
href="bibliography.html" target="_blank">Saussy<span>Saussy, Haun.
&quot;Magnetic Language: Athanasius Kircher and Communication.&quot;
<U>Athanasius Kircher: The Last Man Who Knew Everything.</U> Ed. by Paula
Findlen. New York: Routledge, 2004. 263.</span></a> 263). In other words,
operating within the programmatic epistemology of the seventeenth-century,
Kircher begins first by <i>reducing</i> knowledge to a set of universal
primitives, then mechanically permutes those constants to generate new
knowledge.</p>';
      break;
case '412':
      echo '<blockquote><p class="blockquote">Language for Kircher partakes
of two domains: a sympathetic network linking the particles of the cosmos
together, and a technology of language based on the shuffling of letters. (<a
```

```
href="bibliography.html" target="_blank">Saussy<span>Saussy, Haun.
&quot;Magnetic Language: Athanasius Kircher and Communication.&quot;
<U>Athanasius Kircher: The Last Man Who Knew Everything.</U> Ed. by Paula
Findlen. New York: Routledge, 2004. 263.</span></a> 265)</p></blockquote>';
      break;
case '413':
      echo '<p>Like Leibniz, Harsd&ouml;rffer and Kuhlmann, Athanasius
Kircher, too, was a German working within an Empire of culturally,
linguistically and religiously disparate states recently torn apart by the
Thirty Years War. From this angle, his desire to find a machine for
generating a universal language for communication &mdash; a project
commissioned by the Hapsburg Emperor Ferdinand III (see <a
href="bibliography.html" target="_blank">Saussy<span>Saussy, Haun.
&quot;Magnetic Language: Athanasius Kircher and Communication.&quot;
<U>Athanasius Kircher: The Last Man Who Knew Everything.</U> Ed. by Paula
Findlen. New York: Routledge, 2004. 263.</span></a> 268) &mdash; was not a
symptom of baroque quirkiness but of the very real challenges facing
intellectuals within a heterogeneous environment. For if reading and writing
become mechanical practices, then communication becomes the material
manipulation of language, a form of nonalphabetic, programmatic
literacy.</p>';
      break;
case '414':
      echo '<p>Athanasius Kircher&#39;s <i>Polygraphia nova</i> is one of
dozens, possibly hundreds of universal language projects from the seventeenth
century. Yet, unlike many of its predecessors and successors, Kircher&#39;s
project takes a technical turn: he invents a chest, his &quot;Glottotactic
Ark,&quot; filled with slats of wood inscribed with letters. Much the way a
cryptographic volvelle could be used to encode and decode ciphers, the wooden
slats or tablets of Kircher&#39;s ark could be used to mechanically permute
and thereby translate languages.</p><p>As Saussy points out, only
Kircher&#39;s patrons were given these combinatory text-generating machines;
it survives now in the form of diagrams and descriptions within his
<i>Polygraphia nova</i> (<a href="bibliography.html"
target="_blank">Saussy<span>Saussy, Haun. &quot;Magnetic Language: Athanasius
Kircher and Communication.&quot; <U>Athanasius Kircher: The Last Man Who Knew
Everything.</U> Ed. by Paula Findlen. New York: Routledge, 2004.
263.</span></a> 270-1).</p>';
      break;
case '415':
      echo '<p>Kircher explicitly divorces his mechanical text-generating
device from the printed word. As he writes, while attempting to find a common
root to all languages, suddenly</p><blockquote><p class="blockquote">the same
thing happened to me that might happen to a typesetter who has several pages
of type laid out and ready for putting under the press: by some inexplicable
chance the bonds dissolve and the letters rain down onto the floor, retaining
no trace of their true former meaning and no longer capable of being brought
back to their lost prototype. (Kircher <i>Turris Babel</i> 218, quoted in <a
href="bibliography.html" target="_blank">Saussy<span>Saussy, Haun.
&quot;Magnetic Language: Athanasius Kircher and Communication.&quot;
<U>Athanasius Kircher: The Last Man Who Knew Everything.</U> Ed. by Paula
Findlen. New York: Routledge, 2004. 263.</span></a>
270)</p></blockquote><p>Thus type, laid out linearly through a process of
accretion, could not easily accommodate Kircher&#39;s programmatic
epistemology of reduction and mechnical permutation.</p>';
      break;
case '416':
```

```
      echo '<blockquote><p class=&quot;blockqoute&quot;>The lesson of the
machine is that no matter how marvelous, it is still not miraculous; no
matter how many combinations a finite set of elements cn produce, its number
still falls infinitely short of infinity; and thus the triumphant display of
large numbers stands for the exhaustion of language as much as for its
fecundity. It is a melancholy Lullism. (<a href="bibliography.html"
target="_blank">Saussy<span>Saussy, Haun. &quot;Magnetic Language: Athanasius
Kircher and Communication.&quot; <U>Athanasius Kircher: The Last Man Who Knew
Everything.</U> Ed. by Paula Findlen. New York: Routledge, 2004.
263.</span></a> 276-7).</p></blockquote>';
      break;
case '417':
      echo'<blockquote><p class="blockquote">Since the tablets may be moved
about as many times as there are combinations of the letter sof the alphabet,
it is clear, then, that there can be no end to this undertaking, as is
demonstrated by the number 25852016738884976666400000 which represents the
number of combinations of the 24 letters of the alphabet. ... Surely there is
no conceivable sentence occurring in any language which cannot be represented
on the tablets; thus this narrow box and the letters enclosed therein suprass
all the libraries of the whole world. (Kircher 151, quoted in <a
href="bibliography.html" target="_blank">Saussy<span>Saussy, Haun.
&quot;Magnetic Language: Athanasius Kircher and Communication.&quot;
<U>Athanasius Kircher: The Last Man Who Knew Everything.</U> Ed. by Paula
Findlen. New York: Routledge, 2004. 263.</span></a> 272)</p></blockquote>';
      break;
case '418':
      echo '<p>Of course, Kircher and other Lullists of the seventeenth-
century were later skewered by Jonathan Swift in <i>Gulliver&#39;s
Travels</i> (1726). At the Academy of Lagado, Gulliver encounters a frame
containing a grid of wooden cubes with language inscribed on them. By turning
the handles of the frame, the pupils of the Academy form lines dictated to
scribes who form the text into books. As Swift writes, &quot;Six hours a day
the young students were employed in this labour; and the professor showed me
several volumes in large folio, already collected, of broken sentences, which
he intended to piece together, and out of those rich materials, to give the
world a complete body of all arts and sciences&quot; (<a
href="bibliography.html" target="_blank">Swift<span>Swift, Jonathan.
<U>Gulliver&#39;s Travels: Complete, Authoritative Text with Biographical and
Historical Contexts. New York: Palgrave Macmillan, 1995.</span></a>
III.5).</p>';
      break;
case '419':
      echo'<p>Kircher&#39;s <i>Polygraphia nova</i> is often (and rightly)
situated within the web of universal language project proliferating across
Europe during the same period. Yet what if, instead of linking Kircher&#39;s
project to abstract theories of language, we link it to a notion of literacy
&mdash; in other words, to a site of human interaction with a material
instance of language? From this angle, Kircher does not  &mdash; or, more
accurately, not <i>only</i> &mdash; aim to form a new grammar or lexicon, but
seeks a new method of reading and writing based on the material manipulation
of slotted bits of wood. In other words, like Leibniz&#39;s binary system,
Kircher&#39;s technical polygraphy is a read/write system, such that knowing
the abstract logic behind its function equates with knowing all of its
possible outcomes.</p>';
      break;
case '1000':
      echo '<a href="#"><img src="images/thumbnails/denckring_thumbnail.JPG"
/><span>Harsd&ouml;rffer&#39;s  <i>F&uuml;nffacher Denckring der Teutschen
```

```
Sprache</i>, or  Five-fold  Thought-ring of the German Language, from the
<i>Delici&aelig; Physico-Mathematic&aelig;, Oder, Mathemat. Und
Philosophische Erquickstunden...</i> (1651); image from the Beinecke Library
at Yale.</span></a>';
      break;
case '1001':
      echo '<a href="#"><img
src="images/thumbnails/denckring_bottomleft_thumbnail.JPG" /><span>The bottom
left and top right corners of Harsd&ouml;rffer&#39;s  <i>F&uuml;nffacher
Denckring der Teutschen Sprache</i>, or  Five-fold  Thought-ring of the
German Language, from the <i>Delici&aelig; Physico-Mathematic&aelig;, Oder,
Mathemat. Und Philosophische Erquickstunden...</i> (1651); image from the
Beinecke Library at Yale.</span></a>';
      break;
case '1002':
      echo '<a href="#"><img
src="images/thumbnails/denckring_topleft_thumbnail.JPG" /><span>The top left
and bottom right corners of Harsd&ouml;rffer&#39;s  <i>F&uuml;nffacher
Denckring der Teutschen Sprache</i>, or  Five-fold  Thought-ring of the
German Language, from the <i>Delici&aelig; Physico-Mathematic&aelig;, Oder,
Mathemat. Und Philosophische Erquickstunden...</i> (1651); image from the
Beinecke Library at Yale.</span></a>';
      break;
case '1003':
      echo '<a href="http://diglib.hab.de/drucke/ko-
108/start.htm?image=00005" target="_blank"><img
src="images/thumbnails/aristarchus_titlepage_thumbnail.jpg" /><span>The title
page of Martin Opitz&#39;s <i>Aristarchus; sive, de contemptu linguae
Teutonicae</i> (1618); image from the Herzog August Bibliothek.</span></a>';
      break;
case '1004':
      echo '<a href="http://diglib.hab.de/drucke/ko-
108/start.htm?image=00017" target="_blank"><img
src="images/thumbnails/aristarchus_2_thumbnail.jpg" /><span>From Martin
Opitz&#39;s <i>Aristarchus; sive, de contemptu linguae Teutonicae</i> (1618);
image from the Herzog August Bibliothek.</span></a>';
      break;
case '1005':
      echo '<a href="http://diglib.hab.de/drucke/ko-
108/start.htm?image=00027" target="_blank"><img
src="images/thumbnails/aristarchus_3_thumbnail.jpg" /><span>From Martin
Opitz&#39;s <i>Aristarchus; sive, de contemptu linguae Teutonicae</i> (1618);
image from the Herzog August Bibliothek.</span></a>';
      break;
case '1006':
      echo '<a href="http://diglib.hab.de/drucke/ko-
108/start.htm?image=00029" target="_blank"><img
src="images/thumbnails/aristarchus_4_thumbnail.jpg" /><span>From Martin
Opitz&#39;s <i>Aristarchus; sive, de contemptu linguae Teutonicae</i> (1618);
image from the Herzog August Bibliothek.</span></a>';
      break;
case '1007':
      echo '<a href="#"><img
src="images/thumbnails/schottel_portrait_thumbnail.jpg" /><span>A portrait of
Justus Georg Schottelius (1612-1676).</span></a>';
      break;
case '1008':
```

```
      echo '<a href="#"><img
src="images/thumbnails/brueghel_tower_thumbnail.jpg" /><span>Pieter Brueghel
the Elder&#39;s <i>Tower of Babel</i> (1563).</span></a>';
      break;
case '1009':
      echo '<a href="#"><img src="images/thumbnails/letters_A_thumbnail.JPG"
/><img src="images/thumbnails/letters_R_thumbnail.JPG" /><span>Images from a
visual alphabet in G. P. Harsd&ouml;rffer&#39;s <i>Delici&aelig; Physico-
Mathematic&aelig;, Oder, Mathemat. Und Philosophische Erquickstunden...</i>
(1651); image from the Herzog August Bibliothek.</span></a>';
      break;
case '1010':
      echo '<a href="#"><img src="images/thumbnails/letters_K_thumbnail.JPG"
/><img src="images/thumbnails/letters_U_thumbnail.JPG" /><span>Images from a
visual alphabet in G. P. Harsd&ouml;rffer&#39;s <i>Delici&aelig; Physico-
Mathematic&aelig;, Oder, Mathemat. Und Philosophische Erquickstunden...</i>
(1651); image from the Herzog August Bibliothek.</span></a>';
      break;
case '1011':
      echo '<a href="http://www.swilliams.ca/bibles/" target="_blank"><img
src="images/thumbnails/harmonies_thumbnail.png" /><span>From a Little Gidding
Harmony.</span></a>';
      break;
case '1012':
      echo '<a
href="http://www.mirroroftheworld.com.au/inspiration/printed/schatzbehalter.p
hp" target="_blank"><img src="images/thumbnails/schatzbehalter_thumbnail.jpg"
/><span>From a Stephan Fridolin&#39;s <i>Schatzbehalter</i> (1491); image
from the the State Library of Victoria</a>.</span></a>';
      break;
case '1013':
      echo '<a href="http://en.wikipedia.org/wiki/File:Jbulwer.jpg"
target="_blank"><img src="images/thumbnails/chirologia_thumbnail.jpg"
/><span>From John Bulwer&#39;s <i>Chirologia</i>; image from
Wikimedia.</span></a>';
      break;
case '1014':
      echo '<a
href="http://www.robertsabuda.com/everythingpopup/suzannekarr2.asp"
target="_blank"><img src="images/thumbnails/pointingfinger_thumbnail.jpg"
/><span>From Giambattista della Porta&#39;s <i>De occvltis literarvm
notis</i> (1593).</span></a>.)</p>';
      break;
case '1015':
      echo '<a href="#"><img
src="images/thumbnails/comenius_orbis_fire_thumbnail.JPG" /><span>An image
from Comenius&#39; <i>Orbis Sensualium Pictus</i> (1658).</span></a>';
      break;
case '1016':
      echo '<a href="#"><img
src="images/thumbnails/Europe_map_1648_thumbnail.PNG" /><span>A political map
of Europe circa 1648.</span></a>';
      break;
case '1017':
      echo '<a href="#"><img
src="images/thumbnails/leibniz_calculator_thumbnail.jpg" /><br /><p><span>A
model of Leibniz&#39;s calculator.</span></a>';
      break;
```

```
case '1019':
      echo '<a href="#"><img src="images/thumbnails/iching_thumbnail.PNG"
/><span>A representation of the Chinese I-Ching.</span></a>';
      break;
case '1020':
      echo '<a href="#"><img src="images/thumbnails/gysin_thumbnail.jpg"
/><span>A portion of Brion Gysin&#39;s permuted poem, &quot;RUB OUT THE
WORD&quot;.</span></a>';
      break;
case '1021':
      echo '<a href="#"><img
src="images/thumbnails/llull_arsdem_thumbnail.gif" /><span>Image from
Llull&#39;s <i>Ars demonstrativa</i>; image from the Biblioteca Nazionale
Marciana.</span></a>';
      break;
case '1022':
      echo '<a href="#"><img
src="images/thumbnails/llull_arsdem_s_thumbnail.gif" /><span>Image from
Llull&#39;s <i>Ars demonstrativa</i>; image from the Biblioteca Nazionale
Marciana.</span></a>';
      break;
case '1023':
      echo '<a href="#"><img
src="images/thumbnails/llull_arsdem_A_thumbnail.gif" /><span>Image from
Llull&#39;s <i>Ars demonstrativa</i>; image from the Biblioteca Nazionale
Marciana.</span></a>';
      break;
case '1024':
      echo '<a href="http://diglib.hab.de/drucke/6-3-quod-
2f/start.htm?image=00033" target="_blank"><img
src="images/thumbnails/kircher_arsmagna_T_thumbnail.jpg" /><span>The Second
Figure of Llull&#39;s <i>Ars magna</i>, as represented in Kircher&#39;s
<i>Ars magna sciendi sive combinatorica</i> (1669); image from the Herzog
August Bibliothek.</span></a>';
      break;
case '1025':
      echo '<a href="http://diglib.hab.de/drucke/6-3-quod-
2f/start.htm?image=00036" target="_blank"><img
src="images/thumbnails/kircher_llull_3_thumbnail.jpg" /><span>The Third
Figure of Llull&#39;s <i>Ars magna</i>, as represented in Kircher&#39;s
<i>Ars magna sciendi sive combinatorica</i> (1669); image from the Herzog
August Bibliothek.</span></a>';
      break;
case '1026':
      echo '<a href="http://diglib.hab.de/drucke/6-3-quod-
2f/start.htm?image=00039" target="_blank"><img
src="images/thumbnails/kircher_llull_4_thumbnail.jpg" /><span>The build-your-
own version of the Fourth Figure of Llull&#39;s <i>Ars magna</i>, as
represented in Kircher&#39;s <i>Ars magna sciendi sive combinatorica</i>
(1669); image from the Herzog August Bibliothek.</span></a>';
      break;
case '1027':
      echo '<a
href="http://www.newcastle.edu.au/service/archives/digitalscriptorium/arsmem/
index.html" target="_blank"><img src="images/thumbnails/arsmem_thumbnail.jpg"
/><span>Teaching the art of memory using the hand; from a German incunabula,
circa 1490; image from the University of Newcastle Cultural Collections
Unit.</span></a>';
```

```
      break;
case '1028':
      echo '<a href="#"><img
src="images/thumbnails/caramuel_mariastella_thumbnail.png"
/><span>Caramuel&#39;s &quot;Maria Stella&quot;, from the <i>Metametrica</i>
(1663).</span></a>';
      break;
case '1029':
      echo '<a href="#"><img
src="images/thumbnails/caramuel_cylinder_thumbnail.png"
/><span>Caramuel&#39;s text-generating cylinder, from the <i>Metametrica</i>
(1663).</span></a>';
      break;
case '1030':
      echo '<a href="#"><img
src="images/thumbnails/kircher_namesofgod_thumbnail.png"
/><span>Kircher&#39;s circular, Kabbalistic diagram showing the seventy-two
names of God.</span></a>';
      break;
case '1031':
      echo '<a href="#"><img
src="images/thumbnails/kircher_music_box_thumbnail.png" /><span>A music box
for automatically generating melodies, from Kircher&#39;s <i>Musurgia
Universalis</i> (1650). His &quot;Glottotactic Ark&quot; for translating
languages looks very similar to this.</span></a>';
      break;
case '1032':
      echo '<blockquote><p class="blockquote"><i>Possible Epigraphs for the
Soul</i> (from Charles O. Hartman&#39;s &quot;Monologues of Soul &amp;
Body&quot;, partially written using the computer program TRAVESTY)</p><p
class="blockquote">&quot;Little by little&quot; &mdash; this is Maeterlinck
&mdash;<br/>&quot;the years teach every man that truth alone<br/>is
marvelous.&quot; Fabulous old fraud.</p></blockquote>';
      break;
case '1033':
      echo '<a href="#"><img
src="images/thumbnails/kircher_icons_thumbnail.jpg" /><span>From
Kircher&#39;s <i>Ars magna sciendi sive combinatorica</i> (1669). Kircher
recommended the use of icons for the different categories within the Lullian
art.</span></a>';
      break;
 case '1034':
      echo '<blockquote><p class="blockquote"><i><b>Auf</b></i> Nacht / Dunst
/ Schlacht / Frost / Wind / See / Hitz / S&uuml;d / Ost / West / Nord / Sonn
/ Feur / und <i>Plagen</i> /<br/><br/><i><b>Folgt</b></i> Tag / Glantz /
Blutt / Schnee / Still / Land /Blitz / W&auml;rmd / Hitz / Lust / K&auml;lt /
Licht / Brand / und <i>Noth</i>:<br/><br/><i><b>Auf</i></b> Leid / Pein /
Schmach / Angst / Krig / Ach / Kreutz / Streit / Hohn / Schmertz / Qual /
T&uuml;kk / Schimpf / als <i>Spott</i> /</p></blockquote>';
      break;
 };?>
```

```
/*
___toplefthori.php___
by Whitney Anne Trettien

This file loads the horizontal links for the top left quadrant.
*/

<?php $cOption = $_GET['o'];  switch($cOption) {
case '1':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(34);"><div id="down-arrow"></div></span>';
      break;
case '2':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(3);"><div id="down-arrow"></div></span';
      break;
case '3':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(46);"><div id="down-arrow"></div></span';
      break;
case '4':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(1000);"><div id="down-
arrow"></div></span';
      break;
case '5':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(6);"><div id="down-arrow"></div></span>';
      break;
case '6':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(8);"><div id="down-arrow"></div></span';
      break;
case '7':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(8);"><div id="down-arrow"></div></span';
      break;
case '8':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(11);"><div id="down-arrow"></div></span';
      break;
case '9':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(10);"><div id="down-arrow"></div></span>';
      break;
case '10':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(34);"><div id="down-arrow"></div></span';
      break;
case '11':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(12);"><div id="down-arrow"></div></span';
      break;
case '12':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(13);"><div id="down-arrow"></div></span';
      break;
case '13':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(28);"><div id="down-arrow"></div></span>';
      break;
case '14':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(15);"><div id="down-arrow"></div></span';
      break;
case '15':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(34);"><div id="down-arrow"></div></span';
      break;
case '16':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(17);"><div id="down-arrow"></div></span';
      break;
case '17':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(21);"><div id="down-arrow"></div></span>';
      break;
case '18':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(19);"><div id="down-arrow"></div></span';
      break;
case '19':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(20);"><div id="down-arrow"></div></span';
      break;
case '20':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(21);"><div id="down-arrow"></div></span';
      break;
case '21':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(22);"><div id="down-arrow"></div></span>';
      break;
case '22':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(31);"><div id="down-arrow"></div></span';
      break;
case '23':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(24);"><div id="down-arrow"></div></span';
      break;
case '24':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(26);"><div id="down-arrow"></div></span';
      break;
case '25':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(1030);"><div id="down-
arrow"></div></span>';
      break;
case '26':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(207);"><div id="down-arrow"></div></span';
      break;
case '27':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(53);"><div id="down-arrow"></div></span';
      break;
case '28':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(59);"><div id="down-arrow"></div></span';
      break;
case '29':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(30);"><div id="down-arrow"></div></span>';
      break;
case '30':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(53);"><div id="down-arrow"></div></span';
      break;
case '31':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(37);"><div id="down-arrow"></div></span';
      break;
case '32':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(11);"><div id="down-arrow"></div></span';
      break;
case '33':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(35);"><div id="down-arrow"></div></span>';
      break;
case '34':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(36);"><div id="down-arrow"></div></span';
      break;
case '35':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(39);"><div id="down-arrow"></div></span';
      break;
case '36':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(409);"><div id="down-arrow"></div></span';
      break;
case '37':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(46);"><div id="down-arrow"></div></span>';
      break;
case '38':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(110);"><div id="down-arrow"></div></span';
      break;
case '39':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(130);"><div id="down-arrow"></div></span';
      break;
case '40':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(41);"><div id="down-arrow"></div></span';
      break;
case '41':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(42);"><div id="down-arrow"></div></span>';
```

```
      break;
case '42':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(44);"><div id="down-arrow"></div></span';
      break;
case '43':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(44);"><div id="down-arrow"></div></span';
      break;
case '44':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(175);"><div id="down-arrow"></div></span';
      break;
case '45':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(13);"><div id="down-arrow"></div></span>';
      break;
case '46':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(48);"><div id="down-arrow"></div></span';
      break;
case '4611':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(1015);"><div id="down-
arrow"></div></span';
      break;
case '4622':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(4633);"><div id="down-
arrow"></div></span';
      break;
case '4633':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(1030);"><div id="down-
arrow"></div></span';
      break;
case '4644':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(4655);"><div id="down-
arrow"></div></span';
      break;
case '4655':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(175);"><div id="down-arrow"></div></span';
      break;
case '47':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(48);"><div id="down-arrow"></div></span';
      break;
case '48':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(1010);"><div id="down-
arrow"></div></span';
      break;
case '49':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(57);"><div id="down-arrow"></div></span>';
      break;
```

```
case '53':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(1002);"><div id="down-
arrow"></div></span';
      break;
case '54':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(56);"><div id="down-arrow"></div></span>';
      break;
case '55':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(5511);"><div id="down-
arrow"></div></span';
      break;
case '5511':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(5644);"><div id="down-
arrow"></div></span';
      break;
case '5522':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(5611);"><div id="down-
arrow"></div></span';
      break;
case '56':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(58);"><div id="down-arrow"></div></span';
      break;
case '5611':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(1013);"><div id="down-
arrow"></div></span';
      break;
case '5622':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(5633);"><div id="down-
arrow"></div></span';
      break;
case '5633':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(57);"><div id="down-arrow"></div></span';
      break;
case '5644':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(5522);"><div id="down-
arrow"></div></span';
      break;
case '57':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(56);"><div id="down-arrow"></div></span';
      break;
case '58':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(28);"><div id="down-arrow"></div></span>';
      break;
case '59':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(1016);"><div id="down-
arrow"></div></span';
      break;
case '60':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(61);"><div id="down-arrow"></div></span';
      break;
case '61':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(43);"><div id="down-arrow"></div></span';
      break;
case '62':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(63);"><div id="down-arrow"></div></span>';
      break;
case '63':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(38);"><div id="down-arrow"></div></span';
      break;
case '64':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(65);"><div id="down-arrow"></div></span';
      break;
case '65':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(1004);"><div id="down-
arrow"></div></span';
      break;
case '66':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(67);"><div id="down-arrow"></div></span>';
      break;
case '67':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(1006);"><div id="down-
arrow"></div></span';
      break;
case '68':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(1);"><div id="down-arrow"></div></span';
      break;
case '100':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(401);"><div id="down-
arrow"></div></span>';
      break;
case '101':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(130);"><div id="down-arrow"></div></span';
      break;
case '102':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(104);"><div id="down-arrow"></div></span';
      break;
case '103':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(104);"><div id="down-
arrow"></div></span>';
      break;
case '104':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(105);"><div id="down-arrow"></div></span';
      break;
case '105':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(106);"><div id="down-arrow"></div></span';
      break;
case '106':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(107);"><div id="down-arrow"></div></span';
      break;
case '107':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(102);"><div id="down-arrow"></div></span';
      break;
case '108':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(121);"><div id="down-arrow"></div></span';
      break;
case '109':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(133);"><div id="down-arrow"></div></span';
      break;
case '110':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(115);"><div id="down-arrow"></div></span';
      break;
case '111':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(117);"><div id="down-arrow"></div></span';
      break;
case '112':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(41);"><div id="down-arrow"></div></span';
      break;
case '113':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(117);"><div id="down-arrow"></div></span';
      break;
case '114':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(113);"><div id="down-arrow"></div></span';
      break;
case '115':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(116);"><div id="down-arrow"></div></span';
      break;
case '116':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(126);"><div id="down-arrow"></div></span';
      break;
case '117':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(113);"><div id="down-arrow"></div></span';
      break;
case '118':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(32);"><div id="down-arrow"></div></span';
      break;
case '119':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(120);"><div id="down-arrow"></div></span';
      break;
case '11911':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(118);"><div id="down-arrow"></div></span';
      break;
case '120':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(12011);"><div id="down-
arrow"></div></span';
      break;
case '12011':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(14811);"><div id="down-
arrow"></div></span';
      break;
case '121':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(123);"><div id="down-arrow"></div></span';
      break;
case '122':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(1011);"><div id="down-
arrow"></div></span';
      break;
case '123':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(125);"><div id="down-arrow"></div></span';
      break;
case '124':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(141);"><div id="down-arrow"></div></span';
      break;
case '126':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(127);"><div id="down-arrow"></div></span';
      break;
case '127':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(129);"><div id="down-arrow"></div></span';
      break;
case '128':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(130);"><div id="down-arrow"></div></span';
      break;
case '129':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(130);"><div id="down-arrow"></div></span';
      break;
```

```
case '12911':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(4);"><div id="down-arrow"></div></span';
      break;
case '130':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(134);"><div id="down-arrow"></div></span';
      break;
case '131':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(12011);"><div id="down-
arrow"></div></span';
      break;
case '13111':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(13122);"><div id="down-
arrow"></div></span';
      break;
case '13122':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(132);"><div id="down-arrow"></div></span';
      break;
case '132':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(133);"><div id="down-arrow"></div></span';
      break;
case '133':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(109);"><div id="down-arrow"></div></span';
      break;
case '134':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(135);"><div id="down-arrow"></div></span';
      break;
case '135':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(136);"><div id="down-arrow"></div></span';
      break;
case '136':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(126);"><div id="down-arrow"></div></span';
      break;
case '137':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(141);"><div id="down-arrow"></div></span';
      break;
case '138':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(139);"><div id="down-arrow"></div></span';
      break;
case '139':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(117);"><div id="down-arrow"></div></span';
      break;
case '140':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(158);"><div id="down-arrow"></div></span';
      break;
```

```
case '141':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(143);"><div id="down-arrow"></div></span';
      break;
case '142':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(57);"><div id="down-arrow"></div></span';
      break;
case '143':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(14811);"><div id="down-
arrow"></div></span';
      break;
case '144':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(41);"><div id="down-arrow"></div></span';
      break;
case '146':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(147);"><div id="down-arrow"></div></span';
      break;
case '147':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(148);"><div id="down-arrow"></div></span';
      break;
case '14711':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(14722);"><div id="down-
arrow"></div></span';
      break;
case '14722':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(14811);"><div id="down-
arrow"></div></span';
      break;
case '148':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(1012);"><div id="down-
arrow"></div></span';
      break;
case '14811':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(37);"><div id="down-arrow"></div></span';
      break;
case '14822':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(149);"><div id="down-arrow"></div></span';
      break;
case '14833':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(100);"><div id="down-arrow"></div></span';
      break;
case '149':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(150);"><div id="down-arrow"></div></span';
      break;
case '150':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(152);"><div id="down-arrow"></div></span';
      break;
case '151':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(153);"><div id="down-arrow"></div></span';
      break;
case '152':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(115);"><div id="down-arrow"></div></span';
      break;
case '153':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(404);"><div id="down-arrow"></div></span';
      break;
case '154':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(156);"><div id="down-arrow"></div></span';
      break;
case '155':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(159);"><div id="down-arrow"></div></span';
      break;
case '156':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(128);"><div id="down-arrow"></div></span';
      break;
case '157':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(171);"><div id="down-arrow"></div></span';
      break;
case '158':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(157);"><div id="down-arrow"></div></span';
      break;
case '159':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(1008);"><div id="down-
arrow"></div></span';
      break;
case '160':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(156);"><div id="down-arrow"></div></span';
      break;
case '161':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(1020);"><div id="down-
arrow"></div></span';
      break;
case '162':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(163);"><div id="down-arrow"></div></span';
      break;
case '163':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(219);"><div id="down-arrow"></div></span';
      break;
case '164':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(250);"><div id="down-arrow"></div></span';
      break;
case '170':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(172);"><div id="down-arrow"></div></span';
      break;
case '171':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(172);"><div id="down-arrow"></div></span';
      break;
case '172':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(180);"><div id="down-arrow"></div></span';
      break;
case '173':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(174);"><div id="down-arrow"></div></span';
      break;
case '174':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(177);"><div id="down-arrow"></div></span';
      break;

case '175':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(176);"><div id="down-arrow"></div></span';
      break;
case '176':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(178);"><div id="down-arrow"></div></span';
      break;
case '177':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(178);"><div id="down-arrow"></div></span';
      break;
case '178':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(1032);"><div id="down-
arrow"></div></span';
      break;
case '179':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(180);"><div id="down-arrow"></div></span';
      break;
case '180':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(182);"><div id="down-arrow"></div></span';
      break;
case '181':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(182);"><div id="down-arrow"></div></span';
      break;
case '182':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(184);"><div id="down-arrow"></div></span';
      break;
case '183':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(184);"><div id="down-arrow"></div></span';
      break;
case '184':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(171);"><div id="down-arrow"></div></span';
      break;
case '185':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(134);"><div id="down-arrow"></div></span';
      break;
case '186':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(12011);"><div id="down-
arrow"></div></span';
      break;
case '190':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(1029);"><div id="down-
arrow"></div></span';
      break;
case '200':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(202);"><div id="down-arrow"></div></span';
      break;
case '201':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(203);"><div id="down-arrow"></div></span';
      break;
case '202':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(204);"><div id="down-arrow"></div></span';
      break;
case '203':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(232);"><div id="down-arrow"></div></span';
      break;
case '204':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(254);"><div id="down-arrow"></div></span';
      break;
case '205':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(207);"><div id="down-arrow"></div></span';
      break;
case '206':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(234);"><div id="down-arrow"></div></span';
      break;
case '207':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(208);"><div id="down-arrow"></div></span';
      break;
case '208':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(209);"><div id="down-arrow"></div></span';
      break;
case '209':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(211);"><div id="down-arrow"></div></span';
      break;
case '210':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(212);"><div id="down-arrow"></div></span';
      break;
case '211':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(213);"><div id="down-arrow"></div></span';
      break;
case '212':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(214);"><div id="down-arrow"></div></span';
      break;
case '213':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(216);"><div id="down-arrow"></div></span';
      break;
case '214':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(34);"><div id="down-arrow"></div></span';
      break;
case '215':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(219);"><div id="down-arrow"></div></span';
      break;
case '216':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(234);"><div id="down-arrow"></div></span';
      break;
case '219':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(220);"><div id="down-arrow"></div></span';
      break;
case '220':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(1017);"><div id="down-
arrow"></div></span';
      break;
case '221':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(223);"><div id="down-arrow"></div></span';
      break;
case '222':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(1017);"><div id="down-
arrow"></div></span';
      break;
case '223':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(225);"><div id="down-arrow"></div></span';
      break;
case '224':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(1019);"><div id="down-
arrow"></div></span';
      break;
```

```
case '225':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(226);"><div id="down-arrow"></div></span';
      break;
case '226':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(228);"><div id="down-arrow"></div></span';
      break;
case '227':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(229);"><div id="down-arrow"></div></span';
      break;
case '228':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(300);"><div id="down-arrow"></div></span';
      break;
case '229':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(231);"><div id="down-arrow"></div></span';
      break;
case '230':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(19);"><div id="down-arrow"></div></span';
      break;
case '231':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(39);"><div id="down-arrow"></div></span';
      break;
case '232':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(205);"><div id="down-arrow"></div></span';
      break;
case '233':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(250);"><div id="down-arrow"></div></span';
      break;
case '234':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(303);"><div id="down-arrow"></div></span';
      break;
case '235':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(212);"><div id="down-arrow"></div></span';
      break;
case '250':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(253);"><div id="down-arrow"></div></span';
      break;
case '251':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(255);"><div id="down-arrow"></div></span';
      break;
case '252':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(255);"><div id="down-arrow"></div></span';
      break;
case '253':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(262);"><div id="down-arrow"></div></span';
      break;
case '254':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(4);"><div id="down-arrow"></div></span';
      break;
case '255':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(263);"><div id="down-arrow"></div></span';
      break;
case '256':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(257);"><div id="down-arrow"></div></span';
      break;
case '257':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(258);"><div id="down-arrow"></div></span';
      break;
case '258':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(205);"><div id="down-arrow"></div></span';
      break;
case '259':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(261);"><div id="down-arrow"></div></span';
      break;
case '260':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(14833);"><div id="down-
arrow"></div></span';
      break;
case '261':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(230);"><div id="down-arrow"></div></span';
      break;
case '262':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(205);"><div id="down-arrow"></div></span';
      break;
case '263':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(265);"><div id="down-arrow"></div></span';
      break;
case '264':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(267);"><div id="down-arrow"></div></span';
      break;
case '265':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(267);"><div id="down-arrow"></div></span';
      break;
case '266':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(258);"><div id="down-arrow"></div></span';
      break;
case '267':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(143);"><div id="down-arrow"></div></span';
      break;
case '300':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(301);"><div id="down-arrow"></div></span';
      break;
case '301':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(302);"><div id="down-arrow"></div></span';
      break;
case '302':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(303);"><div id="down-arrow"></div></span';
      break;
case '303':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(304);"><div id="down-arrow"></div></span';
      break;
case '304':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(1021);"><div id="down-
arrow"></div></span';
      break;
case '305':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(302);"><div id="down-arrow"></div></span';
      break;
case '306':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(1022);"><div id="down-
arrow"></div></span';
      break;
case '307':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(308);"><div id="down-arrow"></div></span';
      break;
case '308':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(301);"><div id="down-arrow"></div></span';
      break;
case '309':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(1023);"><div id="down-
arrow"></div></span';
      break;
case '310':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(311);"><div id="down-arrow"></div></span';
      break;
case '311':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(1025);"><div id="down-
arrow"></div></span';
      break;
case '312':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(313);"><div id="down-arrow"></div></span';
```

```
      break;
case '313':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(322);"><div id="down-arrow"></div></span';
      break;
case '314':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(315);"><div id="down-arrow"></div></span';
      break;
case '315':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(317);"><div id="down-arrow"></div></span';
      break;
case '316':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(318);"><div id="down-arrow"></div></span';
      break;
case '317':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(301);"><div id="down-arrow"></div></span';
      break;
case '318':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(302);"><div id="down-arrow"></div></span';
      break;
case '319':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(1033);"><div id="down-
arrow"></div></span';
      break;
case '320':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(33);"><div id="down-arrow"></div></span';
      break;
case '321':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(35);"><div id="down-arrow"></div></span';
      break;
case '322':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(100);"><div id="down-arrow"></div></span';
      break;
case '323':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(5611);"><div id="down-
arrow"></div></span';
      break;
 case '400':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(401);"><div id="down-arrow"></div></span';
      break;
 case '401':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(1008);"><div id="down-
arrow"></div></span';
      break;
 case '402':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(134);"><div id="down-arrow"></div></span';
      break;
 case '403':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(406);"><div id="down-arrow"></div></span';
      break;
 case '404':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(1028);"><div id="down-
arrow"></div></span';
      break;
 case '405':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(406);"><div id="down-arrow"></div></span';
      break;
 case '406':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(408);"><div id="down-arrow"></div></span';
      break;
 case '408':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(409);"><div id="down-arrow"></div></span';
      break;
 case '409':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(410);"><div id="down-arrow"></div></span';
      break;
 case '410':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(403);"><div id="down-arrow"></div></span';
      break;
 case '411':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(215);"><div id="down-arrow"></div></span';
      break;
 case '412':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(413);"><div id="down-arrow"></div></span';
      break;
 case '413':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(59);"><div id="down-arrow"></div></span';
      break;
 case '414':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(1031);"><div id="down-
arrow"></div></span';
      break;
 case '415':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(417);"><div id="down-arrow"></div></span';
      break;
 case '416':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(300);"><div id="down-arrow"></div></span';
      break;
 case '417':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(418);"><div id="down-arrow"></div></span';
      break;
 case '418':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(419);"><div id="down-arrow"></div></span';
      break;
 case '419':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(219);"><div id="down-arrow"></div></span';
      break;
case '1000':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(1028);"><div id="down-
arrow"></div></span';
      break;
case '1001':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(1002);"><div id="down-
arrow"></div></span';
      break;
case '1002':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(1001);"><div id="down-
arrow"></div></span';
      break;
case '1003':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(1004);"><div id="down-
arrow"></div></span';
      break;
case '1004':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(67);"><div id="down-arrow"></div></span';
      break;
case '1005':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(1006);"><div id="down-
arrow"></div></span';
      break;
case '1006':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(68);"><div id="down-arrow"></div></span';
      break;
case '1007':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(18);"><div id="down-arrow"></div></span';
      break;
case '1008':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(49);"><div id="down-arrow"></div></span';
      break;
case '1009':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(49);"><div id="down-arrow"></div></span';
      break;
case '1010':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(49);"><div id="down-arrow"></div></span';
      break;
case '1011':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(119);"><div id="down-arrow"></div></span';
      break;
case '1012':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(56);"><div id="down-arrow"></div></span';
      break;
case '1013':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(58);"><div id="down-arrow"></div></span';
      break;
case '1014':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(5611);"><div id="down-
arrow"></div></span';
      break;
case '1015':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(4655);"><div id="down-
arrow"></div></span';
      break;
case '1016':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(61);"><div id="down-arrow"></div></span';
      break;
case '1017':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(221);"><div id="down-arrow"></div></span';
      break;
case '1019':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(226);"><div id="down-arrow"></div></span';
      break;
case '1020':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(164);"><div id="down-arrow"></div></span';
      break;
case '1021':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(307);"><div id="down-arrow"></div></span';
      break;
case '1022':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(322);"><div id="down-arrow"></div></span';
      break;
case '1023':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(310);"><div id="down-arrow"></div></span';
      break;
case '1024':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(1025);"><div id="down-
arrow"></div></span';
      break;
```

```
case '1025':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(1026);"><div id="down-
arrow"></div></span';
      break;
case '1026':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(322);"><div id="down-arrow"></div></span';
      break;
case '1027':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(129);"><div id="down-arrow"></div></span';
      break;
 case '1028':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(406);"><div id="down-arrow"></div></span';
      break;
 case '1029':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(408);"><div id="down-arrow"></div></span';
      break;
 case '1030':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(1000);"><div id="down-
arrow"></div></span';
      break;
 case '1031':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(1026);"><div id="down-
arrow"></div></span';
      break;
 case '1032':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(1020);"><div id="down-
arrow"></div></span';
      break;
case '1033':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(409);"><div id="down-arrow"></div></span';
      break;
case '1034':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomlefthori(263);"><div id="down-arrow"></div></span';
      break;
default:
      echo 'whoops';    }?>
```

```
/*
___topleftvert.php___
by Whitney Anne Trettien

This file loads the vertical links for the top left quadrant.
*/

<?php $cOption = $_GET['o'];  switch($cOption) {
case '1':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(2);"><div id="right-arrow"></div></span>';
      break;
case '2':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(48);"><div id="right-arrow"></div></span';
      break;
case '3':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(4);"><div id="right-arrow"></div></span';
      break;
case '4':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(5);"><div id="right-arrow"></div></span';
      break;
case '5':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(7);"><div id="right-arrow"></div></span>';
      break;
case '6':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(9);"><div id="right-arrow"></div></span';
      break;
case '7':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(9);"><div id="right-arrow"></div></span';
      break;
case '8':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(14);"><div id="right-arrow"></div></span';
      break;
case '9':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(177);"><div id="right-arrow"></div></span>';
      break;
case '10':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(28);"><div id="right-arrow"></div></span';
      break;
case '11':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(13);"><div id="right-arrow"></div></span';
      break;
case '12':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(53);"><div id="right-arrow"></div></span';
      break;
case '13':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(33);"><div id="right-arrow"></div></span>';
      break;
case '14':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(18);"><div id="right-arrow"></div></span';
      break;
case '15':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(16);"><div id="right-arrow"></div></span';
      break;
case '16':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(1007);"><div id="right-arrow"></div></span';
      break;
case '17':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(18);"><div id="right-arrow"></div></span>';
      break;
case '18':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(21);"><div id="right-arrow"></div></span';
      break;
case '19':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(400);"><div id="right-arrow"></div></span';
      break;
case '20':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(29);"><div id="right-arrow"></div></span';
      break;
case '21':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(46);"><div id="right-arrow"></div></span>';
      break;
case '22':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(23);"><div id="right-arrow"></div></span';
      break;
case '23':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(27);"><div id="right-arrow"></div></span';
      break;
case '24':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(25);"><div id="right-arrow"></div></span';
      break;
case '25':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(27);"><div id="right-arrow"></div></span>';
      break;
case '26':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(43);"><div id="right-arrow"></div></span';
      break;
case '27':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(29);"><div id="right-arrow"></div></span';
```

```
      break;
case '28':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(129);"><div id="right-arrow"></div></span';
      break;
case '29':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(48);"><div id="right-arrow"></div></span>';
      break;
case '30':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(31);"><div id="right-arrow"></div></span';
      break;
case '31':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(13);"><div id="right-arrow"></div></span';
      break;
case '32':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(59);"><div id="right-arrow"></div></span';
      break;
case '33':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(34);"><div id="right-arrow"></div></span>';
      break;
case '34':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(35);"><div id="right-arrow"></div></span';
      break;
case '35':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(40);"><div id="right-arrow"></div></span';
      break;
case '36':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(203);"><div id="right-arrow"></div></span';
      break;
case '37':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(39);"><div id="right-arrow"></div></span>';
      break;
case '38':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(37);"><div id="right-arrow"></div></span';
      break;
case '39':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(38);"><div id="right-arrow"></div></span';
      break;
case '40':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(4);"><div id="right-arrow"></div></span';
      break;
case '41':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(43);"><div id="right-arrow"></div></span>';
      break;
case '42':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(46);"><div id="right-arrow"></div></span';
      break;
case '43':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(1008);"><div id="right-arrow"></div></span';
      break;
case '44':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(45);"><div id="right-arrow"></div></span';
      break;
case '45':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(59);"><div id="right-arrow"></div></span>';
      break;
case '46':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(4611);"><div id="right-arrow"></div></span';
      break;
case '4611':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(4622);"><div id="right-arrow"></div></span';
      break;
case '4622':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(4655);"><div id="right-arrow"></div></span';
      break;
case '4633':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(4644);"><div id="right-arrow"></div></span';
      break;
case '4644':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(100);"><div id="right-arrow"></div></span';
      break;
case '4655':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(38);"><div id="right-arrow"></div></span';
      break;
case '47':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(49);"><div id="right-arrow"></div></span';
      break;
case '48':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(1009);"><div id="right-arrow"></div></span';
      break;
case '49':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(119);"><div id="right-arrow"></div></span>';
      break;
case '53':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(1001);"><div id="right-arrow"></div></span';
      break;
case '54':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(57);"><div id="right-arrow"></div></span>';
```

```
      break;
case '55':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(5522);"><div id="right-arrow"></div></span';
      break;
case '5511':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(5611);"><div id="right-arrow"></div></span';
      break;
case '5522':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(1014);"><div id="right-arrow"></div></span';
      break;
case '56':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(39);"><div id="right-arrow"></div></span';
      break;
case '5611':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(5622);"><div id="right-arrow"></div></span';
      break;
case '5622':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(56);"><div id="right-arrow"></div></span';
      break;
case '5633':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(1012);"><div id="right-arrow"></div></span';
      break;
case '5644':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(57);"><div id="right-arrow"></div></span';
      break;
case '57':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(3);"><div id="right-arrow"></div></span';
      break;
case '58':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(59);"><div id="right-arrow"></div></span>';
      break;
case '59':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(60);"><div id="right-arrow"></div></span';
      break;
case '60':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(43);"><div id="right-arrow"></div></span';
      break;
case '61':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(62);"><div id="right-arrow"></div></span';
      break;
case '62':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(64);"><div id="right-arrow"></div></span>';
      break;
case '63':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(64);"><div id="right-arrow"></div></span';
      break;
case '64':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(1003);"><div id="right-arrow"></div></span';
      break;
case '65':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(66);"><div id="right-arrow"></div></span';
      break;
case '66':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(1005);"><div id="right-
arrow"></div></span>';
      break;
case '67':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(68);"><div id="right-arrow"></div></span';
      break;
case '68':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(18);"><div id="right-arrow"></div></span';
      break;
case '100':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(101);"><div id="right-arrow"></div></span>';
      break;
case '101':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(1002);"><div id="right-arrow"></div></span';
      break;
case '102':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(103);"><div id="right-arrow"></div></span';
      break;
case '103':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(105);"><div id="right-arrow"></div></span>';
      break;
case '104':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(103);"><div id="right-arrow"></div></span';
      break;
case '105':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(107);"><div id="right-arrow"></div></span';
      break;
case '106':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(109);"><div id="right-arrow"></div></span';
      break;
case '107':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(13111);"><div id="right-
arrow"></div></span';
      break;
case '108':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(111);"><div id="right-arrow"></div></span';
      break;
case '109':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(42);"><div id="right-arrow"></div></span';
      break;
case '110':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(108);"><div id="right-arrow"></div></span';
      break;
case '111':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(112);"><div id="right-arrow"></div></span';
      break;
case '112':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(100);"><div id="right-arrow"></div></span';
      break;
case '113':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(140);"><div id="right-arrow"></div></span';
      break;
case '114':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(122);"><div id="right-arrow"></div></span';
      break;
case '115':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(111);"><div id="right-arrow"></div></span';
      break;
case '116':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(118);"><div id="right-arrow"></div></span';
      break;
case '117':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(134);"><div id="right-arrow"></div></span';
      break;
case '118':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(102);"><div id="right-arrow"></div></span';
      break;
case '119':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(11911);"><div id="right-
arrow"></div></span';
      break;
case '11911':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(120);"><div id="right-arrow"></div></span';
      break;
case '120':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(126);"><div id="right-arrow"></div></span';
      break;
case '12011':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(4);"><div id="right-arrow"></div></span';
      break;
case '121':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(59);"><div id="right-arrow"></div></span';
      break;
case '122':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(102);"><div id="right-arrow"></div></span';
      break;
case '123':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(124);"><div id="right-arrow"></div></span';
      break;
case '124':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(403);"><div id="right-arrow"></div></span';
      break;
case '126':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(128);"><div id="right-arrow"></div></span';
      break;
case '127':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(128);"><div id="right-arrow"></div></span';
      break;
case '128':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(129);"><div id="right-arrow"></div></span';
      break;
case '129':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(100);"><div id="right-arrow"></div></span';
      break;
case '12911':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(150);"><div id="right-arrow"></div></span';
      break;
case '130':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(131);"><div id="right-arrow"></div></span';
      break;
case '131':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(13111);"><div id="right-
arrow"></div></span';
      break;
case '13111':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(132);"><div id="right-arrow"></div></span';
      break;
case '13122':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(148);"><div id="right-arrow"></div></span';
      break;
case '132':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(14711);"><div id="right-
arrow"></div></span';
      break;
case '133':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(14822);"><div id="right-
arrow"></div></span';
      break;
case '134':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(138);"><div id="right-arrow"></div></span';
      break;
case '135':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(137);"><div id="right-arrow"></div></span';
      break;
case '136':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(141);"><div id="right-arrow"></div></span';
      break;
case '137':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(130);"><div id="right-arrow"></div></span';
      break;
case '138':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(140);"><div id="right-arrow"></div></span';
      break;
case '139':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(43);"><div id="right-arrow"></div></span';
      break;
case '140':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(111);"><div id="right-arrow"></div></span';
      break;
case '141':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(142);"><div id="right-arrow"></div></span';
      break;
case '142':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(143);"><div id="right-arrow"></div></span';
      break;
case '143':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(144);"><div id="right-arrow"></div></span';
      break;
case '144':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(146);"><div id="right-arrow"></div></span';
      break;
case '146':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(13122);"><div id="right-
arrow"></div></span';
      break;
```

```
case '147':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(14711);"><div id="right-
arrow"></div></span';
      break;
case '14711':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(149);"><div id="right-arrow"></div></span';
      break;
case '14722':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(148);"><div id="right-arrow"></div></span';
      break;
case '148':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(49);"><div id="right-arrow"></div></span';
      break;
case '14811':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(13);"><div id="right-arrow"></div></span';
      break;
case '14822':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(113);"><div id="right-arrow"></div></span';
      break;
case '14833':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(1000);"><div id="right-arrow"></div></span';
      break;
case '149':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(154);"><div id="right-arrow"></div></span';
      break;
case '150':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(151);"><div id="right-arrow"></div></span';
      break;
case '151':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(152);"><div id="right-arrow"></div></span';
      break;
case '152':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(153);"><div id="right-arrow"></div></span';
      break;
case '153':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(4);"><div id="right-arrow"></div></span';
      break;
case '154':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(155);"><div id="right-arrow"></div></span';
      break;
case '155':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(158);"><div id="right-arrow"></div></span';
      break;
case '156':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(157);"><div id="right-arrow"></div></span';
      break;
case '157':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(129);"><div id="right-arrow"></div></span';
      break;
case '158':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(140);"><div id="right-arrow"></div></span';
      break;
case '159':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(156);"><div id="right-arrow"></div></span';
      break;
case '160':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(1034);"><div id="right-arrow"></div></span';
      break;
case '161':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(162);"><div id="right-arrow"></div></span';
      break;
case '162':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(1020);"><div id="right-arrow"></div></span';
      break;
case '163':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(164);"><div id="right-arrow"></div></span';
      break;
case '164':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(130);"><div id="right-arrow"></div></span';
      break;
case '170':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(171);"><div id="right-arrow"></div></span';
      break;
case '171':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(53);"><div id="right-arrow"></div></span';
      break;
case '172':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(173);"><div id="right-arrow"></div></span';
      break;
case '173':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(177);"><div id="right-arrow"></div></span';
      break;
case '174':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(176);"><div id="right-arrow"></div></span';
      break;
case '175':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(178);"><div id="right-arrow"></div></span';
```

```
      break;
case '176':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(14811);"><div id="right-
arrow"></div></span';
      break;
case '177':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(176);"><div id="right-arrow"></div></span';
      break;
case '178':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(179);"><div id="right-arrow"></div></span';
      break;
case '179':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(403);"><div id="right-arrow"></div></span';
      break;
case '180':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(181);"><div id="right-arrow"></div></span';
      break;
case '181':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(184);"><div id="right-arrow"></div></span';
      break;
case '182':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(183);"><div id="right-arrow"></div></span';
      break;
case '183':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(185);"><div id="right-arrow"></div></span';
      break;
case '184':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(185);"><div id="right-arrow"></div></span';
      break;
case '185':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(403);"><div id="right-arrow"></div></span';
      break;
case '186':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(171);"><div id="right-arrow"></div></span';
      break;
case '190':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(1000);"><div id="right-arrow"></div></span';
      break;
case '200':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(201);"><div id="right-arrow"></div></span';
      break;
case '201':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(202);"><div id="right-arrow"></div></span';
      break;
```

```
case '202':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(203);"><div id="right-arrow"></div></span';
      break;
case '203':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(204);"><div id="right-arrow"></div></span';
      break;
case '204':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(205);"><div id="right-arrow"></div></span';
      break;
case '205':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(206);"><div id="right-arrow"></div></span';
      break;
case '206':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(207);"><div id="right-arrow"></div></span';
      break;
case '207':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(4);"><div id="right-arrow"></div></span';
      break;
case '208':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(210);"><div id="right-arrow"></div></span';
      break;
case '209':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(173);"><div id="right-arrow"></div></span';
      break;
case '210':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(211);"><div id="right-arrow"></div></span';
      break;
case '211':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(212);"><div id="right-arrow"></div></span';
      break;
case '212':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(213);"><div id="right-arrow"></div></span';
      break;
case '213':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(215);"><div id="right-arrow"></div></span';
      break;
case '214':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(303);"><div id="right-arrow"></div></span';
      break;
case '215':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(216);"><div id="right-arrow"></div></span';
      break;
case '216':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(222);"><div id="right-arrow"></div></span';
      break;
case '219':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(214);"><div id="right-arrow"></div></span';
      break;
case '220':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(221);"><div id="right-arrow"></div></span';
      break;
case '221':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(222);"><div id="right-arrow"></div></span';
      break;
case '222':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(223);"><div id="right-arrow"></div></span';
      break;
case '223':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(224);"><div id="right-arrow"></div></span';
      break;
case '224':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(225);"><div id="right-arrow"></div></span';
      break;
case '225':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(1019);"><div id="right-arrow"></div></span';
      break;
case '226':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(227);"><div id="right-arrow"></div></span';
      break;
case '227':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(228);"><div id="right-arrow"></div></span';
      break;
case '228':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(229);"><div id="right-arrow"></div></span';
      break;
case '229':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(230);"><div id="right-arrow"></div></span';
      break;
case '230':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(231);"><div id="right-arrow"></div></span';
      break;
case '231':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(250);"><div id="right-arrow"></div></span';
      break;
case '232':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(233);"><div id="right-arrow"></div></span';
```

```
      break;
case '233':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(205);"><div id="right-arrow"></div></span';
      break;
case '234':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(219);"><div id="right-arrow"></div></span';
      break;
case '235':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(208);"><div id="right-arrow"></div></span';
      break;
case '250':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(251);"><div id="right-arrow"></div></span';
      break;
case '251':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(252);"><div id="right-arrow"></div></span';
      break;
case '252':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(253);"><div id="right-arrow"></div></span';
      break;
case '253':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(254);"><div id="right-arrow"></div></span';
      break;
case '254':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(255);"><div id="right-arrow"></div></span';
      break;
case '255':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(256);"><div id="right-arrow"></div></span';
      break;
case '256':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(263);"><div id="right-arrow"></div></span';
      break;
case '257':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(143);"><div id="right-arrow"></div></span';
      break;
case '258':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(259);"><div id="right-arrow"></div></span';
      break;
case '259':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(260);"><div id="right-arrow"></div></span';
      break;
case '260':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(161);"><div id="right-arrow"></div></span';
      break;
case '261':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(129);"><div id="right-arrow"></div></span';
      break;
case '262':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(319);"><div id="right-arrow"></div></span';
      break;
case '263':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(264);"><div id="right-arrow"></div></span';
      break;
case '264':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(265);"><div id="right-arrow"></div></span';
      break;
case '265':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(266);"><div id="right-arrow"></div></span';
      break;
case '266':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(2);"><div id="right-arrow"></div></span';
      break;
case '267':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(2);"><div id="right-arrow"></div></span';
      break;
case '300':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(303);"><div id="right-arrow"></div></span';
      break;
case '301':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(59);"><div id="right-arrow"></div></span';
      break;
case '302':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(14);"><div id="right-arrow"></div></span';
      break;
case '303':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(320);"><div id="right-arrow"></div></span';
      break;
case '304':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(305);"><div id="right-arrow"></div></span';
      break;
case '305':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(306);"><div id="right-arrow"></div></span';
      break;
case '306':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(307);"><div id="right-arrow"></div></span';
      break;
case '307':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(171);"><div id="right-arrow"></div></span';
```

```
      break;
case '308':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(309);"><div id="right-arrow"></div></span';
      break;
case '309':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(310);"><div id="right-arrow"></div></span';
      break;
case '310':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(1024);"><div id="right-arrow"></div></span';
      break;
case '311':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(312);"><div id="right-arrow"></div></span';
      break;
case '312':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(1026);"><div id="right-arrow"></div></span';
      break;
case '313':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(315);"><div id="right-arrow"></div></span';
      break;
case '314':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(172);"><div id="right-arrow"></div></span';
      break;
case '315':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(316);"><div id="right-arrow"></div></span';
      break;
case '316':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(317);"><div id="right-arrow"></div></span';
      break;
case '317':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(319);"><div id="right-arrow"></div></span';
      break;
case '318':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(319);"><div id="right-arrow"></div></span';
      break;
case '319':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(4611);"><div id="right-arrow"></div></span';
      break;
case '320':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(321);"><div id="right-arrow"></div></span';
      break;
case '321':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(323);"><div id="right-arrow"></div></span';
      break;
case '322':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(3);"><div id="right-arrow"></div></span';
      break;
case '323':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(1027);"><div id="right-arrow"></div></span';
      break;

case '400':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(402);"><div id="right-arrow"></div></span';
      break;
case '401':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(402);"><div id="right-arrow"></div></span';
      break;
case '402':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(403);"><div id="right-arrow"></div></span';
      break;
case '403':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(404);"><div id="right-arrow"></div></span';
      break;
case '404':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(405);"><div id="right-arrow"></div></span';
      break;
case '405':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(1029);"><div id="right-arrow"></div></span';
      break;
case '406':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(315);"><div id="right-arrow"></div></span';
      break;
case '408':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(4);"><div id="right-arrow"></div></span';
      break;
case '409':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(1030);"><div id="right-arrow"></div></span';
      break;
case '410':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(411);"><div id="right-arrow"></div></span';
      break;
case '411':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(414);"><div id="right-arrow"></div></span';
      break;
case '412':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(415);"><div id="right-arrow"></div></span';
      break;
case '413':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(212);"><div id="right-arrow"></div></span';
      break;
case '414':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(412);"><div id="right-arrow"></div></span';
      break;
case '415':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(419);"><div id="right-arrow"></div></span';
      break;
case '416':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(417);"><div id="right-arrow"></div></span';
      break;
case '417':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(419);"><div id="right-arrow"></div></span';
      break;
case '418':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(262);"><div id="right-arrow"></div></span';
      break;
case '419':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(173);"><div id="right-arrow"></div></span';
      break;
case '1000':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(8);"><div id="right-arrow"></div></span';
      break;
case '1001':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(54);"><div id="right-arrow"></div></span';
      break;
case '1002':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(55);"><div id="right-arrow"></div></span';
      break;
case '1003':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(66);"><div id="right-arrow"></div></span';
      break;
case '1004':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(1005);"><div id="right-arrow"></div></span';
      break;
case '1005':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(67);"><div id="right-arrow"></div></span';
      break;
case '1006':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(15);"><div id="right-arrow"></div></span';
      break;
case '1007':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(17);"><div id="right-arrow"></div></span';
```

```
      break;
case '1008':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(44);"><div id="right-arrow"></div></span';
      break;
case '1009':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(4611);"><div id="right-arrow"></div></span';
      break;
case '1010':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(4611);"><div id="right-arrow"></div></span';
      break;
case '1011':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(114);"><div id="right-arrow"></div></span';
      break;
case '1012':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(57);"><div id="right-arrow"></div></span';
      break;
case '1013':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(5622);"><div id="right-arrow"></div></span';
      break;
case '1014':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(5644);"><div id="right-arrow"></div></span';
      break;
case '1015':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(4622);"><div id="right-arrow"></div></span';
      break;
case '1016':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(60);"><div id="right-arrow"></div></span';
      break;
case '1017':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(223);"><div id="right-arrow"></div></span';
      break;
case '1019':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(225);"><div id="right-arrow"></div></span';
      break;
case '1020':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(163);"><div id="right-arrow"></div></span';
      break;
case '1021':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(306);"><div id="right-arrow"></div></span';
      break;
case '1022':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(323);"><div id="right-arrow"></div></span';
      break;
case '1023':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(1024);"><div id="right-arrow"></div></span';
      break;
case '1024':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(311);"><div id="right-arrow"></div></span';
      break;
case '1025':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(312);"><div id="right-arrow"></div></span';
      break;
case '1026':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(314);"><div id="right-arrow"></div></span';
      break;
case '1027':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(5633);"><div id="right-arrow"></div></span';
      break;
case '1028':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(1029);"><div id="right-arrow"></div></span';
      break;
case '1029':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(406);"><div id="right-arrow"></div></span';
      break;
case '1030':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(1028);"><div id="right-arrow"></div></span';
      break;
case '1031':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(1029);"><div id="right-arrow"></div></span';
      break;
case '1032':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(179);"><div id="right-arrow"></div></span';
      break;
case '1033':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(1031);"><div id="right-arrow"></div></span';
      break;
case '1034':
      echo '<span style="cursor: pointer"
onclick="loadContentToprightvert(1029);"><div id="right-arrow"></div></span';
      break;
default:
      echo 'whoops';    }?>
```

```
/*
___toprighthori.php___
by Whitney Anne Trettien

This file loads the horizontal links for the top right quadrant.
*/

<?php $cOption = $_GET['o'];  switch($cOption) {
case '1':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(34);"><div id="down-
arrow"></div></span>';
      break;
case '2':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(3);"><div id="down-arrow"></div></span';
      break;
case '3':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(46);"><div id="down-arrow"></div></span';
      break;
case '4':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(1000);"><div id="down-
arrow"></div></span';
      break;
case '5':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(6);"><div id="down-arrow"></div></span>';
      break;
case '6':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(8);"><div id="down-arrow"></div></span';
      break;
case '7':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(8);"><div id="down-arrow"></div></span';
      break;
case '8':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(11);"><div id="down-arrow"></div></span';
      break;
case '9':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(10);"><div id="down-
arrow"></div></span>';
      break;
case '10':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(34);"><div id="down-arrow"></div></span';
      break;
case '11':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(12);"><div id="down-arrow"></div></span';
      break;
case '12':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(13);"><div id="down-arrow"></div></span';
```

```
      break;
case '13':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(28);"><div id="down-
arrow"></div></span>';
      break;
case '14':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(15);"><div id="down-arrow"></div></span';
      break;
case '15':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(34);"><div id="down-arrow"></div></span';
      break;
case '16':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(17);"><div id="down-arrow"></div></span';
      break;
case '17':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(21);"><div id="down-
arrow"></div></span>';
      break;
case '18':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(19);"><div id="down-arrow"></div></span';
      break;
case '19':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(20);"><div id="down-arrow"></div></span';
      break;
case '20':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(21);"><div id="down-arrow"></div></span';
      break;
case '21':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(22);"><div id="down-
arrow"></div></span>';
      break;
case '22':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(31);"><div id="down-arrow"></div></span';
      break;
case '23':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(24);"><div id="down-arrow"></div></span';
      break;
case '24':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(26);"><div id="down-arrow"></div></span';
      break;
case '25':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(1030);"><div id="down-
arrow"></div></span>';
      break;
case '26':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(207);"><div id="down-
arrow"></div></span';
      break;
case '27':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(53);"><div id="down-arrow"></div></span';
      break;
case '28':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(59);"><div id="down-arrow"></div></span';
      break;
case '29':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(30);"><div id="down-
arrow"></div></span>';
      break;
case '30':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(53);"><div id="down-arrow"></div></span';
      break;
case '31':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(37);"><div id="down-arrow"></div></span';
      break;
case '32':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(11);"><div id="down-arrow"></div></span';
      break;
case '33':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(35);"><div id="down-
arrow"></div></span>';
      break;
case '34':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(36);"><div id="down-arrow"></div></span';
      break;
case '35':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(39);"><div id="down-arrow"></div></span';
      break;
case '36':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(409);"><div id="down-
arrow"></div></span';
      break;
case '37':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(46);"><div id="down-
arrow"></div></span>';
      break;
case '38':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(110);"><div id="down-
arrow"></div></span';
      break;
case '39':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(130);"><div id="down-
arrow"></div></span';
      break;
case '40':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(41);"><div id="down-arrow"></div></span';
      break;
case '41':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(42);"><div id="down-
arrow"></div></span>';
      break;
case '42':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(44);"><div id="down-arrow"></div></span';
      break;
case '43':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(44);"><div id="down-arrow"></div></span';
      break;
case '44':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(175);"><div id="down-
arrow"></div></span';
      break;
case '45':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(13);"><div id="down-
arrow"></div></span>';
      break;
case '46':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(48);"><div id="down-arrow"></div></span';
      break;
case '4611':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(1015);"><div id="down-
arrow"></div></span';
      break;
case '4622':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(4633);"><div id="down-
arrow"></div></span';
      break;
case '4633':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(1030);"><div id="down-
arrow"></div></span';
      break;
case '4644':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(4655);"><div id="down-
arrow"></div></span';
      break;
case '4655':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(175);"><div id="down-
arrow"></div></span';
      break;
case '47':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(48);"><div id="down-arrow"></div></span';
      break;
case '48':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(1010);"><div id="down-
arrow"></div></span';
      break;
case '49':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(57);"><div id="down-
arrow"></div></span>';
      break;
case '53':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(1002);"><div id="down-
arrow"></div></span';
      break;
case '54':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(56);"><div id="down-
arrow"></div></span>';
      break;
case '55':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(5511);"><div id="down-
arrow"></div></span';
      break;
case '5511':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(5644);"><div id="down-
arrow"></div></span';
      break;
case '5522':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(5611);"><div id="down-
arrow"></div></span';
      break;
case '56':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(58);"><div id="down-arrow"></div></span';
      break;
case '5611':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(1013);"><div id="down-
arrow"></div></span';
      break;
case '5622':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(5633);"><div id="down-
arrow"></div></span';
      break;
case '5633':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(57);"><div id="down-arrow"></div></span';
      break;
case '5644':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(5522);"><div id="down-
arrow"></div></span';
      break;
case '57':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(56);"><div id="down-arrow"></div></span';
      break;
case '58':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(28);"><div id="down-
arrow"></div></span>';
      break;
case '59':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(1016);"><div id="down-
arrow"></div></span';
      break;
case '60':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(61);"><div id="down-arrow"></div></span';
      break;
case '61':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(43);"><div id="down-arrow"></div></span';
      break;
case '62':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(63);"><div id="down-
arrow"></div></span>';
      break;
case '63':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(38);"><div id="down-arrow"></div></span';
      break;
case '64':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(65);"><div id="down-arrow"></div></span';
      break;
case '65':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(1004);"><div id="down-
arrow"></div></span';
      break;
case '66':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(67);"><div id="down-
arrow"></div></span>';
      break;
case '67':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(1006);"><div id="down-
arrow"></div></span';
      break;
```

```
case '68':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(1);"><div id="down-arrow"></div></span';
      break;
case '100':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(401);"><div id="down-
arrow"></div></span>';
      break;
case '101':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(130);"><div id="down-
arrow"></div></span';
      break;
case '102':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(104);"><div id="down-
arrow"></div></span';
      break;
case '103':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(104);"><div id="down-
arrow"></div></span>';
      break;
case '104':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(105);"><div id="down-
arrow"></div></span';
      break;
case '105':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(106);"><div id="down-
arrow"></div></span';
      break;
case '106':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(107);"><div id="down-
arrow"></div></span';
      break;
case '107':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(102);"><div id="down-
arrow"></div></span';
      break;
case '108':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(121);"><div id="down-
arrow"></div></span';
      break;
case '109':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(133);"><div id="down-
arrow"></div></span';
      break;
case '110':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(115);"><div id="down-
arrow"></div></span';
```

```
      break;
case '111':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(117);"><div id="down-
arrow"></div></span';
      break;
case '112':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(41);"><div id="down-arrow"></div></span';
      break;
case '113':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(117);"><div id="down-
arrow"></div></span';
      break;
case '114':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(113);"><div id="down-
arrow"></div></span';
      break;
case '115':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(116);"><div id="down-
arrow"></div></span';
      break;
case '116':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(126);"><div id="down-
arrow"></div></span';
      break;
case '117':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(113);"><div id="down-
arrow"></div></span';
      break;
case '118':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(32);"><div id="down-arrow"></div></span';
      break;
case '119':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(120);"><div id="down-
arrow"></div></span';
      break;
case '11911':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(118);"><div id="down-
arrow"></div></span';
      break;
case '120':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(12011);"><div id="down-
arrow"></div></span';
      break;
case '12011':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(14811);"><div id="down-
arrow"></div></span';
```

```
      break;
case '121':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(123);"><div id="down-
arrow"></div></span';
      break;
case '122':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(1011);"><div id="down-
arrow"></div></span';
      break;
case '123':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(125);"><div id="down-
arrow"></div></span';
      break;
case '124':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(141);"><div id="down-
arrow"></div></span';
      break;
case '126':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(127);"><div id="down-
arrow"></div></span';
      break;
case '127':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(129);"><div id="down-
arrow"></div></span';
      break;
case '128':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(130);"><div id="down-
arrow"></div></span';
      break;
case '129':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(130);"><div id="down-
arrow"></div></span';
      break;
case '12911':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(4);"><div id="down-arrow"></div></span';
      break;
case '130':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(134);"><div id="down-
arrow"></div></span';
      break;
case '131':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(12011);"><div id="down-
arrow"></div></span';
      break;
case '13111':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(13122);"><div id="down-
arrow"></div></span';
      break;
case '13122':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(132);"><div id="down-
arrow"></div></span';
      break;
case '132':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(133);"><div id="down-
arrow"></div></span';
      break;
case '133':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(109);"><div id="down-
arrow"></div></span';
      break;
case '134':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(135);"><div id="down-
arrow"></div></span';
      break;
case '135':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(136);"><div id="down-
arrow"></div></span';
      break;
case '136':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(126);"><div id="down-
arrow"></div></span';
      break;
case '137':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(141);"><div id="down-
arrow"></div></span';
      break;
case '138':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(139);"><div id="down-
arrow"></div></span';
      break;
case '139':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(117);"><div id="down-
arrow"></div></span';
      break;
case '140':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(158);"><div id="down-
arrow"></div></span';
      break;
case '141':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(143);"><div id="down-
arrow"></div></span';
```

```
      break;
case '142':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(57);"><div id="down-arrow"></div></span';
      break;
case '143':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(14811);"><div id="down-
arrow"></div></span';
      break;
case '144':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(41);"><div id="down-arrow"></div></span';
      break;
case '146':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(147);"><div id="down-
arrow"></div></span';
      break;
case '147':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(148);"><div id="down-
arrow"></div></span';
      break;
case '14711':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(14722);"><div id="down-
arrow"></div></span';
      break;
case '14722':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(14811);"><div id="down-
arrow"></div></span';
      break;
case '148':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(1012);"><div id="down-
arrow"></div></span';
      break;
case '14811':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(37);"><div id="down-arrow"></div></span';
      break;
case '14822':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(149);"><div id="down-
arrow"></div></span';
      break;
case '14833':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(100);"><div id="down-
arrow"></div></span';
      break;
case '149':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(150);"><div id="down-
arrow"></div></span';
      break;
```

```
case '150':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(152);"><div id="down-
arrow"></div></span';
      break;
case '151':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(153);"><div id="down-
arrow"></div></span';
      break;
case '152':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(115);"><div id="down-
arrow"></div></span';
      break;
case '153':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(404);"><div id="down-
arrow"></div></span';
      break;
case '154':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(156);"><div id="down-
arrow"></div></span';
      break;
case '155':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(159);"><div id="down-
arrow"></div></span';
      break;
case '156':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(128);"><div id="down-
arrow"></div></span';
      break;
case '157':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(171);"><div id="down-
arrow"></div></span';
      break;
case '158':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(157);"><div id="down-
arrow"></div></span';
      break;
case '159':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(1008);"><div id="down-
arrow"></div></span';
      break;
case '160':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(156);"><div id="down-
arrow"></div></span';
      break;
case '161':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(1020);"><div id="down-
arrow"></div></span';
      break;
case '162':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(163);"><div id="down-
arrow"></div></span';
      break;
case '163':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(219);"><div id="down-
arrow"></div></span';
      break;
case '164':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(250);"><div id="down-
arrow"></div></span';
      break;
case '170':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(172);"><div id="down-
arrow"></div></span';
      break;
case '171':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(172);"><div id="down-
arrow"></div></span';
      break;
case '172':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(180);"><div id="down-
arrow"></div></span';
      break;
case '173':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(174);"><div id="down-
arrow"></div></span';
      break;
case '174':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(177);"><div id="down-
arrow"></div></span';
      break;

case '175':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(176);"><div id="down-
arrow"></div></span';
      break;
case '176':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(178);"><div id="down-
arrow"></div></span';
      break;
case '177':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(178);"><div id="down-
arrow"></div></span';
      break;
case '178':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(1032);"><div id="down-
arrow"></div></span';
      break;
case '179':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(180);"><div id="down-
arrow"></div></span';
      break;
case '180':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(182);"><div id="down-
arrow"></div></span';
      break;
case '181':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(182);"><div id="down-
arrow"></div></span';
      break;
case '182':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(184);"><div id="down-
arrow"></div></span';
      break;
case '183':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(184);"><div id="down-
arrow"></div></span';
      break;
case '184':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(171);"><div id="down-
arrow"></div></span';
      break;
case '185':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(134);"><div id="down-
arrow"></div></span';
      break;
case '186':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(12011);"><div id="down-
arrow"></div></span';
      break;
case '190':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(1029);"><div id="down-
arrow"></div></span';
      break;
case '200':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(202);"><div id="down-
arrow"></div></span';
```

```
      break;
case '201':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(203);"><div id="down-
arrow"></div></span';
      break;
case '202':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(204);"><div id="down-
arrow"></div></span';
      break;
case '203':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(232);"><div id="down-
arrow"></div></span';
      break;
case '204':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(254);"><div id="down-
arrow"></div></span';
      break;
case '205':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(207);"><div id="down-
arrow"></div></span';
      break;
case '206':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(234);"><div id="down-
arrow"></div></span';
      break;
case '207':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(208);"><div id="down-
arrow"></div></span';
      break;
case '208':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(209);"><div id="down-
arrow"></div></span';
      break;
case '209':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(211);"><div id="down-
arrow"></div></span';
      break;
case '210':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(212);"><div id="down-
arrow"></div></span';
      break;
case '211':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(213);"><div id="down-
arrow"></div></span';
      break;
case '212':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(214);"><div id="down-
arrow"></div></span';
      break;
case '213':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(216);"><div id="down-
arrow"></div></span';
      break;
case '214':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(34);"><div id="down-arrow"></div></span';
      break;
case '215':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(219);"><div id="down-
arrow"></div></span';
      break;
case '216':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(234);"><div id="down-
arrow"></div></span';
      break;
case '219':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(220);"><div id="down-
arrow"></div></span';
      break;
case '220':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(1017);"><div id="down-
arrow"></div></span';
      break;
case '221':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(223);"><div id="down-
arrow"></div></span';
      break;
case '222':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(1017);"><div id="down-
arrow"></div></span';
      break;
case '223':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(225);"><div id="down-
arrow"></div></span';
      break;
case '224':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(1019);"><div id="down-
arrow"></div></span';
      break;
case '225':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(226);"><div id="down-
arrow"></div></span';
      break;
```

```
case '226':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(228);"><div id="down-
arrow"></div></span';
      break;
case '227':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(229);"><div id="down-
arrow"></div></span';
      break;
case '228':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(300);"><div id="down-
arrow"></div></span';
      break;
case '229':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(231);"><div id="down-
arrow"></div></span';
      break;
case '230':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(19);"><div id="down-arrow"></div></span';
      break;
case '231':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(39);"><div id="down-arrow"></div></span';
      break;
case '232':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(205);"><div id="down-
arrow"></div></span';
      break;
case '233':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(250);"><div id="down-
arrow"></div></span';
      break;
case '234':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(303);"><div id="down-
arrow"></div></span';
      break;
case '235':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(212);"><div id="down-
arrow"></div></span';
      break;
case '250':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(253);"><div id="down-
arrow"></div></span';
      break;
case '251':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(255);"><div id="down-
arrow"></div></span';
      break;
```

```
case '252':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(255);"><div id="down-
arrow"></div></span';
      break;
case '253':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(262);"><div id="down-
arrow"></div></span';
      break;
case '254':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(4);"><div id="down-arrow"></div></span';
      break;
case '255':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(263);"><div id="down-
arrow"></div></span';
      break;
case '256':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(257);"><div id="down-
arrow"></div></span';
      break;
case '257':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(258);"><div id="down-
arrow"></div></span';
      break;
case '258':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(205);"><div id="down-
arrow"></div></span';
      break;
case '259':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(261);"><div id="down-
arrow"></div></span';
      break;
case '260':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(14833);"><div id="down-
arrow"></div></span';
      break;
case '261':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(230);"><div id="down-
arrow"></div></span';
      break;
case '262':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(205);"><div id="down-
arrow"></div></span';
      break;
case '263':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(265);"><div id="down-
arrow"></div></span';
```

```
      break;
case '264':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(267);"><div id="down-
arrow"></div></span';
      break;
case '265':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(267);"><div id="down-
arrow"></div></span';
      break;
case '266':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(258);"><div id="down-
arrow"></div></span';
      break;
case '267':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(143);"><div id="down-
arrow"></div></span';
      break;
case '300':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(301);"><div id="down-
arrow"></div></span';
      break;
case '301':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(302);"><div id="down-
arrow"></div></span';
      break;
case '302':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(303);"><div id="down-
arrow"></div></span';
      break;
case '303':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(304);"><div id="down-
arrow"></div></span';
      break;
case '304':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(1021);"><div id="down-
arrow"></div></span';
      break;
case '305':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(302);"><div id="down-
arrow"></div></span';
      break;
case '306':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(1022);"><div id="down-
arrow"></div></span';
      break;
case '307':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(308);"><div id="down-
arrow"></div></span';
      break;
case '308':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(301);"><div id="down-
arrow"></div></span';
      break;
case '309':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(1023);"><div id="down-
arrow"></div></span';
      break;
case '310':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(311);"><div id="down-
arrow"></div></span';
      break;
case '311':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(1025);"><div id="down-
arrow"></div></span';
      break;
case '312':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(313);"><div id="down-
arrow"></div></span';
      break;
case '313':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(322);"><div id="down-
arrow"></div></span';
      break;
case '314':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(315);"><div id="down-
arrow"></div></span';
      break;
case '315':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(317);"><div id="down-
arrow"></div></span';
      break;
case '316':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(318);"><div id="down-
arrow"></div></span';
      break;
case '317':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(301);"><div id="down-
arrow"></div></span';
      break;
case '318':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(302);"><div id="down-
arrow"></div></span';
```

```
      break;
case '319':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(1033);"><div id="down-
arrow"></div></span';
      break;
case '320':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(33);"><div id="down-arrow"></div></span';
      break;
case '321':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(35);"><div id="down-arrow"></div></span';
      break;
case '322':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(100);"><div id="down-
arrow"></div></span';
      break;
case '323':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(5611);"><div id="down-
arrow"></div></span';
      break;
 case '400':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(401);"><div id="down-
arrow"></div></span';
      break;
 case '401':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(1008);"><div id="down-
arrow"></div></span';
      break;
 case '402':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(134);"><div id="down-
arrow"></div></span';
      break;
 case '403':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(406);"><div id="down-
arrow"></div></span';
      break;
 case '404':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(1028);"><div id="down-
arrow"></div></span';
      break;
 case '405':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(406);"><div id="down-
arrow"></div></span';
      break;
 case '406':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(408);"><div id="down-
arrow"></div></span';
```

```
      break;
 case '408':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(409);"><div id="down-
arrow"></div></span';
      break;
 case '409':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(410);"><div id="down-
arrow"></div></span';
      break;
 case '410':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(403);"><div id="down-
arrow"></div></span';
      break;
 case '411':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(215);"><div id="down-
arrow"></div></span';
      break;
 case '412':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(413);"><div id="down-
arrow"></div></span';
      break;
 case '413':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(59);"><div id="down-arrow"></div></span';
      break;
 case '414':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(1031);"><div id="down-
arrow"></div></span';
      break;
 case '415':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(417);"><div id="down-
arrow"></div></span';
      break;
 case '416':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(300);"><div id="down-
arrow"></div></span';
      break;
 case '417':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(418);"><div id="down-
arrow"></div></span';
      break;
 case '418':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(419);"><div id="down-
arrow"></div></span';
      break;
 case '419':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(219);"><div id="down-
arrow"></div></span';
      break;
case '1000':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(1028);"><div id="down-
arrow"></div></span';
      break;
case '1001':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(1002);"><div id="down-
arrow"></div></span';
      break;
case '1002':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(1001);"><div id="down-
arrow"></div></span';
      break;
case '1003':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(1004);"><div id="down-
arrow"></div></span';
      break;
case '1004':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(67);"><div id="down-arrow"></div></span';
      break;
case '1005':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(1006);"><div id="down-
arrow"></div></span';
      break;
case '1006':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(68);"><div id="down-arrow"></div></span';
      break;
case '1007':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(18);"><div id="down-arrow"></div></span';
      break;
case '1008':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(49);"><div id="down-arrow"></div></span';
      break;
case '1009':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(49);"><div id="down-arrow"></div></span';
      break;
case '1010':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(49);"><div id="down-arrow"></div></span';
      break;
case '1011':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(119);"><div id="down-
arrow"></div></span';
      break;
```

```
case '1012':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(56);"><div id="down-arrow"></div></span';
      break;
case '1013':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(58);"><div id="down-arrow"></div></span';
      break;
case '1014':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(5611);"><div id="down-
arrow"></div></span';
      break;
case '1015':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(4655);"><div id="down-
arrow"></div></span';
      break;
case '1016':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(61);"><div id="down-arrow"></div></span';
      break;
case '1017':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(221);"><div id="down-
arrow"></div></span';
      break;
case '1019':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(226);"><div id="down-
arrow"></div></span';
      break;
case '1020':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(164);"><div id="down-
arrow"></div></span';
      break;
case '1021':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(307);"><div id="down-
arrow"></div></span';
      break;
case '1022':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(322);"><div id="down-
arrow"></div></span';
      break;
case '1023':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(310);"><div id="down-
arrow"></div></span';
      break;
case '1024':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(1025);"><div id="down-
arrow"></div></span';
      break;
case '1025':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(1026);"><div id="down-
arrow"></div></span';
      break;
case '1026':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(322);"><div id="down-
arrow"></div></span';
      break;
case '1027':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(129);"><div id="down-
arrow"></div></span';
      break;
 case '1028':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(406);"><div id="down-
arrow"></div></span';
      break;
 case '1029':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(408);"><div id="down-
arrow"></div></span';
      break;
 case '1030':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(1000);"><div id="down-
arrow"></div></span';
      break;
 case '1031':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(1026);"><div id="down-
arrow"></div></span';
      break;
 case '1032':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(1020);"><div id="down-
arrow"></div></span';
      break;
case '1033':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(409);"><div id="down-
arrow"></div></span';
      break;
case '1034':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrighthori(263);"><div id="down-
arrow"></div></span';
      break;
default:
      echo 'whoops';    }?>
```

```
/*
___toprightvert.php___
by Whitney Anne Trettien

This file loads the vertical links for the top right quadrant.
*/

<?php $cOption = $_GET['o'];  switch($cOption) {
case '1':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(2);"><div id="left-arrow"></div></span>';
      break;
case '2':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(48);"><div id="left-arrow"></div></span';
      break;
case '3':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(4);"><div id="left-arrow"></div></span';
      break;
case '4':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(5);"><div id="left-arrow"></div></span';
      break;
case '5':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(7);"><div id="left-arrow"></div></span>';
      break;
case '6':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(9);"><div id="left-arrow"></div></span';
      break;
case '7':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(9);"><div id="left-arrow"></div></span';
      break;
case '8':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(14);"><div id="left-arrow"></div></span';
      break;
case '9':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(177);"><div id="left-arrow"></div></span>';
      break;
case '10':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(28);"><div id="left-arrow"></div></span';
      break;
case '11':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(13);"><div id="left-arrow"></div></span';
      break;
case '12':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(53);"><div id="left-arrow"></div></span';
      break;
case '13':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(33);"><div id="left-arrow"></div></span>';
      break;
case '14':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(18);"><div id="left-arrow"></div></span';
      break;
case '15':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(16);"><div id="left-arrow"></div></span';
      break;
case '16':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(1007);"><div id="left-arrow"></div></span';
      break;
case '17':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(18);"><div id="left-arrow"></div></span>';
      break;
case '18':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(21);"><div id="left-arrow"></div></span';
      break;
case '19':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(400);"><div id="left-arrow"></div></span';
      break;
case '20':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(29);"><div id="left-arrow"></div></span';
      break;
case '21':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(46);"><div id="left-arrow"></div></span>';
      break;
case '22':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(23);"><div id="left-arrow"></div></span';
      break;
case '23':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(27);"><div id="left-arrow"></div></span';
      break;
case '24':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(25);"><div id="left-arrow"></div></span';
      break;
case '25':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(27);"><div id="left-arrow"></div></span>';
      break;
case '26':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(43);"><div id="left-arrow"></div></span';
      break;
case '27':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(29);"><div id="left-arrow"></div></span';
```

```
      break;
case '28':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(129);"><div id="left-arrow"></div></span';
      break;
case '29':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(48);"><div id="left-arrow"></div></span>';
      break;
case '30':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(31);"><div id="left-arrow"></div></span';
      break;
case '31':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(13);"><div id="left-arrow"></div></span';
      break;
case '32':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(59);"><div id="left-arrow"></div></span';
      break;
case '33':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(34);"><div id="left-arrow"></div></span>';
      break;
case '34':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(35);"><div id="left-arrow"></div></span';
      break;
case '35':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(40);"><div id="left-arrow"></div></span';
      break;
case '36':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(203);"><div id="left-arrow"></div></span';
      break;
case '37':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(39);"><div id="left-arrow"></div></span>';
      break;
case '38':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(37);"><div id="left-arrow"></div></span';
      break;
case '39':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(38);"><div id="left-arrow"></div></span';
      break;
case '40':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(4);"><div id="left-arrow"></div></span';
      break;
case '41':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(43);"><div id="left-arrow"></div></span>';
      break;
case '42':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(46);"><div id="left-arrow"></div></span';
      break;
case '43':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(1008);"><div id="left-arrow"></div></span';
      break;
case '44':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(45);"><div id="left-arrow"></div></span';
      break;
case '45':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(59);"><div id="left-arrow"></div></span>';
      break;
case '46':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(4611);"><div id="left-arrow"></div></span';
      break;
case '4611':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(4622);"><div id="left-arrow"></div></span';
      break;
case '4622':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(4655);"><div id="left-arrow"></div></span';
      break;
case '4633':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(4644);"><div id="left-arrow"></div></span';
      break;
case '4644':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(100);"><div id="left-arrow"></div></span';
      break;
case '4655':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(38);"><div id="left-arrow"></div></span';
      break;
case '47':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(49);"><div id="left-arrow"></div></span';
      break;
case '48':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(1009);"><div id="left-arrow"></div></span';
      break;
case '49':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(119);"><div id="left-arrow"></div></span>';
      break;
case '53':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(1001);"><div id="left-arrow"></div></span';
      break;
case '54':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(57);"><div id="left-arrow"></div></span>';
```

```
      break;
case '55':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(5522);"><div id="left-arrow"></div></span';
      break;
case '5511':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(5611);"><div id="left-arrow"></div></span';
      break;
case '5522':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(1014);"><div id="left-arrow"></div></span';
      break;
case '56':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(39);"><div id="left-arrow"></div></span';
      break;
case '5611':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(5622);"><div id="left-arrow"></div></span';
      break;
case '5622':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(56);"><div id="left-arrow"></div></span';
      break;
case '5633':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(1012);"><div id="left-arrow"></div></span';
      break;
case '5644':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(57);"><div id="left-arrow"></div></span';
      break;
case '57':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(3);"><div id="left-arrow"></div></span';
      break;
case '58':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(59);"><div id="left-arrow"></div></span>';
      break;
case '59':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(60);"><div id="left-arrow"></div></span';
      break;
case '60':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(43);"><div id="left-arrow"></div></span';
      break;
case '61':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(62);"><div id="left-arrow"></div></span';
      break;
case '62':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(64);"><div id="left-arrow"></div></span>';
      break;
case '63':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(64);"><div id="left-arrow"></div></span';
      break;
case '64':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(1003);"><div id="left-arrow"></div></span';
      break;
case '65':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(66);"><div id="left-arrow"></div></span';
      break;
case '66':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(1005);"><div id="left-arrow"></div></span>';
      break;
case '67':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(68);"><div id="left-arrow"></div></span';
      break;
case '68':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(18);"><div id="left-arrow"></div></span';
      break;
case '100':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(101);"><div id="left-arrow"></div></span>';
      break;
case '101':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(1002);"><div id="left-arrow"></div></span';
      break;
case '102':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(103);"><div id="left-arrow"></div></span';
      break;
case '103':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(105);"><div id="left-arrow"></div></span>';
      break;
case '104':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(103);"><div id="left-arrow"></div></span';
      break;
case '105':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(107);"><div id="left-arrow"></div></span';
      break;
case '106':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(109);"><div id="left-arrow"></div></span';
      break;
case '107':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(13111);"><div id="left-arrow"></div></span';
      break;
case '108':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(111);"><div id="left-arrow"></div></span';
```

```
      break;
case '109':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(42);"><div id="left-arrow"></div></span';
      break;
case '110':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(108);"><div id="left-arrow"></div></span';
      break;
case '111':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(112);"><div id="left-arrow"></div></span';
      break;
case '112':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(100);"><div id="left-arrow"></div></span';
      break;
case '113':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(140);"><div id="left-arrow"></div></span';
      break;
case '114':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(122);"><div id="left-arrow"></div></span';
      break;
case '115':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(111);"><div id="left-arrow"></div></span';
      break;
case '116':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(118);"><div id="left-arrow"></div></span';
      break;
case '117':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(134);"><div id="left-arrow"></div></span';
      break;
case '118':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(102);"><div id="left-arrow"></div></span';
      break;
case '119':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(11911);"><div id="left-arrow"></div></span';
      break;
case '11911':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(120);"><div id="left-arrow"></div></span';
      break;
case '120':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(126);"><div id="left-arrow"></div></span';
      break;
case '12011':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(4);"><div id="left-arrow"></div></span';
      break;
case '121':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(59);"><div id="left-arrow"></div></span';
      break;
case '122':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(102);"><div id="left-arrow"></div></span';
      break;
case '123':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(124);"><div id="left-arrow"></div></span';
      break;
case '124':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(403);"><div id="left-arrow"></div></span';
      break;
case '126':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(128);"><div id="left-arrow"></div></span';
      break;
case '127':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(128);"><div id="left-arrow"></div></span';
      break;
case '128':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(129);"><div id="left-arrow"></div></span';
      break;
case '129':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(100);"><div id="left-arrow"></div></span';
      break;
case '12911':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(150);"><div id="left-arrow"></div></span';
      break;
case '130':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(131);"><div id="left-arrow"></div></span';
      break;
case '131':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(13111);"><div id="left-arrow"></div></span';
      break;
case '13111':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(132);"><div id="left-arrow"></div></span';
      break;
case '13122':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(148);"><div id="left-arrow"></div></span';
      break;
case '132':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(14711);"><div id="left-arrow"></div></span';
      break;
case '133':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(14822);"><div id="left-arrow"></div></span';
```

```
      break;
case '134':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(138);"><div id="left-arrow"></div></span';
      break;
case '135':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(137);"><div id="left-arrow"></div></span';
      break;
case '136':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(141);"><div id="left-arrow"></div></span';
      break;
case '137':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(130);"><div id="left-arrow"></div></span';
      break;
case '138':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(140);"><div id="left-arrow"></div></span';
      break;
case '139':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(43);"><div id="left-arrow"></div></span';
      break;
case '140':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(111);"><div id="left-arrow"></div></span';
      break;
case '141':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(142);"><div id="left-arrow"></div></span';
      break;
case '142':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(143);"><div id="left-arrow"></div></span';
      break;
case '143':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(144);"><div id="left-arrow"></div></span';
      break;
case '144':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(146);"><div id="left-arrow"></div></span';
      break;
case '146':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(13122);"><div id="left-arrow"></div></span';
      break;
case '147':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(14711);"><div id="left-arrow"></div></span';
      break;
case '14711':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(149);"><div id="left-arrow"></div></span';
      break;
case '14722':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(148);"><div id="left-arrow"></div></span';
      break;
case '148':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(49);"><div id="left-arrow"></div></span';
      break;
case '14811':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(13);"><div id="left-arrow"></div></span';
      break;
case '14822':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(113);"><div id="left-arrow"></div></span';
      break;
case '14833':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(1000);"><div id="left-arrow"></div></span';
      break;
case '149':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(154);"><div id="left-arrow"></div></span';
      break;
case '150':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(151);"><div id="left-arrow"></div></span';
      break;
case '151':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(152);"><div id="left-arrow"></div></span';
      break;
case '152':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(153);"><div id="left-arrow"></div></span';
      break;
case '153':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(4);"><div id="left-arrow"></div></span';
      break;
case '154':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(155);"><div id="left-arrow"></div></span';
      break;
case '155':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(158);"><div id="left-arrow"></div></span';
      break;
case '156':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(157);"><div id="left-arrow"></div></span';
      break;
case '157':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(129);"><div id="left-arrow"></div></span';
      break;
case '158':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(140);"><div id="left-arrow"></div></span';
```

```
      break;
case '159':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(156);"><div id="left-arrow"></div></span';
      break;
case '160':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(1034);"><div id="left-arrow"></div></span';
      break;
case '161':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(162);"><div id="left-arrow"></div></span';
      break;
case '162':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(1020);"><div id="left-arrow"></div></span';
      break;
case '163':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(164);"><div id="left-arrow"></div></span';
      break;
case '164':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(130);"><div id="left-arrow"></div></span';
      break;
case '170':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(171);"><div id="left-arrow"></div></span';
      break;
case '171':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(53);"><div id="left-arrow"></div></span';
      break;
case '172':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(173);"><div id="left-arrow"></div></span';
      break;
case '173':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(177);"><div id="left-arrow"></div></span';
      break;
case '174':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(176);"><div id="left-arrow"></div></span';
      break;
case '175':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(178);"><div id="left-arrow"></div></span';
      break;
case '176':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(14811);"><div id="left-arrow"></div></span';
      break;
case '177':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(176);"><div id="left-arrow"></div></span';
      break;
case '178':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(179);"><div id="left-arrow"></div></span';
      break;
case '179':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(403);"><div id="left-arrow"></div></span';
      break;
case '180':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(181);"><div id="left-arrow"></div></span';
      break;
case '181':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(184);"><div id="left-arrow"></div></span';
      break;
case '182':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(183);"><div id="left-arrow"></div></span';
      break;
case '183':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(185);"><div id="left-arrow"></div></span';
      break;
case '184':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(185);"><div id="left-arrow"></div></span';
      break;
case '185':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(403);"><div id="left-arrow"></div></span';
      break;
case '186':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(171);"><div id="left-arrow"></div></span';
      break;
case '190':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(1000);"><div id="left-arrow"></div></span';
      break;
case '200':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(201);"><div id="left-arrow"></div></span';
      break;
case '201':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(202);"><div id="left-arrow"></div></span';
      break;
case '202':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(203);"><div id="left-arrow"></div></span';
      break;
case '203':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(204);"><div id="left-arrow"></div></span';
      break;
case '204':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(205);"><div id="left-arrow"></div></span';
```

```
      break;
case '205':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(206);"><div id="left-arrow"></div></span';
      break;
case '206':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(207);"><div id="left-arrow"></div></span';
      break;
case '207':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(4);"><div id="left-arrow"></div></span';
      break;
case '208':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(210);"><div id="left-arrow"></div></span';
      break;
case '209':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(173);"><div id="left-arrow"></div></span';
      break;
case '210':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(211);"><div id="left-arrow"></div></span';
      break;
case '211':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(212);"><div id="left-arrow"></div></span';
      break;
case '212':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(213);"><div id="left-arrow"></div></span';
      break;
case '213':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(215);"><div id="left-arrow"></div></span';
      break;
case '214':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(303);"><div id="left-arrow"></div></span';
      break;
case '215':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(216);"><div id="left-arrow"></div></span';
      break;
case '216':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(222);"><div id="left-arrow"></div></span';
      break;
case '219':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(214);"><div id="left-arrow"></div></span';
      break;
case '220':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(221);"><div id="left-arrow"></div></span';
      break;
case '221':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(222);"><div id="left-arrow"></div></span';
      break;
case '222':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(223);"><div id="left-arrow"></div></span';
      break;
case '223':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(224);"><div id="left-arrow"></div></span';
      break;
case '224':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(225);"><div id="left-arrow"></div></span';
      break;
case '225':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(1019);"><div id="left-arrow"></div></span';
      break;
case '226':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(227);"><div id="left-arrow"></div></span';
      break;
case '227':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(228);"><div id="left-arrow"></div></span';
      break;
case '228':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(229);"><div id="left-arrow"></div></span';
      break;
case '229':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(230);"><div id="left-arrow"></div></span';
      break;
case '230':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(231);"><div id="left-arrow"></div></span';
      break;
case '231':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(250);"><div id="left-arrow"></div></span';
      break;
case '232':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(233);"><div id="left-arrow"></div></span';
      break;
case '233':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(205);"><div id="left-arrow"></div></span';
      break;
case '234':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(219);"><div id="left-arrow"></div></span';
      break;
case '235':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(208);"><div id="left-arrow"></div></span';
```

```
      break;
case '250':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(251);"><div id="left-arrow"></div></span';
      break;
case '251':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(252);"><div id="left-arrow"></div></span';
      break;
case '252':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(253);"><div id="left-arrow"></div></span';
      break;
case '253':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(254);"><div id="left-arrow"></div></span';
      break;
case '254':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(255);"><div id="left-arrow"></div></span';
      break;
case '255':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(256);"><div id="left-arrow"></div></span';
      break;
case '256':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(263);"><div id="left-arrow"></div></span';
      break;
case '257':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(143);"><div id="left-arrow"></div></span';
      break;
case '258':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(259);"><div id="left-arrow"></div></span';
      break;
case '259':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(260);"><div id="left-arrow"></div></span';
      break;
case '260':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(161);"><div id="left-arrow"></div></span';
      break;
case '261':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(129);"><div id="left-arrow"></div></span';
      break;
case '262':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(319);"><div id="left-arrow"></div></span';
      break;
case '263':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(264);"><div id="left-arrow"></div></span';
      break;
case '264':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(265);"><div id="left-arrow"></div></span';
      break;
case '265':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(266);"><div id="left-arrow"></div></span';
      break;
case '266':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(2);"><div id="left-arrow"></div></span';
      break;
case '267':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(2);"><div id="left-arrow"></div></span';
      break;
case '300':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(303);"><div id="left-arrow"></div></span';
      break;
case '301':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(59);"><div id="left-arrow"></div></span';
      break;
case '302':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(14);"><div id="left-arrow"></div></span';
      break;
case '303':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(320);"><div id="left-arrow"></div></span';
      break;
case '304':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(305);"><div id="left-arrow"></div></span';
      break;
case '305':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(306);"><div id="left-arrow"></div></span';
      break;
case '306':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(307);"><div id="left-arrow"></div></span';
      break;
case '307':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(171);"><div id="left-arrow"></div></span';
      break;
case '308':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(309);"><div id="left-arrow"></div></span';
      break;
case '309':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(310);"><div id="left-arrow"></div></span';
      break;
case '310':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(1024);"><div id="left-arrow"></div></span';
```

```
      break;
case '311':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(312);"><div id="left-arrow"></div></span';
      break;
case '312':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(1026);"><div id="left-arrow"></div></span';
      break;
case '313':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(315);"><div id="left-arrow"></div></span';
      break;
case '314':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(172);"><div id="left-arrow"></div></span';
      break;
case '315':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(316);"><div id="left-arrow"></div></span';
      break;
case '316':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(317);"><div id="left-arrow"></div></span';
      break;
case '317':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(319);"><div id="left-arrow"></div></span';
      break;
case '318':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(319);"><div id="left-arrow"></div></span';
      break;
case '319':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(4611);"><div id="left-arrow"></div></span';
      break;
case '320':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(321);"><div id="left-arrow"></div></span';
      break;
case '321':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(323);"><div id="left-arrow"></div></span';
      break;
case '322':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(3);"><div id="left-arrow"></div></span';
      break;
case '323':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(1027);"><div id="left-arrow"></div></span';
      break;

case '400':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(402);"><div id="left-arrow"></div></span';
      break;
```

```
case '401':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(402);"><div id="left-arrow"></div></span';
      break;
case '402':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(403);"><div id="left-arrow"></div></span';
      break;
case '403':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(404);"><div id="left-arrow"></div></span';
      break;
case '404':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(405);"><div id="left-arrow"></div></span';
      break;
case '405':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(1029);"><div id="left-arrow"></div></span';
      break;
case '406':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(315);"><div id="left-arrow"></div></span';
      break;
case '408':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(4);"><div id="left-arrow"></div></span';
      break;
case '409':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(1030);"><div id="left-arrow"></div></span';
      break;
case '410':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(411);"><div id="left-arrow"></div></span';
      break;
case '411':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(414);"><div id="left-arrow"></div></span';
      break;
case '412':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(415);"><div id="left-arrow"></div></span';
      break;
case '413':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(212);"><div id="left-arrow"></div></span';
      break;
case '414':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(412);"><div id="left-arrow"></div></span';
      break;
case '415':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(419);"><div id="left-arrow"></div></span';
      break;
case '416':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(417);"><div id="left-arrow"></div></span';
      break;
case '417':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(419);"><div id="left-arrow"></div></span';
      break;
case '418':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(262);"><div id="left-arrow"></div></span';
      break;
case '419':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(173);"><div id="left-arrow"></div></span';
      break;
case '1000':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(8);"><div id="left-arrow"></div></span';
      break;
case '1001':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(54);"><div id="left-arrow"></div></span';
      break;
case '1002':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(55);"><div id="left-arrow"></div></span';
      break;
case '1003':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(66);"><div id="left-arrow"></div></span';
      break;
case '1004':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(1005);"><div id="left-arrow"></div></span';
      break;
case '1005':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(67);"><div id="left-arrow"></div></span';
      break;
case '1006':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(15);"><div id="left-arrow"></div></span';
      break;
case '1007':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(17);"><div id="left-arrow"></div></span';
      break;
case '1008':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(44);"><div id="left-arrow"></div></span';
      break;
case '1009':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(4611);"><div id="left-arrow"></div></span';
      break;
case '1010':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(4611);"><div id="left-arrow"></div></span';
```

```
      break;
case '1011':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(114);"><div id="left-arrow"></div></span';
      break;
case '1012':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(57);"><div id="left-arrow"></div></span';
      break;
case '1013':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(5622);"><div id="left-arrow"></div></span';
      break;
case '1014':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(5644);"><div id="left-arrow"></div></span';
      break;
case '1015':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(4622);"><div id="left-arrow"></div></span';
      break;
case '1016':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(60);"><div id="left-arrow"></div></span';
      break;
case '1017':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(223);"><div id="left-arrow"></div></span';
      break;
case '1019':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(225);"><div id="left-arrow"></div></span';
      break;
case '1020':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(163);"><div id="left-arrow"></div></span';
      break;
case '1021':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(306);"><div id="left-arrow"></div></span';
      break;
case '1022':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(323);"><div id="left-arrow"></div></span';
      break;
case '1023':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(1024);"><div id="left-arrow"></div></span';
      break;
case '1024':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(311);"><div id="left-arrow"></div></span';
      break;
case '1025':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(312);"><div id="left-arrow"></div></span';
      break;
case '1026':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(314);"><div id="left-arrow"></div></span';
      break;
case '1027':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(5633);"><div id="left-arrow"></div></span';
      break;
case '1028':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(1029);"><div id="left-arrow"></div></span';
      break;
case '1029':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(406);"><div id="left-arrow"></div></span';
      break;
case '1030':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(1028);"><div id="left-arrow"></div></span';
      break;
case '1031':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(1029);"><div id="left-arrow"></div></span';
      break;
case '1032':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(179);"><div id="left-arrow"></div></span';
      break;
case '1033':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(1031);"><div id="left-arrow"></div></span';
      break;
case '1034':
      echo '<span style="cursor: pointer"
onclick="loadContentTopleftvert(1029);"><div id="left-arrow"></div></span';
      break;
default:
      echo 'whoops';    }?>
```

```
/*
___bottomlefthori.php___
by Whitney Anne Trettien

This file loads the horizontal links for the bottom left quadrant.
*/

<?php $cOption = $_GET['o'];  switch($cOption) {
case '1':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(34);"><div id="up-arrow"></div></span>';
      break;
case '2':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(3);"><div id="up-arrow"></div></span';
      break;
case '3':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(46);"><div id="up-arrow"></div></span';
      break;
case '4':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(1000);"><div id="up-arrow"></div></span';
      break;
case '5':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(6);"><div id="up-arrow"></div></span>';
      break;
case '6':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(8);"><div id="up-arrow"></div></span';
      break;
case '7':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(8);"><div id="up-arrow"></div></span';
      break;
case '8':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(11);"><div id="up-arrow"></div></span';
      break;
case '9':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(10);"><div id="up-arrow"></div></span>';
      break;
case '10':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(34);"><div id="up-arrow"></div></span';
      break;
case '11':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(12);"><div id="up-arrow"></div></span';
      break;
case '12':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(13);"><div id="up-arrow"></div></span';
      break;
case '13':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(28);"><div id="up-arrow"></div></span>';
      break;
case '14':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(15);"><div id="up-arrow"></div></span';
      break;
case '15':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(34);"><div id="up-arrow"></div></span';
      break;
case '16':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(17);"><div id="up-arrow"></div></span';
      break;
case '17':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(21);"><div id="up-arrow"></div></span>';
      break;
case '18':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(19);"><div id="up-arrow"></div></span';
      break;
case '19':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(20);"><div id="up-arrow"></div></span';
      break;
case '20':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(21);"><div id="up-arrow"></div></span';
      break;
case '21':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(22);"><div id="up-arrow"></div></span>';
      break;
case '22':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(31);"><div id="up-arrow"></div></span';
      break;
case '23':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(24);"><div id="up-arrow"></div></span';
      break;
case '24':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(26);"><div id="up-arrow"></div></span';
      break;
case '25':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(1030);"><div id="up-arrow"></div></span>';
      break;
case '26':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(207);"><div id="up-arrow"></div></span';
      break;
case '27':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(53);"><div id="up-arrow"></div></span';
```

```
      break;
case '28':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(59);"><div id="up-arrow"></div></span';
      break;
case '29':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(30);"><div id="up-arrow"></div></span>';
      break;
case '30':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(53);"><div id="up-arrow"></div></span';
      break;
case '31':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(37);"><div id="up-arrow"></div></span';
      break;
case '32':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(11);"><div id="up-arrow"></div></span';
      break;
case '33':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(35);"><div id="up-arrow"></div></span>';
      break;
case '34':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(36);"><div id="up-arrow"></div></span';
      break;
case '35':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(39);"><div id="up-arrow"></div></span';
      break;
case '36':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(409);"><div id="up-arrow"></div></span';
      break;
case '37':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(46);"><div id="up-arrow"></div></span>';
      break;
case '38':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(110);"><div id="up-arrow"></div></span';
      break;
case '39':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(130);"><div id="up-arrow"></div></span';
      break;
case '40':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(41);"><div id="up-arrow"></div></span';
      break;
case '41':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(42);"><div id="up-arrow"></div></span>';
      break;
case '42':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(44);"><div id="up-arrow"></div></span';
      break;
case '43':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(44);"><div id="up-arrow"></div></span';
      break;
case '44':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(175);"><div id="up-arrow"></div></span';
      break;
case '45':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(13);"><div id="up-arrow"></div></span>';
      break;
case '46':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(48);"><div id="up-arrow"></div></span';
      break;
case '4611':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(1015);"><div id="up-arrow"></div></span';
      break;
case '4622':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(4633);"><div id="up-arrow"></div></span';
      break;
case '4633':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(1030);"><div id="up-arrow"></div></span';
      break;
case '4644':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(4655);"><div id="up-arrow"></div></span';
      break;
case '4655':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(175);"><div id="up-arrow"></div></span';
      break;
case '47':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(48);"><div id="up-arrow"></div></span';
      break;
case '48':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(1010);"><div id="up-arrow"></div></span';
      break;
case '49':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(57);"><div id="up-arrow"></div></span>';
      break;
case '53':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(1002);"><div id="up-arrow"></div></span';
      break;
case '54':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(56);"><div id="up-arrow"></div></span>';
```

```
      break;
case '55':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(5511);"><div id="up-arrow"></div></span';
      break;
case '5511':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(5644);"><div id="up-arrow"></div></span';
      break;
case '5522':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(5611);"><div id="up-arrow"></div></span';
      break;
case '56':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(58);"><div id="up-arrow"></div></span';
      break;
case '5611':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(1013);"><div id="up-arrow"></div></span';
      break;
case '5622':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(5633);"><div id="up-arrow"></div></span';
      break;
case '5633':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(57);"><div id="up-arrow"></div></span';
      break;
case '5644':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(5522);"><div id="up-arrow"></div></span';
      break;
case '57':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(56);"><div id="up-arrow"></div></span';
      break;
case '58':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(28);"><div id="up-arrow"></div></span>';
      break;
case '59':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(1016);"><div id="up-arrow"></div></span';
      break;
case '60':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(61);"><div id="up-arrow"></div></span';
      break;
case '61':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(43);"><div id="up-arrow"></div></span';
      break;
case '62':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(63);"><div id="up-arrow"></div></span>';
      break;
case '63':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(38);"><div id="up-arrow"></div></span';
      break;
case '64':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(65);"><div id="up-arrow"></div></span';
      break;
case '65':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(1004);"><div id="up-arrow"></div></span';
      break;
case '66':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(67);"><div id="up-arrow"></div></span>';
      break;
case '67':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(1006);"><div id="up-arrow"></div></span';
      break;
case '68':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(1);"><div id="up-arrow"></div></span';
      break;
case '100':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(401);"><div id="up-arrow"></div></span>';
      break;
case '101':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(130);"><div id="up-arrow"></div></span';
      break;
case '102':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(104);"><div id="up-arrow"></div></span';
      break;
case '103':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(104);"><div id="up-arrow"></div></span>';
      break;
case '104':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(105);"><div id="up-arrow"></div></span';
      break;
case '105':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(106);"><div id="up-arrow"></div></span';
      break;
case '106':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(107);"><div id="up-arrow"></div></span';
      break;
case '107':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(102);"><div id="up-arrow"></div></span';
      break;
case '108':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(121);"><div id="up-arrow"></div></span';
```

```
      break;
case '109':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(133);"><div id="up-arrow"></div></span';
      break;
case '110':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(115);"><div id="up-arrow"></div></span';
      break;
case '111':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(117);"><div id="up-arrow"></div></span';
      break;
case '112':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(41);"><div id="up-arrow"></div></span';
      break;
case '113':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(117);"><div id="up-arrow"></div></span';
      break;
case '114':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(113);"><div id="up-arrow"></div></span';
      break;
case '115':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(116);"><div id="up-arrow"></div></span';
      break;
case '116':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(126);"><div id="up-arrow"></div></span';
      break;
case '117':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(113);"><div id="up-arrow"></div></span';
      break;
case '118':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(32);"><div id="up-arrow"></div></span';
      break;
case '119':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(120);"><div id="up-arrow"></div></span';
      break;
case '11911':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(118);"><div id="up-arrow"></div></span';
      break;
case '120':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(12011);"><div id="up-arrow"></div></span';
      break;
case '12011':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(14811);"><div id="up-arrow"></div></span';
      break;
case '121':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(123);"><div id="up-arrow"></div></span';
      break;
case '122':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(1011);"><div id="up-arrow"></div></span';
      break;
case '123':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(125);"><div id="up-arrow"></div></span';
      break;
case '124':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(141);"><div id="up-arrow"></div></span';
      break;
case '126':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(127);"><div id="up-arrow"></div></span';
      break;
case '127':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(129);"><div id="up-arrow"></div></span';
      break;
case '128':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(130);"><div id="up-arrow"></div></span';
      break;
case '129':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(130);"><div id="up-arrow"></div></span';
      break;
case '12911':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(4);"><div id="up-arrow"></div></span';
      break;
case '130':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(134);"><div id="up-arrow"></div></span';
      break;
case '131':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(12011);"><div id="up-arrow"></div></span';
      break;
case '13111':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(13122);"><div id="up-arrow"></div></span';
      break;
case '13122':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(132);"><div id="up-arrow"></div></span';
      break;
case '132':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(133);"><div id="up-arrow"></div></span';
      break;
case '133':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(109);"><div id="up-arrow"></div></span';
```

```
      break;
case '134':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(135);"><div id="up-arrow"></div></span';
      break;
case '135':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(136);"><div id="up-arrow"></div></span';
      break;
case '136':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(126);"><div id="up-arrow"></div></span';
      break;
case '137':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(141);"><div id="up-arrow"></div></span';
      break;
case '138':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(139);"><div id="up-arrow"></div></span';
      break;
case '139':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(117);"><div id="up-arrow"></div></span';
      break;
case '140':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(158);"><div id="up-arrow"></div></span';
      break;
case '141':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(143);"><div id="up-arrow"></div></span';
      break;
case '142':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(57);"><div id="up-arrow"></div></span';
      break;
case '143':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(14811);"><div id="up-arrow"></div></span';
      break;
case '144':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(41);"><div id="up-arrow"></div></span';
      break;
case '146':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(147);"><div id="up-arrow"></div></span';
      break;
case '147':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(148);"><div id="up-arrow"></div></span';
      break;
case '14711':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(14722);"><div id="up-arrow"></div></span';
      break;
case '14722':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(14811);"><div id="up-arrow"></div></span';
      break;
case '148':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(1012);"><div id="up-arrow"></div></span';
      break;
case '14811':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(37);"><div id="up-arrow"></div></span';
      break;
case '14822':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(149);"><div id="up-arrow"></div></span';
      break;
case '14833':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(100);"><div id="up-arrow"></div></span';
      break;
case '149':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(150);"><div id="up-arrow"></div></span';
      break;
case '150':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(152);"><div id="up-arrow"></div></span';
      break;
case '151':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(153);"><div id="up-arrow"></div></span';
      break;
case '152':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(115);"><div id="up-arrow"></div></span';
      break;
case '153':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(404);"><div id="up-arrow"></div></span';
      break;
case '154':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(156);"><div id="up-arrow"></div></span';
      break;
case '155':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(159);"><div id="up-arrow"></div></span';
      break;
case '156':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(128);"><div id="up-arrow"></div></span';
      break;
case '157':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(171);"><div id="up-arrow"></div></span';
      break;
case '158':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(157);"><div id="up-arrow"></div></span';
```

```
      break;
case '159':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(1008);"><div id="up-arrow"></div></span';
      break;
case '160':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(156);"><div id="up-arrow"></div></span';
      break;
case '161':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(1020);"><div id="up-arrow"></div></span';
      break;
case '162':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(163);"><div id="up-arrow"></div></span';
      break;
case '163':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(219);"><div id="up-arrow"></div></span';
      break;
case '164':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(250);"><div id="up-arrow"></div></span';
      break;
case '170':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(172);"><div id="up-arrow"></div></span';
      break;
case '171':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(172);"><div id="up-arrow"></div></span';
      break;
case '172':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(180);"><div id="up-arrow"></div></span';
      break;
case '173':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(174);"><div id="up-arrow"></div></span';
      break;
case '174':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(177);"><div id="up-arrow"></div></span';
      break;

case '175':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(176);"><div id="up-arrow"></div></span';
      break;
case '176':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(178);"><div id="up-arrow"></div></span';
      break;
case '177':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(178);"><div id="up-arrow"></div></span';
      break;
```

```
case '178':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(1032);"><div id="up-arrow"></div></span';
      break;
case '179':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(180);"><div id="up-arrow"></div></span';
      break;
case '180':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(182);"><div id="up-arrow"></div></span';
      break;
case '181':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(182);"><div id="up-arrow"></div></span';
      break;
case '182':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(184);"><div id="up-arrow"></div></span';
      break;
case '183':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(184);"><div id="up-arrow"></div></span';
      break;
case '184':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(171);"><div id="up-arrow"></div></span';
      break;
case '185':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(134);"><div id="up-arrow"></div></span';
      break;
case '186':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(12011);"><div id="up-arrow"></div></span';
      break;
case '190':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(1029);"><div id="up-arrow"></div></span';
      break;
case '200':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(202);"><div id="up-arrow"></div></span';
      break;
case '201':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(203);"><div id="up-arrow"></div></span';
      break;
case '202':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(204);"><div id="up-arrow"></div></span';
      break;
case '203':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(232);"><div id="up-arrow"></div></span';
      break;
case '204':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(254);"><div id="up-arrow"></div></span';
      break;
case '205':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(207);"><div id="up-arrow"></div></span';
      break;
case '206':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(234);"><div id="up-arrow"></div></span';
      break;
case '207':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(208);"><div id="up-arrow"></div></span';
      break;
case '208':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(209);"><div id="up-arrow"></div></span';
      break;
case '209':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(211);"><div id="up-arrow"></div></span';
      break;
case '210':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(212);"><div id="up-arrow"></div></span';
      break;
case '211':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(213);"><div id="up-arrow"></div></span';
      break;
case '212':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(214);"><div id="up-arrow"></div></span';
      break;
case '213':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(216);"><div id="up-arrow"></div></span';
      break;
case '214':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(34);"><div id="up-arrow"></div></span';
      break;
case '215':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(219);"><div id="up-arrow"></div></span';
      break;
case '216':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(234);"><div id="up-arrow"></div></span';
      break;
case '219':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(220);"><div id="up-arrow"></div></span';
      break;
case '220':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(1017);"><div id="up-arrow"></div></span';
```

```
      break;
case '221':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(223);"><div id="up-arrow"></div></span';
      break;
case '222':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(1017);"><div id="up-arrow"></div></span';
      break;
case '223':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(225);"><div id="up-arrow"></div></span';
      break;
case '224':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(1019);"><div id="up-arrow"></div></span';
      break;
case '225':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(226);"><div id="up-arrow"></div></span';
      break;
case '226':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(228);"><div id="up-arrow"></div></span';
      break;
case '227':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(229);"><div id="up-arrow"></div></span';
      break;
case '228':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(300);"><div id="up-arrow"></div></span';
      break;
case '229':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(231);"><div id="up-arrow"></div></span';
      break;
case '230':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(19);"><div id="up-arrow"></div></span';
      break;
case '231':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(39);"><div id="up-arrow"></div></span';
      break;
case '232':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(205);"><div id="up-arrow"></div></span';
      break;
case '233':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(250);"><div id="up-arrow"></div></span';
      break;
case '234':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(303);"><div id="up-arrow"></div></span';
      break;
case '235':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(212);"><div id="up-arrow"></div></span';
      break;
case '250':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(253);"><div id="up-arrow"></div></span';
      break;
case '251':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(255);"><div id="up-arrow"></div></span';
      break;
case '252':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(255);"><div id="up-arrow"></div></span';
      break;
case '253':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(262);"><div id="up-arrow"></div></span';
      break;
case '254':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(4);"><div id="up-arrow"></div></span';
      break;
case '255':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(263);"><div id="up-arrow"></div></span';
      break;
case '256':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(257);"><div id="up-arrow"></div></span';
      break;
case '257':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(258);"><div id="up-arrow"></div></span';
      break;
case '258':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(205);"><div id="up-arrow"></div></span';
      break;
case '259':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(261);"><div id="up-arrow"></div></span';
      break;
case '260':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(14833);"><div id="up-arrow"></div></span';
      break;
case '261':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(230);"><div id="up-arrow"></div></span';
      break;
case '262':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(205);"><div id="up-arrow"></div></span';
      break;
case '263':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(265);"><div id="up-arrow"></div></span';
```

```
      break;
case '264':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(267);"><div id="up-arrow"></div></span';
      break;
case '265':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(267);"><div id="up-arrow"></div></span';
      break;
case '266':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(258);"><div id="up-arrow"></div></span';
      break;
case '267':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(143);"><div id="up-arrow"></div></span';
      break;
case '300':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(301);"><div id="up-arrow"></div></span';
      break;
case '301':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(302);"><div id="up-arrow"></div></span';
      break;
case '302':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(303);"><div id="up-arrow"></div></span';
      break;
case '303':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(304);"><div id="up-arrow"></div></span';
      break;
case '304':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(1021);"><div id="up-arrow"></div></span';
      break;
case '305':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(302);"><div id="up-arrow"></div></span';
      break;
case '306':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(1022);"><div id="up-arrow"></div></span';
      break;
case '307':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(308);"><div id="up-arrow"></div></span';
      break;
case '308':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(301);"><div id="up-arrow"></div></span';
      break;
case '309':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(1023);"><div id="up-arrow"></div></span';
      break;
case '310':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(311);"><div id="up-arrow"></div></span';
      break;
case '311':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(1025);"><div id="up-arrow"></div></span';
      break;
case '312':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(313);"><div id="up-arrow"></div></span';
      break;
case '313':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(322);"><div id="up-arrow"></div></span';
      break;
case '314':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(315);"><div id="up-arrow"></div></span';
      break;
case '315':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(317);"><div id="up-arrow"></div></span';
      break;
case '316':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(318);"><div id="up-arrow"></div></span';
      break;
case '317':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(301);"><div id="up-arrow"></div></span';
      break;
case '318':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(302);"><div id="up-arrow"></div></span';
      break;
case '319':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(1033);"><div id="up-arrow"></div></span';
      break;
case '320':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(33);"><div id="up-arrow"></div></span';
      break;
case '321':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(35);"><div id="up-arrow"></div></span';
      break;
case '322':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(100);"><div id="up-arrow"></div></span';
      break;
case '323':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(5611);"><div id="up-arrow"></div></span';
      break;
 case '400':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(401);"><div id="up-arrow"></div></span';
```

```
      break;
 case '401':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(1008);"><div id="up-arrow"></div></span';
      break;
 case '402':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(134);"><div id="up-arrow"></div></span';
      break;
 case '403':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(406);"><div id="up-arrow"></div></span';
      break;
 case '404':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(1028);"><div id="up-arrow"></div></span';
      break;
 case '405':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(406);"><div id="up-arrow"></div></span';
      break;
 case '406':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(408);"><div id="up-arrow"></div></span';
      break;
 case '408':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(409);"><div id="up-arrow"></div></span';
      break;
 case '409':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(410);"><div id="up-arrow"></div></span';
      break;
 case '410':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(403);"><div id="up-arrow"></div></span';
      break;
 case '411':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(215);"><div id="up-arrow"></div></span';
      break;
 case '412':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(413);"><div id="up-arrow"></div></span';
      break;
 case '413':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(59);"><div id="up-arrow"></div></span';
      break;
 case '414':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(1031);"><div id="up-arrow"></div></span';
      break;
 case '415':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(417);"><div id="up-arrow"></div></span';
      break;
 case '416':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(300);"><div id="up-arrow"></div></span';
      break;
 case '417':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(418);"><div id="up-arrow"></div></span';
      break;
 case '418':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(419);"><div id="up-arrow"></div></span';
      break;
 case '419':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(219);"><div id="up-arrow"></div></span';
      break;
case '1000':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(1028);"><div id="up-arrow"></div></span';
      break;
case '1001':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(1002);"><div id="up-arrow"></div></span';
      break;
case '1002':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(1001);"><div id="up-arrow"></div></span';
      break;
case '1003':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(1004);"><div id="up-arrow"></div></span';
      break;
case '1004':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(67);"><div id="up-arrow"></div></span';
      break;
case '1005':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(1006);"><div id="up-arrow"></div></span';
      break;
case '1006':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(68);"><div id="up-arrow"></div></span';
      break;
case '1007':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(18);"><div id="up-arrow"></div></span';
      break;
case '1008':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(49);"><div id="up-arrow"></div></span';
      break;
case '1009':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(49);"><div id="up-arrow"></div></span';
      break;
case '1010':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(49);"><div id="up-arrow"></div></span';
```

```
      break;
case '1011':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(119);"><div id="up-arrow"></div></span';
      break;
case '1012':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(56);"><div id="up-arrow"></div></span';
      break;
case '1013':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(58);"><div id="up-arrow"></div></span';
      break;
case '1014':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(5611);"><div id="up-arrow"></div></span';
      break;
case '1015':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(4655);"><div id="up-arrow"></div></span';
      break;
case '1016':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(61);"><div id="up-arrow"></div></span';
      break;
case '1017':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(221);"><div id="up-arrow"></div></span';
      break;
case '1019':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(226);"><div id="up-arrow"></div></span';
      break;
case '1020':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(164);"><div id="up-arrow"></div></span';
      break;
case '1021':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(307);"><div id="up-arrow"></div></span';
      break;
case '1022':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(322);"><div id="up-arrow"></div></span';
      break;
case '1023':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(310);"><div id="up-arrow"></div></span';
      break;
case '1024':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(1025);"><div id="up-arrow"></div></span';
      break;
case '1025':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(1026);"><div id="up-arrow"></div></span';
      break;
case '1026':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(322);"><div id="up-arrow"></div></span';
      break;
case '1027':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(129);"><div id="up-arrow"></div></span';
      break;
 case '1028':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(406);"><div id="up-arrow"></div></span';
      break;
 case '1029':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(408);"><div id="up-arrow"></div></span';
      break;
 case '1030':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(1000);"><div id="up-arrow"></div></span';
      break;
 case '1031':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(1026);"><div id="up-arrow"></div></span';
      break;
 case '1032':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(1020);"><div id="up-arrow"></div></span';
      break;
case '1033':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(409);"><div id="up-arrow"></div></span';
      break;
case '1034':
      echo '<span style="cursor: pointer"
onclick="loadContentToplefthori(263);"><div id="up-arrow"></div></span';
      break;
default:
      echo 'whoops';    }?>
```

```
/*
___bottomleftvert.php___
by Whitney Anne Trettien

This file loads the vertical links for the bottom left quadrant.
*/

<?php $cOption = $_GET['o'];  switch($cOption) {
case '1':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(2);"><div id="right-
arrow"></div></span>';
      break;
case '2':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(48);"><div id="right-
arrow"></div></span';
      break;
case '3':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(4);"><div id="right-arrow"></div></span';
      break;
case '4':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(5);"><div id="right-arrow"></div></span';
      break;
case '5':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(7);"><div id="right-
arrow"></div></span>';
      break;
case '6':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(9);"><div id="right-arrow"></div></span';
      break;
case '7':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(9);"><div id="right-arrow"></div></span';
      break;
case '8':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(14);"><div id="right-
arrow"></div></span';
      break;
case '9':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(177);"><div id="right-
arrow"></div></span>';
      break;
case '10':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(28);"><div id="right-
arrow"></div></span';
      break;
case '11':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(13);"><div id="right-
arrow"></div></span';
```

```
      break;
case '12':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(53);"><div id="right-
arrow"></div></span';
      break;
case '13':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(33);"><div id="right-
arrow"></div></span>';
      break;
case '14':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(18);"><div id="right-
arrow"></div></span';
      break;
case '15':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(16);"><div id="right-
arrow"></div></span';
      break;
case '16':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(1007);"><div id="right-
arrow"></div></span';
      break;
case '17':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(18);"><div id="right-
arrow"></div></span>';
      break;
case '18':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(21);"><div id="right-
arrow"></div></span';
      break;
case '19':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(400);"><div id="right-
arrow"></div></span';
      break;
case '20':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(29);"><div id="right-
arrow"></div></span';
      break;
case '21':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(46);"><div id="right-
arrow"></div></span>';
      break;
case '22':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(23);"><div id="right-
arrow"></div></span';
      break;
case '23':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(27);"><div id="right-
arrow"></div></span';
      break;
case '24':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(25);"><div id="right-
arrow"></div></span';
      break;
case '25':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(27);"><div id="right-
arrow"></div></span>';
      break;
case '26':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(43);"><div id="right-
arrow"></div></span';
      break;
case '27':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(29);"><div id="right-
arrow"></div></span';
      break;
case '28':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(129);"><div id="right-
arrow"></div></span';
      break;
case '29':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(48);"><div id="right-
arrow"></div></span>';
      break;
case '30':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(31);"><div id="right-
arrow"></div></span';
      break;
case '31':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(13);"><div id="right-
arrow"></div></span';
      break;
case '32':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(59);"><div id="right-
arrow"></div></span';
      break;
case '33':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(34);"><div id="right-
arrow"></div></span>';
      break;
case '34':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(35);"><div id="right-
arrow"></div></span';
```

```
      break;
case '35':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(40);"><div id="right-
arrow"></div></span';
      break;
case '36':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(203);"><div id="right-
arrow"></div></span';
      break;
case '37':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(39);"><div id="right-
arrow"></div></span>';
      break;
case '38':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(37);"><div id="right-
arrow"></div></span';
      break;
case '39':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(38);"><div id="right-
arrow"></div></span';
      break;
case '40':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(4);"><div id="right-arrow"></div></span';
      break;
case '41':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(43);"><div id="right-
arrow"></div></span>';
      break;
case '42':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(46);"><div id="right-
arrow"></div></span';
      break;
case '43':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(1008);"><div id="right-
arrow"></div></span';
      break;
case '44':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(45);"><div id="right-
arrow"></div></span';
      break;
case '45':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(59);"><div id="right-
arrow"></div></span>';
      break;
case '46':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(4611);"><div id="right-
arrow"></div></span';
      break;
case '4611':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(4622);"><div id="right-
arrow"></div></span';
      break;
case '4622':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(4655);"><div id="right-
arrow"></div></span';
      break;
case '4633':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(4644);"><div id="right-
arrow"></div></span';
      break;
case '4644':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(100);"><div id="right-
arrow"></div></span';
      break;
case '4655':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(38);"><div id="right-
arrow"></div></span';
      break;
case '47':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(49);"><div id="right-
arrow"></div></span';
      break;
case '48':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(1009);"><div id="right-
arrow"></div></span';
      break;
case '49':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(119);"><div id="right-
arrow"></div></span>';
      break;
case '53':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(1001);"><div id="right-
arrow"></div></span';
      break;
case '54':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(57);"><div id="right-
arrow"></div></span>';
      break;
case '55':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(5522);"><div id="right-
arrow"></div></span';
```

```
      break;
case '5511':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(5611);"><div id="right-
arrow"></div></span';
      break;
case '5522':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(1014);"><div id="right-
arrow"></div></span';
      break;
case '56':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(39);"><div id="right-
arrow"></div></span';
      break;
case '5611':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(5622);"><div id="right-
arrow"></div></span';
      break;
case '5622':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(56);"><div id="right-
arrow"></div></span';
      break;
case '5633':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(1012);"><div id="right-
arrow"></div></span';
      break;
case '5644':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(57);"><div id="right-
arrow"></div></span';
      break;
case '57':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(3);"><div id="right-arrow"></div></span';
      break;
case '58':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(59);"><div id="right-
arrow"></div></span>';
      break;
case '59':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(60);"><div id="right-
arrow"></div></span';
      break;
case '60':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(43);"><div id="right-
arrow"></div></span';
      break;
case '61':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(62);"><div id="right-
arrow"></div></span';
      break;
case '62':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(64);"><div id="right-
arrow"></div></span>';
      break;
case '63':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(64);"><div id="right-
arrow"></div></span';
      break;
case '64':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(1003);"><div id="right-
arrow"></div></span';
      break;
case '65':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(66);"><div id="right-
arrow"></div></span';
      break;
case '66':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(1005);"><div id="right-
arrow"></div></span>';
      break;
case '67':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(68);"><div id="right-
arrow"></div></span';
      break;
case '68':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(18);"><div id="right-
arrow"></div></span';
      break;
case '100':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(101);"><div id="right-
arrow"></div></span>';
      break;
case '101':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(1002);"><div id="right-
arrow"></div></span';
      break;
case '102':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(103);"><div id="right-
arrow"></div></span';
      break;
case '103':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(105);"><div id="right-
arrow"></div></span>';
```

```
      break;
case '104':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(103);"><div id="right-
arrow"></div></span';
      break;
case '105':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(107);"><div id="right-
arrow"></div></span';
      break;
case '106':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(109);"><div id="right-
arrow"></div></span';
      break;
case '107':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(13111);"><div id="right-
arrow"></div></span';
      break;
case '108':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(111);"><div id="right-
arrow"></div></span';
      break;
case '109':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(42);"><div id="right-
arrow"></div></span';
      break;
case '110':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(108);"><div id="right-
arrow"></div></span';
      break;
case '111':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(112);"><div id="right-
arrow"></div></span';
      break;
case '112':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(100);"><div id="right-
arrow"></div></span';
      break;
case '113':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(140);"><div id="right-
arrow"></div></span';
      break;
case '114':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(122);"><div id="right-
arrow"></div></span';
      break;
case '115':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(111);"><div id="right-
arrow"></div></span';
      break;
case '116':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(118);"><div id="right-
arrow"></div></span';
      break;
case '117':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(134);"><div id="right-
arrow"></div></span';
      break;
case '118':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(102);"><div id="right-
arrow"></div></span';
      break;
case '119':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(11911);"><div id="right-
arrow"></div></span';
      break;
case '11911':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(120);"><div id="right-
arrow"></div></span';
      break;
case '120':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(126);"><div id="right-
arrow"></div></span';
      break;
case '12011':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(4);"><div id="right-arrow"></div></span';
      break;
case '121':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(59);"><div id="right-
arrow"></div></span';
      break;
case '122':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(102);"><div id="right-
arrow"></div></span';
      break;
case '123':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(124);"><div id="right-
arrow"></div></span';
      break;
case '124':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(403);"><div id="right-
arrow"></div></span';
      break;
```

```
case '126':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(128);"><div id="right-
arrow"></div></span';
      break;
case '127':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(128);"><div id="right-
arrow"></div></span';
      break;
case '128':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(129);"><div id="right-
arrow"></div></span';
      break;
case '129':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(100);"><div id="right-
arrow"></div></span';
      break;
case '12911':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(150);"><div id="right-
arrow"></div></span';
      break;
case '130':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(131);"><div id="right-
arrow"></div></span';
      break;
case '131':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(13111);"><div id="right-
arrow"></div></span';
      break;
case '13111':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(132);"><div id="right-
arrow"></div></span';
      break;
case '13122':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(148);"><div id="right-
arrow"></div></span';
      break;
case '132':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(14711);"><div id="right-
arrow"></div></span';
      break;
case '133':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(14822);"><div id="right-
arrow"></div></span';
      break;
case '134':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(138);"><div id="right-
arrow"></div></span';
      break;
case '135':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(137);"><div id="right-
arrow"></div></span';
      break;
case '136':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(141);"><div id="right-
arrow"></div></span';
      break;
case '137':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(130);"><div id="right-
arrow"></div></span';
      break;
case '138':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(140);"><div id="right-
arrow"></div></span';
      break;
case '139':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(43);"><div id="right-
arrow"></div></span';
      break;
case '140':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(111);"><div id="right-
arrow"></div></span';
      break;
case '141':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(142);"><div id="right-
arrow"></div></span';
      break;
case '142':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(143);"><div id="right-
arrow"></div></span';
      break;
case '143':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(144);"><div id="right-
arrow"></div></span';
      break;
case '144':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(146);"><div id="right-
arrow"></div></span';
      break;
case '146':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(13122);"><div id="right-
arrow"></div></span';
```

```
      break;
case '147':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(14711);"><div id="right-
arrow"></div></span';
      break;
case '14711':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(149);"><div id="right-
arrow"></div></span';
      break;
case '14722':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(148);"><div id="right-
arrow"></div></span';
      break;
case '148':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(49);"><div id="right-
arrow"></div></span';
      break;
case '14811':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(13);"><div id="right-
arrow"></div></span';
      break;
case '14822':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(113);"><div id="right-
arrow"></div></span';
      break;
case '14833':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(1000);"><div id="right-
arrow"></div></span';
      break;
case '149':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(154);"><div id="right-
arrow"></div></span';
      break;
case '150':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(151);"><div id="right-
arrow"></div></span';
      break;
case '151':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(152);"><div id="right-
arrow"></div></span';
      break;
case '152':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(153);"><div id="right-
arrow"></div></span';
      break;
case '153':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(4);"><div id="right-arrow"></div></span';
      break;
case '154':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(155);"><div id="right-
arrow"></div></span';
      break;
case '155':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(158);"><div id="right-
arrow"></div></span';
      break;
case '156':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(157);"><div id="right-
arrow"></div></span';
      break;
case '157':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(129);"><div id="right-
arrow"></div></span';
      break;
case '158':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(140);"><div id="right-
arrow"></div></span';
      break;
case '159':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(156);"><div id="right-
arrow"></div></span';
      break;
case '160':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(1034);"><div id="right-
arrow"></div></span';
      break;
case '161':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(162);"><div id="right-
arrow"></div></span';
      break;
case '162':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(1020);"><div id="right-
arrow"></div></span';
      break;
case '163':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(164);"><div id="right-
arrow"></div></span';
      break;
case '164':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(130);"><div id="right-
arrow"></div></span';
      break;
```

```
case '170':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(171);"><div id="right-
arrow"></div></span';
      break;
case '171':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(53);"><div id="right-
arrow"></div></span';
      break;
case '172':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(173);"><div id="right-
arrow"></div></span';
      break;
case '173':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(177);"><div id="right-
arrow"></div></span';
      break;
case '174':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(176);"><div id="right-
arrow"></div></span';
      break;
case '175':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(178);"><div id="right-
arrow"></div></span';
      break;
case '176':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(14811);"><div id="right-
arrow"></div></span';
      break;
case '177':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(176);"><div id="right-
arrow"></div></span';
      break;
case '178':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(179);"><div id="right-
arrow"></div></span';
      break;
case '179':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(403);"><div id="right-
arrow"></div></span';
      break;
case '180':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(181);"><div id="right-
arrow"></div></span';
      break;
case '181':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(184);"><div id="right-
arrow"></div></span';
      break;
case '182':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(183);"><div id="right-
arrow"></div></span';
      break;
case '183':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(185);"><div id="right-
arrow"></div></span';
      break;
case '184':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(185);"><div id="right-
arrow"></div></span';
      break;
case '185':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(403);"><div id="right-
arrow"></div></span';
      break;
case '186':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(171);"><div id="right-
arrow"></div></span';
      break;
case '190':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(1000);"><div id="right-
arrow"></div></span';
      break;
case '200':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(201);"><div id="right-
arrow"></div></span';
      break;
case '201':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(202);"><div id="right-
arrow"></div></span';
      break;
case '202':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(203);"><div id="right-
arrow"></div></span';
      break;
case '203':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(204);"><div id="right-
arrow"></div></span';
      break;
case '204':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(205);"><div id="right-
arrow"></div></span';
```

```
      break;
case '205':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(206);"><div id="right-
arrow"></div></span';
      break;
case '206':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(207);"><div id="right-
arrow"></div></span';
      break;
case '207':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(4);"><div id="right-arrow"></div></span';
      break;
case '208':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(210);"><div id="right-
arrow"></div></span';
      break;
case '209':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(173);"><div id="right-
arrow"></div></span';
      break;
case '210':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(211);"><div id="right-
arrow"></div></span';
      break;
case '211':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(212);"><div id="right-
arrow"></div></span';
      break;
case '212':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(213);"><div id="right-
arrow"></div></span';
      break;
case '213':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(215);"><div id="right-
arrow"></div></span';
      break;
case '214':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(303);"><div id="right-
arrow"></div></span';
      break;
case '215':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(216);"><div id="right-
arrow"></div></span';
      break;
case '216':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(222);"><div id="right-
arrow"></div></span';
      break;
case '219':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(214);"><div id="right-
arrow"></div></span';
      break;
case '220':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(221);"><div id="right-
arrow"></div></span';
      break;
case '221':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(222);"><div id="right-
arrow"></div></span';
      break;
case '222':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(223);"><div id="right-
arrow"></div></span';
      break;
case '223':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(224);"><div id="right-
arrow"></div></span';
      break;
case '224':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(225);"><div id="right-
arrow"></div></span';
      break;
case '225':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(1019);"><div id="right-
arrow"></div></span';
      break;
case '226':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(227);"><div id="right-
arrow"></div></span';
      break;
case '227':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(228);"><div id="right-
arrow"></div></span';
      break;
case '228':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(229);"><div id="right-
arrow"></div></span';
      break;
case '229':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(230);"><div id="right-
arrow"></div></span';
```

```
      break;
case '230':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(231);"><div id="right-
arrow"></div></span';
      break;
case '231':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(250);"><div id="right-
arrow"></div></span';
      break;
case '232':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(233);"><div id="right-
arrow"></div></span';
      break;
case '233':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(205);"><div id="right-
arrow"></div></span';
      break;
case '234':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(219);"><div id="right-
arrow"></div></span';
      break;
case '235':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(208);"><div id="right-
arrow"></div></span';
      break;
case '250':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(251);"><div id="right-
arrow"></div></span';
      break;
case '251':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(252);"><div id="right-
arrow"></div></span';
      break;
case '252':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(253);"><div id="right-
arrow"></div></span';
      break;
case '253':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(254);"><div id="right-
arrow"></div></span';
      break;
case '254':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(255);"><div id="right-
arrow"></div></span';
      break;
case '255':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(256);"><div id="right-
arrow"></div></span';
      break;
case '256':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(263);"><div id="right-
arrow"></div></span';
      break;
case '257':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(143);"><div id="right-
arrow"></div></span';
      break;
case '258':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(259);"><div id="right-
arrow"></div></span';
      break;
case '259':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(260);"><div id="right-
arrow"></div></span';
      break;
case '260':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(161);"><div id="right-
arrow"></div></span';
      break;
case '261':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(129);"><div id="right-
arrow"></div></span';
      break;
case '262':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(319);"><div id="right-
arrow"></div></span';
      break;
case '263':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(264);"><div id="right-
arrow"></div></span';
      break;
case '264':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(265);"><div id="right-
arrow"></div></span';
      break;
case '265':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(266);"><div id="right-
arrow"></div></span';
      break;
case '266':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(2);"><div id="right-arrow"></div></span';
      break;
```

```
case '267':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(2);"><div id="right-arrow"></div></span';
      break;
case '300':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(303);"><div id="right-
arrow"></div></span';
      break;
case '301':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(59);"><div id="right-
arrow"></div></span';
      break;
case '302':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(14);"><div id="right-
arrow"></div></span';
      break;
case '303':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(320);"><div id="right-
arrow"></div></span';
      break;
case '304':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(305);"><div id="right-
arrow"></div></span';
      break;
case '305':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(306);"><div id="right-
arrow"></div></span';
      break;
case '306':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(307);"><div id="right-
arrow"></div></span';
      break;
case '307':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(171);"><div id="right-
arrow"></div></span';
      break;
case '308':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(309);"><div id="right-
arrow"></div></span';
      break;
case '309':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(310);"><div id="right-
arrow"></div></span';
      break;
case '310':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(1024);"><div id="right-
arrow"></div></span';
```

```
      break;
case '311':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(312);"><div id="right-
arrow"></div></span';
      break;
case '312':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(1026);"><div id="right-
arrow"></div></span';
      break;
case '313':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(315);"><div id="right-
arrow"></div></span';
      break;
case '314':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(172);"><div id="right-
arrow"></div></span';
      break;
case '315':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(316);"><div id="right-
arrow"></div></span';
      break;
case '316':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(317);"><div id="right-
arrow"></div></span';
      break;
case '317':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(319);"><div id="right-
arrow"></div></span';
      break;
case '318':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(319);"><div id="right-
arrow"></div></span';
      break;
case '319':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(4611);"><div id="right-
arrow"></div></span';
      break;
case '320':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(321);"><div id="right-
arrow"></div></span';
      break;
case '321':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(323);"><div id="right-
arrow"></div></span';
      break;
case '322':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(3);"><div id="right-arrow"></div></span';
      break;
case '323':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(1027);"><div id="right-
arrow"></div></span';
      break;

case '400':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(402);"><div id="right-
arrow"></div></span';
      break;
case '401':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(402);"><div id="right-
arrow"></div></span';
      break;
case '402':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(403);"><div id="right-
arrow"></div></span';
      break;
case '403':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(404);"><div id="right-
arrow"></div></span';
      break;
case '404':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(405);"><div id="right-
arrow"></div></span';
      break;
case '405':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(1029);"><div id="right-
arrow"></div></span';
      break;
case '406':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(315);"><div id="right-
arrow"></div></span';
      break;
case '408':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(4);"><div id="right-arrow"></div></span';
      break;
case '409':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(1030);"><div id="right-
arrow"></div></span';
      break;
case '410':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(411);"><div id="right-
arrow"></div></span';
      break;
```

```
case '411':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(414);"><div id="right-
arrow"></div></span';
      break;
case '412':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(415);"><div id="right-
arrow"></div></span';
      break;
case '413':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(212);"><div id="right-
arrow"></div></span';
      break;
case '414':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(412);"><div id="right-
arrow"></div></span';
      break;
case '415':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(419);"><div id="right-
arrow"></div></span';
      break;
case '416':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(417);"><div id="right-
arrow"></div></span';
      break;
case '417':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(419);"><div id="right-
arrow"></div></span';
      break;
case '418':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(262);"><div id="right-
arrow"></div></span';
      break;
case '419':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(173);"><div id="right-
arrow"></div></span';
      break;
case '1000':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(8);"><div id="right-arrow"></div></span';
      break;
case '1001':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(54);"><div id="right-
arrow"></div></span';
      break;
case '1002':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(55);"><div id="right-
arrow"></div></span';
```

```
      break;
case '1003':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(66);"><div id="right-
arrow"></div></span';
      break;
case '1004':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(1005);"><div id="right-
arrow"></div></span';
      break;
case '1005':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(67);"><div id="right-
arrow"></div></span';
      break;
case '1006':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(15);"><div id="right-
arrow"></div></span';
      break;
case '1007':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(17);"><div id="right-
arrow"></div></span';
      break;
case '1008':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(44);"><div id="right-
arrow"></div></span';
      break;
case '1009':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(4611);"><div id="right-
arrow"></div></span';
      break;
case '1010':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(4611);"><div id="right-
arrow"></div></span';
      break;
case '1011':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(114);"><div id="right-
arrow"></div></span';
      break;
case '1012':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(57);"><div id="right-
arrow"></div></span';
      break;
case '1013':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(5622);"><div id="right-
arrow"></div></span';
      break;
case '1014':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(5644);"><div id="right-
arrow"></div></span';
      break;
case '1015':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(4622);"><div id="right-
arrow"></div></span';
      break;
case '1016':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(60);"><div id="right-
arrow"></div></span';
      break;
case '1017':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(223);"><div id="right-
arrow"></div></span';
      break;
case '1019':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(225);"><div id="right-
arrow"></div></span';
      break;
case '1020':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(163);"><div id="right-
arrow"></div></span';
      break;
case '1021':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(306);"><div id="right-
arrow"></div></span';
      break;
case '1022':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(323);"><div id="right-
arrow"></div></span';
      break;
case '1023':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(1024);"><div id="right-
arrow"></div></span';
      break;
case '1024':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(311);"><div id="right-
arrow"></div></span';
      break;
case '1025':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(312);"><div id="right-
arrow"></div></span';
      break;
case '1026':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(314);"><div id="right-
arrow"></div></span';
```

```
      break;
case '1027':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(5633);"><div id="right-
arrow"></div></span';
      break;
case '1028':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(1029);"><div id="right-
arrow"></div></span';
      break;
case '1029':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(406);"><div id="right-
arrow"></div></span';
      break;
case '1030':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(1028);"><div id="right-
arrow"></div></span';
      break;
case '1031':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(1029);"><div id="right-
arrow"></div></span';
      break;
case '1032':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(179);"><div id="right-
arrow"></div></span';
      break;
case '1033':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(1031);"><div id="right-
arrow"></div></span';
      break;
case '1034':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomrightvert(1029);"><div id="right-
arrow"></div></span';
      break;
default:
      echo 'whoops';    }?>
```

```
/*
___bottomrighthori.php___
by Whitney Anne Trettien

This file loads the horizontal links for the bottom right quadrant.
*/

<?php $cOption = $_GET['o'];  switch($cOption) {
case '1':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(34);"><div id="up-arrow"></div></span>';
      break;
case '2':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(3);"><div id="up-arrow"></div></span';
      break;
case '3':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(46);"><div id="up-arrow"></div></span';
      break;
case '4':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(1000);"><div id="up-arrow"></div></span';
      break;
case '5':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(6);"><div id="up-arrow"></div></span>';
      break;
case '6':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(8);"><div id="up-arrow"></div></span';
      break;
case '7':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(8);"><div id="up-arrow"></div></span';
      break;
case '8':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(11);"><div id="up-arrow"></div></span';
      break;
case '9':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(10);"><div id="up-arrow"></div></span>';
      break;
case '10':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(34);"><div id="up-arrow"></div></span';
      break;
case '11':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(12);"><div id="up-arrow"></div></span';
      break;
case '12':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(13);"><div id="up-arrow"></div></span';
      break;
case '13':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(28);"><div id="up-arrow"></div></span>';
      break;
case '14':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(15);"><div id="up-arrow"></div></span';
      break;
case '15':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(34);"><div id="up-arrow"></div></span';
      break;
case '16':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(17);"><div id="up-arrow"></div></span';
      break;
case '17':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(21);"><div id="up-arrow"></div></span>';
      break;
case '18':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(19);"><div id="up-arrow"></div></span';
      break;
case '19':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(20);"><div id="up-arrow"></div></span';
      break;
case '20':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(21);"><div id="up-arrow"></div></span';
      break;
case '21':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(22);"><div id="up-arrow"></div></span>';
      break;
case '22':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(31);"><div id="up-arrow"></div></span';
      break;
case '23':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(24);"><div id="up-arrow"></div></span';
      break;
case '24':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(26);"><div id="up-arrow"></div></span';
      break;
case '25':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(1030);"><div id="up-arrow"></div></span>';
      break;
case '26':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(207);"><div id="up-arrow"></div></span';
      break;
case '27':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(53);"><div id="up-arrow"></div></span';
```

```
      break;
case '28':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(59);"><div id="up-arrow"></div></span';
      break;
case '29':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(30);"><div id="up-arrow"></div></span>';
      break;
case '30':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(53);"><div id="up-arrow"></div></span';
      break;
case '31':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(37);"><div id="up-arrow"></div></span';
      break;
case '32':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(11);"><div id="up-arrow"></div></span';
      break;
case '33':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(35);"><div id="up-arrow"></div></span>';
      break;
case '34':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(36);"><div id="up-arrow"></div></span';
      break;
case '35':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(39);"><div id="up-arrow"></div></span';
      break;
case '36':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(409);"><div id="up-arrow"></div></span';
      break;
case '37':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(46);"><div id="up-arrow"></div></span>';
      break;
case '38':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(110);"><div id="up-arrow"></div></span';
      break;
case '39':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(130);"><div id="up-arrow"></div></span';
      break;
case '40':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(41);"><div id="up-arrow"></div></span';
      break;
case '41':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(42);"><div id="up-arrow"></div></span>';
      break;
case '42':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(44);"><div id="up-arrow"></div></span';
      break;
case '43':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(44);"><div id="up-arrow"></div></span';
      break;
case '44':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(175);"><div id="up-arrow"></div></span';
      break;
case '45':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(13);"><div id="up-arrow"></div></span>';
      break;
case '46':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(48);"><div id="up-arrow"></div></span';
      break;
case '4611':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(1015);"><div id="up-arrow"></div></span';
      break;
case '4622':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(4633);"><div id="up-arrow"></div></span';
      break;
case '4633':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(1030);"><div id="up-arrow"></div></span';
      break;
case '4644':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(4655);"><div id="up-arrow"></div></span';
      break;
case '4655':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(175);"><div id="up-arrow"></div></span';
      break;
case '47':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(48);"><div id="up-arrow"></div></span';
      break;
case '48':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(1010);"><div id="up-arrow"></div></span';
      break;
case '49':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(57);"><div id="up-arrow"></div></span>';
      break;
case '53':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(1002);"><div id="up-arrow"></div></span';
      break;
case '54':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(56);"><div id="up-arrow"></div></span>';
```

```
      break;
case '55':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(5511);"><div id="up-arrow"></div></span';
      break;
case '5511':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(5644);"><div id="up-arrow"></div></span';
      break;
case '5522':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(5611);"><div id="up-arrow"></div></span';
      break;
case '56':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(58);"><div id="up-arrow"></div></span';
      break;
case '5611':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(1013);"><div id="up-arrow"></div></span';
      break;
case '5622':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(5633);"><div id="up-arrow"></div></span';
      break;
case '5633':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(57);"><div id="up-arrow"></div></span';
      break;
case '5644':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(5522);"><div id="up-arrow"></div></span';
      break;
case '57':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(56);"><div id="up-arrow"></div></span';
      break;
case '58':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(28);"><div id="up-arrow"></div></span>';
      break;
case '59':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(1016);"><div id="up-arrow"></div></span';
      break;
case '60':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(61);"><div id="up-arrow"></div></span';
      break;
case '61':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(43);"><div id="up-arrow"></div></span';
      break;
case '62':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(63);"><div id="up-arrow"></div></span>';
      break;
case '63':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(38);"><div id="up-arrow"></div></span';
      break;
case '64':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(65);"><div id="up-arrow"></div></span';
      break;
case '65':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(1004);"><div id="up-arrow"></div></span';
      break;
case '66':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(67);"><div id="up-arrow"></div></span>';
      break;
case '67':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(1006);"><div id="up-arrow"></div></span';
      break;
case '68':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(1);"><div id="up-arrow"></div></span';
      break;
case '100':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(401);"><div id="up-arrow"></div></span>';
      break;
case '101':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(130);"><div id="up-arrow"></div></span';
      break;
case '102':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(104);"><div id="up-arrow"></div></span';
      break;
case '103':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(104);"><div id="up-arrow"></div></span>';
      break;
case '104':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(105);"><div id="up-arrow"></div></span';
      break;
case '105':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(106);"><div id="up-arrow"></div></span';
      break;
case '106':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(107);"><div id="up-arrow"></div></span';
      break;
case '107':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(102);"><div id="up-arrow"></div></span';
      break;
case '108':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(121);"><div id="up-arrow"></div></span';
```

```
      break;
case '109':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(133);"><div id="up-arrow"></div></span';
      break;
case '110':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(115);"><div id="up-arrow"></div></span';
      break;
case '111':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(117);"><div id="up-arrow"></div></span';
      break;
case '112':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(41);"><div id="up-arrow"></div></span';
      break;
case '113':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(117);"><div id="up-arrow"></div></span';
      break;
case '114':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(113);"><div id="up-arrow"></div></span';
      break;
case '115':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(116);"><div id="up-arrow"></div></span';
      break;
case '116':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(126);"><div id="up-arrow"></div></span';
      break;
case '117':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(113);"><div id="up-arrow"></div></span';
      break;
case '118':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(32);"><div id="up-arrow"></div></span';
      break;
case '119':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(120);"><div id="up-arrow"></div></span';
      break;
case '11911':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(118);"><div id="up-arrow"></div></span';
      break;
case '120':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(12011);"><div id="up-arrow"></div></span';
      break;
case '12011':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(14811);"><div id="up-arrow"></div></span';
      break;
case '121':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(123);"><div id="up-arrow"></div></span';
      break;
case '122':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(1011);"><div id="up-arrow"></div></span';
      break;
case '123':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(125);"><div id="up-arrow"></div></span';
      break;
case '124':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(141);"><div id="up-arrow"></div></span';
      break;
case '126':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(127);"><div id="up-arrow"></div></span';
      break;
case '127':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(129);"><div id="up-arrow"></div></span';
      break;
case '128':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(130);"><div id="up-arrow"></div></span';
      break;
case '129':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(130);"><div id="up-arrow"></div></span';
      break;
case '12911':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(4);"><div id="up-arrow"></div></span';
      break;
case '130':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(134);"><div id="up-arrow"></div></span';
      break;
case '131':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(12011);"><div id="up-arrow"></div></span';
      break;
case '13111':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(13122);"><div id="up-arrow"></div></span';
      break;
case '13122':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(132);"><div id="up-arrow"></div></span';
      break;
case '132':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(133);"><div id="up-arrow"></div></span';
      break;
case '133':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(109);"><div id="up-arrow"></div></span';
```

```
      break;
case '134':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(135);"><div id="up-arrow"></div></span';
      break;
case '135':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(136);"><div id="up-arrow"></div></span';
      break;
case '136':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(126);"><div id="up-arrow"></div></span';
      break;
case '137':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(141);"><div id="up-arrow"></div></span';
      break;
case '138':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(139);"><div id="up-arrow"></div></span';
      break;
case '139':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(117);"><div id="up-arrow"></div></span';
      break;
case '140':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(158);"><div id="up-arrow"></div></span';
      break;
case '141':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(143);"><div id="up-arrow"></div></span';
      break;
case '142':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(57);"><div id="up-arrow"></div></span';
      break;
case '143':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(14811);"><div id="up-arrow"></div></span';
      break;
case '144':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(41);"><div id="up-arrow"></div></span';
      break;
case '146':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(147);"><div id="up-arrow"></div></span';
      break;
case '147':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(148);"><div id="up-arrow"></div></span';
      break;
case '14711':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(14722);"><div id="up-arrow"></div></span';
      break;
case '14722':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(14811);"><div id="up-arrow"></div></span';
      break;
case '148':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(1012);"><div id="up-arrow"></div></span';
      break;
case '14811':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(37);"><div id="up-arrow"></div></span';
      break;
case '14822':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(149);"><div id="up-arrow"></div></span';
      break;
case '14833':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(100);"><div id="up-arrow"></div></span';
      break;
case '149':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(150);"><div id="up-arrow"></div></span';
      break;
case '150':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(152);"><div id="up-arrow"></div></span';
      break;
case '151':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(153);"><div id="up-arrow"></div></span';
      break;
case '152':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(115);"><div id="up-arrow"></div></span';
      break;
case '153':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(404);"><div id="up-arrow"></div></span';
      break;
case '154':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(156);"><div id="up-arrow"></div></span';
      break;
case '155':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(159);"><div id="up-arrow"></div></span';
      break;
case '156':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(128);"><div id="up-arrow"></div></span';
      break;
case '157':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(171);"><div id="up-arrow"></div></span';
      break;
case '158':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(157);"><div id="up-arrow"></div></span';
```

```
      break;
case '159':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(1008);"><div id="up-arrow"></div></span';
      break;
case '160':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(156);"><div id="up-arrow"></div></span';
      break;
case '161':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(1020);"><div id="up-arrow"></div></span';
      break;
case '162':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(163);"><div id="up-arrow"></div></span';
      break;
case '163':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(219);"><div id="up-arrow"></div></span';
      break;
case '164':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(250);"><div id="up-arrow"></div></span';
      break;
case '170':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(172);"><div id="up-arrow"></div></span';
      break;
case '171':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(172);"><div id="up-arrow"></div></span';
      break;
case '172':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(180);"><div id="up-arrow"></div></span';
      break;
case '173':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(174);"><div id="up-arrow"></div></span';
      break;
case '174':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(177);"><div id="up-arrow"></div></span';
      break;

case '175':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(176);"><div id="up-arrow"></div></span';
      break;
case '176':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(178);"><div id="up-arrow"></div></span';
      break;
case '177':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(178);"><div id="up-arrow"></div></span';
      break;
```

```
case '178':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(1032);"><div id="up-arrow"></div></span';
      break;
case '179':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(180);"><div id="up-arrow"></div></span';
      break;
case '180':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(182);"><div id="up-arrow"></div></span';
      break;
case '181':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(182);"><div id="up-arrow"></div></span';
      break;
case '182':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(184);"><div id="up-arrow"></div></span';
      break;
case '183':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(184);"><div id="up-arrow"></div></span';
      break;
case '184':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(171);"><div id="up-arrow"></div></span';
      break;
case '185':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(134);"><div id="up-arrow"></div></span';
      break;
case '186':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(12011);"><div id="up-arrow"></div></span';
      break;
case '190':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(1029);"><div id="up-arrow"></div></span';
      break;
case '200':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(202);"><div id="up-arrow"></div></span';
      break;
case '201':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(203);"><div id="up-arrow"></div></span';
      break;
case '202':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(204);"><div id="up-arrow"></div></span';
      break;
case '203':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(232);"><div id="up-arrow"></div></span';
      break;
case '204':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(254);"><div id="up-arrow"></div></span';
      break;
case '205':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(207);"><div id="up-arrow"></div></span';
      break;
case '206':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(234);"><div id="up-arrow"></div></span';
      break;
case '207':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(208);"><div id="up-arrow"></div></span';
      break;
case '208':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(209);"><div id="up-arrow"></div></span';
      break;
case '209':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(211);"><div id="up-arrow"></div></span';
      break;
case '210':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(212);"><div id="up-arrow"></div></span';
      break;
case '211':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(213);"><div id="up-arrow"></div></span';
      break;
case '212':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(214);"><div id="up-arrow"></div></span';
      break;
case '213':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(216);"><div id="up-arrow"></div></span';
      break;
case '214':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(34);"><div id="up-arrow"></div></span';
      break;
case '215':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(219);"><div id="up-arrow"></div></span';
      break;
case '216':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(234);"><div id="up-arrow"></div></span';
      break;
case '219':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(220);"><div id="up-arrow"></div></span';
      break;
case '220':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(1017);"><div id="up-arrow"></div></span';
```

```
      break;
case '221':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(223);"><div id="up-arrow"></div></span';
      break;
case '222':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(1017);"><div id="up-arrow"></div></span';
      break;
case '223':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(225);"><div id="up-arrow"></div></span';
      break;
case '224':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(1019);"><div id="up-arrow"></div></span';
      break;
case '225':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(226);"><div id="up-arrow"></div></span';
      break;
case '226':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(228);"><div id="up-arrow"></div></span';
      break;
case '227':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(229);"><div id="up-arrow"></div></span';
      break;
case '228':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(300);"><div id="up-arrow"></div></span';
      break;
case '229':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(231);"><div id="up-arrow"></div></span';
      break;
case '230':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(19);"><div id="up-arrow"></div></span';
      break;
case '231':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(39);"><div id="up-arrow"></div></span';
      break;
case '232':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(205);"><div id="up-arrow"></div></span';
      break;
case '233':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(250);"><div id="up-arrow"></div></span';
      break;
case '234':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(303);"><div id="up-arrow"></div></span';
      break;
case '235':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(212);"><div id="up-arrow"></div></span';
      break;
case '250':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(253);"><div id="up-arrow"></div></span';
      break;
case '251':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(255);"><div id="up-arrow"></div></span';
      break;
case '252':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(255);"><div id="up-arrow"></div></span';
      break;
case '253':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(262);"><div id="up-arrow"></div></span';
      break;
case '254':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(4);"><div id="up-arrow"></div></span';
      break;
case '255':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(263);"><div id="up-arrow"></div></span';
      break;
case '256':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(257);"><div id="up-arrow"></div></span';
      break;
case '257':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(258);"><div id="up-arrow"></div></span';
      break;
case '258':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(205);"><div id="up-arrow"></div></span';
      break;
case '259':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(261);"><div id="up-arrow"></div></span';
      break;
case '260':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(14833);"><div id="up-arrow"></div></span';
      break;
case '261':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(230);"><div id="up-arrow"></div></span';
      break;
case '262':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(205);"><div id="up-arrow"></div></span';
      break;
case '263':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(265);"><div id="up-arrow"></div></span';
```

```
      break;
case '264':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(267);"><div id="up-arrow"></div></span';
      break;
case '265':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(267);"><div id="up-arrow"></div></span';
      break;
case '266':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(258);"><div id="up-arrow"></div></span';
      break;
case '267':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(143);"><div id="up-arrow"></div></span';
      break;
case '300':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(301);"><div id="up-arrow"></div></span';
      break;
case '301':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(302);"><div id="up-arrow"></div></span';
      break;
case '302':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(303);"><div id="up-arrow"></div></span';
      break;
case '303':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(304);"><div id="up-arrow"></div></span';
      break;
case '304':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(1021);"><div id="up-arrow"></div></span';
      break;
case '305':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(302);"><div id="up-arrow"></div></span';
      break;
case '306':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(1022);"><div id="up-arrow"></div></span';
      break;
case '307':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(308);"><div id="up-arrow"></div></span';
      break;
case '308':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(301);"><div id="up-arrow"></div></span';
      break;
case '309':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(1023);"><div id="up-arrow"></div></span';
      break;
case '310':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(311);"><div id="up-arrow"></div></span';
      break;
case '311':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(1025);"><div id="up-arrow"></div></span';
      break;
case '312':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(313);"><div id="up-arrow"></div></span';
      break;
case '313':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(322);"><div id="up-arrow"></div></span';
      break;
case '314':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(315);"><div id="up-arrow"></div></span';
      break;
case '315':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(317);"><div id="up-arrow"></div></span';
      break;
case '316':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(318);"><div id="up-arrow"></div></span';
      break;
case '317':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(301);"><div id="up-arrow"></div></span';
      break;
case '318':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(302);"><div id="up-arrow"></div></span';
      break;
case '319':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(1033);"><div id="up-arrow"></div></span';
      break;
case '320':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(33);"><div id="up-arrow"></div></span';
      break;
case '321':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(35);"><div id="up-arrow"></div></span';
      break;
case '322':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(100);"><div id="up-arrow"></div></span';
      break;
case '323':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(5611);"><div id="up-arrow"></div></span';
      break;
 case '400':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(401);"><div id="up-arrow"></div></span';
```

```
      break;
 case '401':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(1008);"><div id="up-arrow"></div></span';
      break;
 case '402':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(134);"><div id="up-arrow"></div></span';
      break;
 case '403':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(406);"><div id="up-arrow"></div></span';
      break;
 case '404':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(1028);"><div id="up-arrow"></div></span';
      break;
 case '405':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(406);"><div id="up-arrow"></div></span';
      break;
 case '406':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(408);"><div id="up-arrow"></div></span';
      break;
 case '408':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(409);"><div id="up-arrow"></div></span';
      break;
 case '409':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(410);"><div id="up-arrow"></div></span';
      break;
 case '410':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(403);"><div id="up-arrow"></div></span';
      break;
 case '411':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(215);"><div id="up-arrow"></div></span';
      break;
 case '412':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(413);"><div id="up-arrow"></div></span';
      break;
 case '413':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(59);"><div id="up-arrow"></div></span';
      break;
 case '414':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(1031);"><div id="up-arrow"></div></span';
      break;
 case '415':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(417);"><div id="up-arrow"></div></span';
      break;
 case '416':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(300);"><div id="up-arrow"></div></span';
      break;
 case '417':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(418);"><div id="up-arrow"></div></span';
      break;
 case '418':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(419);"><div id="up-arrow"></div></span';
      break;
 case '419':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(219);"><div id="up-arrow"></div></span';
      break;
case '1000':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(1028);"><div id="up-arrow"></div></span';
      break;
case '1001':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(1002);"><div id="up-arrow"></div></span';
      break;
case '1002':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(1001);"><div id="up-arrow"></div></span';
      break;
case '1003':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(1004);"><div id="up-arrow"></div></span';
      break;
case '1004':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(67);"><div id="up-arrow"></div></span';
      break;
case '1005':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(1006);"><div id="up-arrow"></div></span';
      break;
case '1006':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(68);"><div id="up-arrow"></div></span';
      break;
case '1007':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(18);"><div id="up-arrow"></div></span';
      break;
case '1008':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(49);"><div id="up-arrow"></div></span';
      break;
case '1009':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(49);"><div id="up-arrow"></div></span';
      break;
case '1010':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(49);"><div id="up-arrow"></div></span';
```

```
      break;
case '1011':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(119);"><div id="up-arrow"></div></span';
      break;
case '1012':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(56);"><div id="up-arrow"></div></span';
      break;
case '1013':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(58);"><div id="up-arrow"></div></span';
      break;
case '1014':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(5611);"><div id="up-arrow"></div></span';
      break;
case '1015':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(4655);"><div id="up-arrow"></div></span';
      break;
case '1016':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(61);"><div id="up-arrow"></div></span';
      break;
case '1017':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(221);"><div id="up-arrow"></div></span';
      break;
case '1019':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(226);"><div id="up-arrow"></div></span';
      break;
case '1020':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(164);"><div id="up-arrow"></div></span';
      break;
case '1021':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(307);"><div id="up-arrow"></div></span';
      break;
case '1022':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(322);"><div id="up-arrow"></div></span';
      break;
case '1023':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(310);"><div id="up-arrow"></div></span';
      break;
case '1024':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(1025);"><div id="up-arrow"></div></span';
      break;
case '1025':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(1026);"><div id="up-arrow"></div></span';
      break;
case '1026':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(322);"><div id="up-arrow"></div></span';
      break;
case '1027':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(129);"><div id="up-arrow"></div></span';
      break;
 case '1028':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(406);"><div id="up-arrow"></div></span';
      break;
 case '1029':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(408);"><div id="up-arrow"></div></span';
      break;
 case '1030':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(1000);"><div id="up-arrow"></div></span';
      break;
 case '1031':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(1026);"><div id="up-arrow"></div></span';
      break;
 case '1032':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(1020);"><div id="up-arrow"></div></span';
      break;
case '1033':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(409);"><div id="up-arrow"></div></span';
      break;
case '1034':
      echo '<span style="cursor: pointer"
onclick="loadContentToprighthori(263);"><div id="up-arrow"></div></span';
      break;
default:
      echo 'whoops';    }?>
```

```
/*
___bottomrightvert.php___
by Whitney Anne Trettien

This file loads the vertical links for the bottom right quadrant.
*/

<?php $cOption = $_GET['o'];  switch($cOption) {
case '1':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(2);"><div id="left-arrow"></div></span>';
      break;
case '2':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(48);"><div id="left-arrow"></div></span';
      break;
case '3':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(4);"><div id="left-arrow"></div></span';
      break;
case '4':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(5);"><div id="left-arrow"></div></span';
      break;
case '5':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(7);"><div id="left-arrow"></div></span>';
      break;
case '6':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(9);"><div id="left-arrow"></div></span';
      break;
case '7':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(9);"><div id="left-arrow"></div></span';
      break;
case '8':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(14);"><div id="left-arrow"></div></span';
      break;
case '9':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(177);"><div id="left-
arrow"></div></span>';
      break;
case '10':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(28);"><div id="left-arrow"></div></span';
      break;
case '11':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(13);"><div id="left-arrow"></div></span';
      break;
case '12':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(53);"><div id="left-arrow"></div></span';
      break;
case '13':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(33);"><div id="left-arrow"></div></span>';
      break;
case '14':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(18);"><div id="left-arrow"></div></span';
      break;
case '15':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(16);"><div id="left-arrow"></div></span';
      break;
case '16':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(1007);"><div id="left-
arrow"></div></span';
      break;
case '17':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(18);"><div id="left-arrow"></div></span>';
      break;
case '18':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(21);"><div id="left-arrow"></div></span';
      break;
case '19':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(400);"><div id="left-arrow"></div></span';
      break;
case '20':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(29);"><div id="left-arrow"></div></span';
      break;
case '21':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(46);"><div id="left-arrow"></div></span>';
      break;
case '22':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(23);"><div id="left-arrow"></div></span';
      break;
case '23':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(27);"><div id="left-arrow"></div></span';
      break;
case '24':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(25);"><div id="left-arrow"></div></span';
      break;
case '25':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(27);"><div id="left-arrow"></div></span>';
      break;
case '26':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(43);"><div id="left-arrow"></div></span';
      break;
case '27':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(29);"><div id="left-arrow"></div></span';
      break;
case '28':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(129);"><div id="left-arrow"></div></span';
      break;
case '29':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(48);"><div id="left-arrow"></div></span>';
      break;
case '30':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(31);"><div id="left-arrow"></div></span';
      break;
case '31':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(13);"><div id="left-arrow"></div></span';
      break;
case '32':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(59);"><div id="left-arrow"></div></span';
      break;
case '33':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(34);"><div id="left-arrow"></div></span>';
      break;
case '34':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(35);"><div id="left-arrow"></div></span';
      break;
case '35':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(40);"><div id="left-arrow"></div></span';
      break;
case '36':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(203);"><div id="left-arrow"></div></span';
      break;
case '37':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(39);"><div id="left-arrow"></div></span>';
      break;
case '38':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(37);"><div id="left-arrow"></div></span';
      break;
case '39':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(38);"><div id="left-arrow"></div></span';
      break;
case '40':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(4);"><div id="left-arrow"></div></span';
      break;
case '41':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(43);"><div id="left-arrow"></div></span>';
```

```
      break;
case '42':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(46);"><div id="left-arrow"></div></span';
      break;
case '43':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(1008);"><div id="left-
arrow"></div></span';
      break;
case '44':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(45);"><div id="left-arrow"></div></span';
      break;
case '45':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(59);"><div id="left-arrow"></div></span>';
      break;
case '46':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(4611);"><div id="left-
arrow"></div></span';
      break;
case '4611':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(4622);"><div id="left-
arrow"></div></span';
      break;
case '4622':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(4655);"><div id="left-
arrow"></div></span';
      break;
case '4633':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(4644);"><div id="left-
arrow"></div></span';
      break;
case '4644':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(100);"><div id="left-arrow"></div></span';
      break;
case '4655':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(38);"><div id="left-arrow"></div></span';
      break;
case '47':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(49);"><div id="left-arrow"></div></span';
      break;
case '48':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(1009);"><div id="left-
arrow"></div></span';
      break;
case '49':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(119);"><div id="left-
arrow"></div></span>';
      break;
case '53':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(1001);"><div id="left-
arrow"></div></span';
      break;
case '54':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(57);"><div id="left-arrow"></div></span>';
      break;
case '55':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(5522);"><div id="left-
arrow"></div></span';
      break;
case '5511':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(5611);"><div id="left-
arrow"></div></span';
      break;
case '5522':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(1014);"><div id="left-
arrow"></div></span';
      break;
case '56':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(39);"><div id="left-arrow"></div></span';
      break;
case '5611':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(5622);"><div id="left-
arrow"></div></span';
      break;
case '5622':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(56);"><div id="left-arrow"></div></span';
      break;
case '5633':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(1012);"><div id="left-
arrow"></div></span';
      break;
case '5644':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(57);"><div id="left-arrow"></div></span';
      break;
case '57':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(3);"><div id="left-arrow"></div></span';
      break;
case '58':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(59);"><div id="left-arrow"></div></span>';
      break;
```

```
case '59':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(60);"><div id="left-arrow"></div></span';
      break;
case '60':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(43);"><div id="left-arrow"></div></span';
      break;
case '61':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(62);"><div id="left-arrow"></div></span';
      break;
case '62':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(64);"><div id="left-arrow"></div></span>';
      break;
case '63':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(64);"><div id="left-arrow"></div></span';
      break;
case '64':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(1003);"><div id="left-
arrow"></div></span';
      break;
case '65':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(66);"><div id="left-arrow"></div></span';
      break;
case '66':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(1005);"><div id="left-
arrow"></div></span>';
      break;
case '67':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(68);"><div id="left-arrow"></div></span';
      break;
case '68':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(18);"><div id="left-arrow"></div></span';
      break;
case '100':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(101);"><div id="left-
arrow"></div></span>';
      break;
case '101':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(1002);"><div id="left-
arrow"></div></span';
      break;
case '102':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(103);"><div id="left-arrow"></div></span';
      break;
case '103':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(105);"><div id="left-
arrow"></div></span>';
      break;
case '104':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(103);"><div id="left-arrow"></div></span';
      break;
case '105':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(107);"><div id="left-arrow"></div></span';
      break;
case '106':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(109);"><div id="left-arrow"></div></span';
      break;
case '107':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(13111);"><div id="left-
arrow"></div></span';
      break;
case '108':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(111);"><div id="left-arrow"></div></span';
      break;
case '109':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(42);"><div id="left-arrow"></div></span';
      break;
case '110':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(108);"><div id="left-arrow"></div></span';
      break;
case '111':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(112);"><div id="left-arrow"></div></span';
      break;
case '112':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(100);"><div id="left-arrow"></div></span';
      break;
case '113':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(140);"><div id="left-arrow"></div></span';
      break;
case '114':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(122);"><div id="left-arrow"></div></span';
      break;
case '115':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(111);"><div id="left-arrow"></div></span';
      break;
case '116':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(118);"><div id="left-arrow"></div></span';
      break;
case '117':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(134);"><div id="left-arrow"></div></span';
      break;
case '118':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(102);"><div id="left-arrow"></div></span';
      break;
case '119':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(11911);"><div id="left-
arrow"></div></span';
      break;
case '11911':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(120);"><div id="left-arrow"></div></span';
      break;
case '120':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(126);"><div id="left-arrow"></div></span';
      break;
case '12011':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(4);"><div id="left-arrow"></div></span';
      break;
case '121':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(59);"><div id="left-arrow"></div></span';
      break;
case '122':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(102);"><div id="left-arrow"></div></span';
      break;
case '123':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(124);"><div id="left-arrow"></div></span';
      break;
case '124':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(403);"><div id="left-arrow"></div></span';
      break;
case '126':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(128);"><div id="left-arrow"></div></span';
      break;
case '127':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(128);"><div id="left-arrow"></div></span';
      break;
case '128':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(129);"><div id="left-arrow"></div></span';
      break;
case '129':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(100);"><div id="left-arrow"></div></span';
      break;
case '12911':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(150);"><div id="left-arrow"></div></span';
      break;
case '130':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(131);"><div id="left-arrow"></div></span';
      break;
case '131':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(13111);"><div id="left-
arrow"></div></span';
      break;
case '13111':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(132);"><div id="left-arrow"></div></span';
      break;
case '13122':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(148);"><div id="left-arrow"></div></span';
      break;
case '132':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(14711);"><div id="left-
arrow"></div></span';
      break;
case '133':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(14822);"><div id="left-
arrow"></div></span';
      break;
case '134':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(138);"><div id="left-arrow"></div></span';
      break;
case '135':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(137);"><div id="left-arrow"></div></span';
      break;
case '136':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(141);"><div id="left-arrow"></div></span';
      break;
case '137':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(130);"><div id="left-arrow"></div></span';
      break;
case '138':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(140);"><div id="left-arrow"></div></span';
      break;
case '139':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(43);"><div id="left-arrow"></div></span';
      break;
case '140':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(111);"><div id="left-arrow"></div></span';
      break;
```

```
case '141':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(142);"><div id="left-arrow"></div></span';
      break;
case '142':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(143);"><div id="left-arrow"></div></span';
      break;
case '143':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(144);"><div id="left-arrow"></div></span';
      break;
case '144':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(146);"><div id="left-arrow"></div></span';
      break;
case '146':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(13122);"><div id="left-
arrow"></div></span';
      break;
case '147':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(14711);"><div id="left-
arrow"></div></span';
      break;
case '14711':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(149);"><div id="left-arrow"></div></span';
      break;
case '14722':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(148);"><div id="left-arrow"></div></span';
      break;
case '148':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(49);"><div id="left-arrow"></div></span';
      break;
case '14811':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(13);"><div id="left-arrow"></div></span';
      break;
case '14822':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(113);"><div id="left-arrow"></div></span';
      break;
case '14833':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(1000);"><div id="left-
arrow"></div></span';
      break;
case '149':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(154);"><div id="left-arrow"></div></span';
      break;
case '150':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(151);"><div id="left-arrow"></div></span';
```

```
      break;
case '151':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(152);"><div id="left-arrow"></div></span';
      break;
case '152':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(153);"><div id="left-arrow"></div></span';
      break;
case '153':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(4);"><div id="left-arrow"></div></span';
      break;
case '154':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(155);"><div id="left-arrow"></div></span';
      break;
case '155':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(158);"><div id="left-arrow"></div></span';
      break;
case '156':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(157);"><div id="left-arrow"></div></span';
      break;
case '157':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(129);"><div id="left-arrow"></div></span';
      break;
case '158':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(140);"><div id="left-arrow"></div></span';
      break;
case '159':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(156);"><div id="left-arrow"></div></span';
      break;
case '160':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(1034);"><div id="left-
arrow"></div></span';
      break;
case '161':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(162);"><div id="left-arrow"></div></span';
      break;
case '162':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(1020);"><div id="left-
arrow"></div></span';
      break;
case '163':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(164);"><div id="left-arrow"></div></span';
      break;
case '164':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(130);"><div id="left-arrow"></div></span';
```

```
      break;
case '170':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(171);"><div id="left-arrow"></div></span';
      break;
case '171':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(53);"><div id="left-arrow"></div></span';
      break;
case '172':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(173);"><div id="left-arrow"></div></span';
      break;
case '173':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(177);"><div id="left-arrow"></div></span';
      break;
case '174':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(176);"><div id="left-arrow"></div></span';
      break;
case '175':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(178);"><div id="left-arrow"></div></span';
      break;
case '176':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(14811);"><div id="left-
arrow"></div></span';
      break;
case '177':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(176);"><div id="left-arrow"></div></span';
      break;
case '178':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(179);"><div id="left-arrow"></div></span';
      break;
case '179':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(403);"><div id="left-arrow"></div></span';
      break;
case '180':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(181);"><div id="left-arrow"></div></span';
      break;
case '181':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(184);"><div id="left-arrow"></div></span';
      break;
case '182':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(183);"><div id="left-arrow"></div></span';
      break;
case '183':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(185);"><div id="left-arrow"></div></span';
      break;
```

```
case '184':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(185);"><div id="left-arrow"></div></span';
      break;
case '185':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(403);"><div id="left-arrow"></div></span';
      break;
case '186':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(171);"><div id="left-arrow"></div></span';
      break;
case '190':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(1000);"><div id="left-
arrow"></div></span';
      break;
case '200':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(201);"><div id="left-arrow"></div></span';
      break;
case '201':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(202);"><div id="left-arrow"></div></span';
      break;
case '202':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(203);"><div id="left-arrow"></div></span';
      break;
case '203':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(204);"><div id="left-arrow"></div></span';
      break;
case '204':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(205);"><div id="left-arrow"></div></span';
      break;
case '205':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(206);"><div id="left-arrow"></div></span';
      break;
case '206':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(207);"><div id="left-arrow"></div></span';
      break;
case '207':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(4);"><div id="left-arrow"></div></span';
      break;
case '208':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(210);"><div id="left-arrow"></div></span';
      break;
case '209':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(173);"><div id="left-arrow"></div></span';
      break;
case '210':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(211);"><div id="left-arrow"></div></span';
      break;
case '211':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(212);"><div id="left-arrow"></div></span';
      break;
case '212':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(213);"><div id="left-arrow"></div></span';
      break;
case '213':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(215);"><div id="left-arrow"></div></span';
      break;
case '214':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(303);"><div id="left-arrow"></div></span';
      break;
case '215':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(216);"><div id="left-arrow"></div></span';
      break;
case '216':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(222);"><div id="left-arrow"></div></span';
      break;
case '219':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(214);"><div id="left-arrow"></div></span';
      break;
case '220':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(221);"><div id="left-arrow"></div></span';
      break;
case '221':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(222);"><div id="left-arrow"></div></span';
      break;
case '222':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(223);"><div id="left-arrow"></div></span';
      break;
case '223':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(224);"><div id="left-arrow"></div></span';
      break;
case '224':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(225);"><div id="left-arrow"></div></span';
      break;
case '225':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(1019);"><div id="left-
arrow"></div></span';
      break;
case '226':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(227);"><div id="left-arrow"></div></span';
      break;
case '227':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(228);"><div id="left-arrow"></div></span';
      break;
case '228':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(229);"><div id="left-arrow"></div></span';
      break;
case '229':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(230);"><div id="left-arrow"></div></span';
      break;
case '230':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(231);"><div id="left-arrow"></div></span';
      break;
case '231':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(250);"><div id="left-arrow"></div></span';
      break;
case '232':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(233);"><div id="left-arrow"></div></span';
      break;
case '233':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(205);"><div id="left-arrow"></div></span';
      break;
case '234':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(219);"><div id="left-arrow"></div></span';
      break;
case '235':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(208);"><div id="left-arrow"></div></span';
      break;
case '250':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(251);"><div id="left-arrow"></div></span';
      break;
case '251':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(252);"><div id="left-arrow"></div></span';
      break;
case '252':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(253);"><div id="left-arrow"></div></span';
      break;
case '253':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(254);"><div id="left-arrow"></div></span';
      break;
case '254':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(255);"><div id="left-arrow"></div></span';
```

```
      break;
case '255':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(256);"><div id="left-arrow"></div></span';
      break;
case '256':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(263);"><div id="left-arrow"></div></span';
      break;
case '257':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(143);"><div id="left-arrow"></div></span';
      break;
case '258':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(259);"><div id="left-arrow"></div></span';
      break;
case '259':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(260);"><div id="left-arrow"></div></span';
      break;
case '260':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(161);"><div id="left-arrow"></div></span';
      break;
case '261':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(129);"><div id="left-arrow"></div></span';
      break;
case '262':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(319);"><div id="left-arrow"></div></span';
      break;
case '263':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(264);"><div id="left-arrow"></div></span';
      break;
case '264':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(265);"><div id="left-arrow"></div></span';
      break;
case '265':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(266);"><div id="left-arrow"></div></span';
      break;
case '266':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(2);"><div id="left-arrow"></div></span';
      break;
case '267':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(2);"><div id="left-arrow"></div></span';
      break;
case '300':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(303);"><div id="left-arrow"></div></span';
      break;
case '301':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(59);"><div id="left-arrow"></div></span';
      break;
case '302':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(14);"><div id="left-arrow"></div></span';
      break;
case '303':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(320);"><div id="left-arrow"></div></span';
      break;
case '304':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(305);"><div id="left-arrow"></div></span';
      break;
case '305':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(306);"><div id="left-arrow"></div></span';
      break;
case '306':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(307);"><div id="left-arrow"></div></span';
      break;
case '307':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(171);"><div id="left-arrow"></div></span';
      break;
case '308':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(309);"><div id="left-arrow"></div></span';
      break;
case '309':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(310);"><div id="left-arrow"></div></span';
      break;
case '310':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(1024);"><div id="left-
arrow"></div></span';
      break;
case '311':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(312);"><div id="left-arrow"></div></span';
      break;
case '312':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(1026);"><div id="left-
arrow"></div></span';
      break;
case '313':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(315);"><div id="left-arrow"></div></span';
      break;
case '314':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(172);"><div id="left-arrow"></div></span';
      break;
case '315':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(316);"><div id="left-arrow"></div></span';
      break;
case '316':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(317);"><div id="left-arrow"></div></span';
      break;
case '317':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(319);"><div id="left-arrow"></div></span';
      break;
case '318':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(319);"><div id="left-arrow"></div></span';
      break;
case '319':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(4611);"><div id="left-
arrow"></div></span';
      break;
case '320':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(321);"><div id="left-arrow"></div></span';
      break;
case '321':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(323);"><div id="left-arrow"></div></span';
      break;
case '322':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(3);"><div id="left-arrow"></div></span';
      break;
case '323':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(1027);"><div id="left-
arrow"></div></span';
      break;

case '400':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(402);"><div id="left-arrow"></div></span';
      break;
case '401':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(402);"><div id="left-arrow"></div></span';
      break;
case '402':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(403);"><div id="left-arrow"></div></span';
      break;
case '403':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(404);"><div id="left-arrow"></div></span';
      break;
case '404':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(405);"><div id="left-arrow"></div></span';
      break;
```

```
case '405':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(1029);"><div id="left-
arrow"></div></span';
      break;
case '406':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(315);"><div id="left-arrow"></div></span';
      break;
case '408':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(4);"><div id="left-arrow"></div></span';
      break;
case '409':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(1030);"><div id="left-
arrow"></div></span';
      break;
case '410':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(411);"><div id="left-arrow"></div></span';
      break;
case '411':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(414);"><div id="left-arrow"></div></span';
      break;
case '412':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(415);"><div id="left-arrow"></div></span';
      break;
case '413':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(212);"><div id="left-arrow"></div></span';
      break;
case '414':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(412);"><div id="left-arrow"></div></span';
      break;
case '415':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(419);"><div id="left-arrow"></div></span';
      break;
case '416':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(417);"><div id="left-arrow"></div></span';
      break;
case '417':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(419);"><div id="left-arrow"></div></span';
      break;
case '418':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(262);"><div id="left-arrow"></div></span';
      break;
case '419':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(173);"><div id="left-arrow"></div></span';
      break;
```

```
case '1000':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(8);"><div id="left-arrow"></div></span';
      break;
case '1001':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(54);"><div id="left-arrow"></div></span';
      break;
case '1002':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(55);"><div id="left-arrow"></div></span';
      break;
case '1003':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(66);"><div id="left-arrow"></div></span';
      break;
case '1004':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(1005);"><div id="left-
arrow"></div></span';
      break;
case '1005':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(67);"><div id="left-arrow"></div></span';
      break;
case '1006':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(15);"><div id="left-arrow"></div></span';
      break;
case '1007':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(17);"><div id="left-arrow"></div></span';
      break;
case '1008':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(44);"><div id="left-arrow"></div></span';
      break;
case '1009':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(4611);"><div id="left-
arrow"></div></span';
      break;
case '1010':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(4611);"><div id="left-
arrow"></div></span';
      break;
case '1011':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(114);"><div id="left-arrow"></div></span';
      break;
case '1012':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(57);"><div id="left-arrow"></div></span';
      break;
case '1013':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(5622);"><div id="left-
arrow"></div></span';
      break;
case '1014':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(5644);"><div id="left-
arrow"></div></span';
      break;
case '1015':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(4622);"><div id="left-
arrow"></div></span';
      break;
case '1016':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(60);"><div id="left-arrow"></div></span';
      break;
case '1017':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(223);"><div id="left-arrow"></div></span';
      break;
case '1019':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(225);"><div id="left-arrow"></div></span';
      break;
case '1020':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(163);"><div id="left-arrow"></div></span';
      break;
case '1021':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(306);"><div id="left-arrow"></div></span';
      break;
case '1022':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(323);"><div id="left-arrow"></div></span';
      break;
case '1023':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(1024);"><div id="left-
arrow"></div></span';
      break;
case '1024':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(311);"><div id="left-arrow"></div></span';
      break;
case '1025':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(312);"><div id="left-arrow"></div></span';
      break;
case '1026':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(314);"><div id="left-arrow"></div></span';
      break;
case '1027':
```

```
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(5633);"><div id="left-
arrow"></div></span';
      break;
case '1028':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(1029);"><div id="left-
arrow"></div></span';
      break;
case '1029':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(406);"><div id="left-arrow"></div></span';
      break;
case '1030':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(1028);"><div id="left-
arrow"></div></span';
      break;
case '1031':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(1029);"><div id="left-
arrow"></div></span';
      break;
case '1032':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(179);"><div id="left-arrow"></div></span';
      break;
case '1033':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(1031);"><div id="left-
arrow"></div></span';
      break;
case '1034':
      echo '<span style="cursor: pointer"
onclick="loadContentBottomleftvert(1029);"><div id="left-
arrow"></div></span';
      break;
default:
      echo 'whoops';    }?>
```

```
/*
___textwithdots.php___
by Whitney Anne Trettien

This file loads whatever has been called from text.php into the "Your Text"
section of the page. The only difference between this file and "text.php" is
the colored square at the beginning of each text chunk.
*/


<?php $cOption = $_GET['o'];
switch($cOption) {
case '1':
      echo '<p><span class="volvelles">__</span><p>nbsp;Around 1650, Georg
Philipp Harsd&ouml;rffer devised an ingenious ballet.  It&#39;s simple:
first, give each dancer a board inscribed with a letter of the alphabet; then
watch as new words or phrases  emerge from dance. The very movement of the
dancer&#39;s bodies will act as a combinatory mechanism from which language
springs (<a href="bibliography.html#westerhoff"
target="_blank">Westerhoff<span>Westerhoff, Jan C. &quot;Poeta Calculans:
Harsd&ouml;rffer, Leibniz, and the <i>Mathesis Universalis</i>.&quot;
<i>Journal of the History of Ideas</i> 60.3 (1999): 449-67.</span></a>
465).</p><p>There is no evidence that Harsd&ouml;rffer ever produced such a
ballet.  Perhaps the first modern conceptual poet, Harsd&ouml;rffer&#39;s
genius lies in his ability to construct reality through  language &mdash;
that is, to use language not as a mirror for the world, but as its
fundamental building blocks.</p>';
      break;
case '2':
      echo '<p><span class="volvelles">__</span><p>nbsp;Harsd&ouml;rffer used
pieces of wood to make anagrams (<a href="bibliography.html#schwenter"
target="_blank"><i>Delitiae</i><span>Harsd&ouml;rffer, Georg Philipp and
Daniel Schwenter. <i>Delici&aelig; Physico-Mathematic&aelig;, Oder, Mathemat.
Und Philosophische Erquickstunden...</i> N&uuml;rnberg: in Verlegung Jeremiae
D&uuml;mlers, 1651.</span></a> II.514), designed letter-dice to teach
children to build word combinations (<a
href="bibliography.html#Harsd&ouml;rffer-trichter"
target="_blank"><i>Trichter</i><span>Harsd&ouml;rffer, Georg Philipp.
<i>Poetischer Trichter; Die Teutsche Dicht- Und Reimkunst, Ohne Behuf Der
Lateinischen Sprache, in VI. Stunden Einzugiessen</i>. Hildesheim, New York:
G. Olms, 1971.</span></a> II.18; see <a href="bibliography.html#westerhoff"
target="_blank">Westerhoff<span>Westerhoff, Jan C. &quot;Poeta Calculans:
Harsd&ouml;rffer, Leibniz, and the <i>Mathesis Universalis</i>.&quot;
<i>Journal of the History of Ideas</i> 60.3 (1999): 449-67.</span></a> 464),
and  assigned numbers to letters to unlock a poem&#39;s hidden values (<a
href="bibliography.html#Harsd&ouml;rffer-trichter"
target="_blank"><i>Trichter</i><span>Harsd&ouml;rffer, Georg Philipp.
<i>Poetischer Trichter; Die Teutsche Dicht- Und Reimkunst, Ohne Behuf Der
Lateinischen Sprache, in VI. Stunden Einzugiessen</i>. Hildesheim, New York:
G. Olms, 1971.</span></a> II.26-9), earning him the title Der  Spielende, or
&#34;the Player,&#34; in the <i>Fruchtbringende Gesellschaft</i>. Each of
these games uses language not  as an abstraction, the purely rational product
of the mind, but as quite literally a material object to be manipulated and
moved, cut-up and combined.</p>';
      break;
case '3':
      echo '<blockquote><p class="blockquote"><span
class="volvelles">__</span><p>nbsp;In this simple mechanization we see how
```

```
the ideas of the  combinatorial nature of the world on the one hand and
language on the other hand meet: letters are inscribed onto material
elements, so that a permutation of these out of itself generates an anagram
without requiring any ingenuity. For the  baroque poet this structural
isomorphism between language and matter explains why this method of anagram
production  works, while this fact itself serves as a further manifestation
of the very same isomorphism. Because language and matter  both follow
combinatorial principles, it is possible to mechanize the former by
inscribing it onto the latter. In the  &#34;language machine&#34; the
combinatorial features of language and the world which are generally distinct
are unified  into a single artifact. (<a href="bibliography.html#westerhoff"
target="_blank">Westerhoff<span>Westerhoff, Jan C. &quot;Poeta Calculans:
Harsd&ouml;rffer, Leibniz, and the <i>Mathesis Universalis</i>.&quot;
<i>Journal of the History of Ideas</i> 60.3 (1999): 449-67.</span></a> 464-
5)</p></blockquote>';
      break;
case '4':
      echo '<p><span class="volvelles">__</span><p>nbsp;The
<i>F&uuml;nffacher Denckring der Teutschen Sprache</i>, or the Five-fold
Thought-ring of the German Language, is a database of German words composed
of five predicate variables:  prefixes (forty-eight values), initial letters
or diphthongs (fifty values), medial letters (twelve values), final letters
of  diphthongs (120 values) and suffixes (twenty-four values). Instead of
using a table structure, however, each variable is  inscribed along the edge
of a disc nested within each of the other discs, forming a simple combinatory
mechanism that can generate any information stored in its circular,
relational database.</p>';
      break;
case '5':
      echo '<blockquote><p class="blockquote"><span
class="volvelles">__</span><p>nbsp;<a href="#">Will ich nun alle
Stammw&ouml;rter ordenlich finden / so fange ich bey  dem A de&szlig; zweyten
Ringes an / und drehe darzu das kleine a de&szlig; dritten Ringes:<span>I
want now to find all stem-words in an orderly manner, so I begin with the A
on the second Ring / and counterrotate the small a on the third
ring:</span></a><a href="#"> dann suche ich den vierten  Ring Aab / Aabb /
Aabd / &amp;c. blinde oder deutunglose W&ouml;rter / bi&szlig; auf das ch /
Aach / Aquisgranum, eine  benamte Stadt in Niderland / Aal / eines Fisches
und eines Schusters Werckzeug Namen / Aas (cadaver)&c.<span>then I seek the
fourth ring Aab / Aabb / Aabd / &c. / blind and meanlingless words / up to
the ch / Aach / Aquisgranum, named a city in the Netherlands / Aal / the name
of a fish and a tool of the cobbler / Aas (cadaver)&c.</span></a>  (<a
href="bibliography.html#schwenter"
target="_blank"><i>Delitiae</i><span>Harsd&ouml;rffer, Georg Philipp and
Daniel Schwenter. <i>Delici&aelig; Physico-Mathematic&aelig;, Oder, Mathemat.
Und Philosophische Erquickstunden...</i> N&uuml;rnberg: in Verlegung Jeremiae
D&uuml;mlers, 1651.</span></a> II. 516-7)</p></blockquote>';
      break;
case '6':
      echo '<p><span class="volvelles">__</span><p>nbsp;By spinning the discs
of the Denckring, the user can generate up to 97,209,600  words &mdash; 300
times as many as are in the most complete printed dictionary ever produced,
the Oxford English  Dictionary.</p>';
      break;
case '7':
      echo '<p><span class="volvelles">__</span><p>nbsp;Harsd&ouml;rffer
claims that <a href="#">&#34;wird sich verhoffentlich kein Wort in unsrer
gantzen Sprache finden / welches nicht auf diesem Ring weisen seyn
```

```
solte&#34;<span>hopefully [there is] no word found in our entire language /
which cannot be be shown on this ring</span></a> (<a
href="bibliography.html#schwenter"
target="_blank"><i>Delitiae</i><span>Harsd&ouml;rffer, Georg Philipp and
Daniel Schwenter. <i>Delici&aelig; Physico-Mathematic&aelig;, Oder, Mathemat.
Und Philosophische Erquickstunden...</i> N&uuml;rnberg: in Verlegung Jeremiae
D&uuml;mlers, 1651.</span></a> II.519), although of  course in reality the
rings are incomplete (<a href="bibliography.html#zeller">Zeller<span>Zeller,
Rosmarie. <i>Spiel Und Konversation Im Barock; Untersuchungen Zu
Harsd&ouml;rffers Gespr&auml;chsspielen.</i> New York: de Gruyter,
1974.</span></a> 167). In addition to representing <a href="#">&#34;die
gantze Teutsche Sprache auf  einem Bl&auml;tlein&#34;<span>the complete
German language on a small page</span></a> (<a
href="bibliography.html#schwenter"
target="_blank"><i>Delitiae</i><span>Harsd&ouml;rffer, Georg Philipp and
Daniel Schwenter. <i>Delici&aelig; Physico-Mathematic&aelig;, Oder, Mathemat.
Und Philosophische Erquickstunden...</i> N&uuml;rnberg: in Verlegung Jeremiae
D&uuml;mlers, 1651.</span></a> II.516), the Denckring also forms new words
that do not yet exist in German but,  because they are formed by combining
correct radicals and letters, could exist as legitimate words.</p>';
      break;
case '8':
      echo '<p><span class="volvelles">__</span><p>nbsp;As Harsd&ouml;rffer
explains, explicitly tying his device to the linguistic  theories of
Schottel,</p><blockquote><p class="blockquote"><a href="#">Ist also dieses
eine unfehlbare Richtigkeit / ein vollst&auml;ndiges Teutsches  Wortbuch zu
verfassen / und beharren wir in der Meinung / da&szlig; alle solche zusammen
gesetzte W&ouml;rter / welche ihre  Deutung w&uuml;rken f&uuml;r gut Teutsch
zul&auml;ssig / sonderlich in den Gedichten<span>It is an infallible accuracy
/ to write a dictionary of all German words / and we insist on the opinion /
that all such words combined together / which are denoted [to be] of good and
proper German / are particularly [useful] in poems</span></a><a href="#"> /
ob sie gleich sonsten nicht  gebr&auml;uchlich / wie hiervon zu lesen der umb
unsere Sprache wolverdiente Herr Schottelius in seiner Einleitung und in
seinen Lobreden der Sprachk&uuml;nst vorgef&uuml;get.<span> / even if they
are not very common / as hereof Mr. Schottel orders our language to be read
in his introduction and chapters on the linguistic arts.</span></a> (<a
href="bibliography.html#schwenter"
target="_blank"><i>Delitiae</i><span>Harsd&ouml;rffer, Georg Philipp and
Daniel Schwenter. <i>Delici&aelig; Physico-Mathematic&aelig;, Oder, Mathemat.
Und Philosophische Erquickstunden...</i> N&uuml;rnberg: in Verlegung Jeremiae
D&uuml;mlers, 1651.</span></a> II.518).</p></blockquote>';
      break;
case '9':
      echo '<p><span class="volvelles">__</span><p>nbsp;Because the Denckring
materializes proper root-words and mechanizes the  proper formula for
combining them, any output it produces must also be proper German. In other
words, in this view  language is fundamentally a material and mechanical
phenomenon: material because it is rooted in monosyllabic  stem-words that
are themselves rooted in &#34;die Teutschen letteren oder Buchstabe&#34; (<a
href="bibliography.html#schottel-haubtsprache" target="_blank">Schottel
<i>HaubtSprache</i><span>Schottelius, Justus Georg. <i>Ausfuehrliche Arbeit
Von Der Teutschen Haubt Sprache, Worin Enthalten Gemelter Dieser Haubt
Sprache Uhrankunft, Uhraltertuhm, Reinlichkeit, Eigenschaft, Vermoegen,
Unvergleichlichkeit, Grundrichtigkeit, Zumahl Die Sprach Kunst Und Vers Kunst
Teutsch Und Guten Theils Lateinisch Voellig Mit Eingebracht, Wie Nicht
Weniger Die Verdoppelung, Ableitung, Die Einleitung, Nahmwoerter, Authores
Vom Teutschen Wesen Und Teutscher Sprache, Von Der Verteutschung, Item Die
```

```
Stammwoerter Der Teutschen Sprache Samt Der Erklaerung Und Derogleichen Viel
Merkwuerdige Sachen ... Ausgefertiget Von Justo-Georgio Schottelio ..</i>
Braunschweig: Gedrukt und verlegt durch C. F. Zilligern, 1663.</span></a>
61.31), and  mechanical because meaning is not computed rationally but
through an absolute and abstracted algorithm. As Newman  points out (writing
more generally of Harsd&ouml;rffer&#39;s linguistics), far from secularizing
the vernacular, its physical  instantiation &#34;in textual form adds to its
authority and testifies to the reality of divinity present in the German
tongue&#34; (<a href="bibliography.html#newman"
target="_blank">Newman<span>Newman, Jane O. <i>Pastoral Conventions: Poetry,
Language, and Thought in Seventeenth-Century  Nuremberg.</i> Baltimore: Johns
Hopkins University Press, 1990.</span></a> 97), bringing the reader closer to
an originary <i>lingua adamica</i> that is divinely, not merely
prescriptively, proper.</p>';
      break;
case '10':
      echo '<blockquote><p class="blockquote"><span
class="volvelles">__</span><p>nbsp;It does not seem to bother
Harsd&ouml;rffer that it is a book, a  man-made, printed text, that he relies
upon for assurances about the divine capacities of the written word &mdash;
precisely the opposite. The availability of this knowledge in textual form
adds to its authority and testifies to the reality of  divinity present in
the German tongue. (<a href="bibliography.html#newman"
target="_blank">Newman<span>Newman, Jane O. <i>Pastoral Conventions: Poetry,
Language, and Thought in Seventeenth-Century  Nuremberg.</i> Baltimore: Johns
Hopkins University Press, 1990.</span></a> 97)</p></blockquote>';
      break;
case '11':
      echo '<p><span class="volvelles">__</span><p>nbsp;On a practical level,
Harsd&ouml;rffer envisions the Denckring as part of his  poetic process. The
device makes its first appearance in his <i>Poetischer Trichter</i> or
<i>Poet&#39;s Funnel</i>, a  book on different theories and forms of verse
and, as is written in the <i>Delitiae Mathematicae et Physicae</i>, <a
href="#">&#34;hat  ... seinen Gebrauch in Erfindung der Reimw&ouml;rter /
wann man die Reimsilben auf dem dritten und vierten Ring suchet  und die
Reimbuchstaben auf dem zweyten Ring darzu drehet&#34;<span>has ... its use in
the invention of rhyme-words / when one seeks the rhymes on the third and
fourth rings, and turns the rhyme-letters of the second ring</span></a> (<a
href="bibliography.html#schwenter"
target="_blank"><i>Delitiae</i><span>Harsd&ouml;rffer, Georg Philipp and
Daniel Schwenter. <i>Delici&aelig; Physico-Mathematic&aelig;, Oder, Mathemat.
Und Philosophische Erquickstunden...</i> N&uuml;rnberg: in Verlegung Jeremiae
D&uuml;mlers, 1651.</span></a> II.518). Similarly, unlike columnar tables
printed in books &mdash; indeed, if the Denckring were printed as a folio
dictionary, it would fill thousands of volumes &mdash; the  structure of
nesting rings allows the poet to identify words that match a particular rhyme
scheme, provides immediate  random access to the lexicon, and is mobile,
encouraging poets to incorporate it into poetic games and performance
pieces.</p>';
      break;
case '12':
      echo '<p><span class="volvelles">__</span><p>nbsp;Perhaps most
importantly, in using the Denckring the poet is not locating an  item in a
list, but dynamically generating meaning in the moment: that is, the very
motion of the poet&#39;s hand  assembles a whole word out of dissembled
parts, engaging the baroque spirit of <i>inventio</i>. More than a mere
storage  device, the Denckring is a writing implement, mediating the
```

```
relationship between the poet&#39;s physical body and the  generated
text.</p>';
      break;
case '13':
      echo '<p><span class="volvelles">__</span><p>nbsp;God assembled the
world from thirty-two divine letters and numbers; the poet  spins a wheel
inscribed with trace elements of the <i>lingua adamica</i> to generate
97,209,600 wor(l)ds, many of which  exist only as possibilities. By embodying
the theory of Stammw&ouml;rter, the Denckring also embodies its
contradictions,  turning the very institution that legitimates it &mdash;
that is, God, the divine &mdash; into a mere mechanism that dumbly spits out
meaning. For if the poet has at her disposal the material elements of
creation and God&#39;s combinatory metric, what  what Book of Nature could be
read that the poet couldn&#39;t already write herself?</p>';
      break;
case '14':
      echo '<p><span class="stammworter">__</span><p>nbsp;Combinatorial
creation, anagrams, the notion of root elements &mdash; these  ideas converge
in the concept of Stammw&ouml;rter, or German morphemes rooted in the
originary language of  Adam.</p>';
      break;
case '15':
      echo '<p><span class="stammworter">__</span><p>nbsp;One of the most
well-known German linguists of the early seventeenth century  and a proponent
of Stammw&ouml;rter theory, Justus Georg Schottel wrote a series of grammars
promoting the antiquity and  expressiveness of German, calling for (like
Opitz&#39;s <i>Aristarchus</i>) a return to the pure, Adamic vernacular. The
first,  <i>Teutsche Sprachkunst</i> (1641), begins with a collection of ten
&#34;Testimonia der Gelarten von der Trefflichkeit der  deutschen
Sprache,&#34; or testimonies from the learned on the excellence of the German
language. Just as the  kabbalist&#39;s Sefirot connects the ten digits to the
the ten manifest attributes of God, each of the ten testimonies in the
<i>Teutsche Sprachkunst</i> expand upon a different quality of excellence
inherent in the German tongue, laying the  foundation for Schottel&#39;s own
commentary on German&#39;s combinatorial capacity.</p>';
      break;
case '16':
      echo '<p><span class="stammworter">__</span><p>nbsp;A year after the
publication of the <i>Teutsche Sprachkunst</i>, Prince Ludwig  I of Anhalt-
K&ouml;then invited Schottel to join the <i>Fruchtbringende Gesellschaft</i>,
where he was granted the title <i>der  Suchende</i>, or &#34;the seeker,&#34;
and the emblem of a cabbage. Shortly after that, Schottel, who had been
tutoring  Duke August the Younger of Brunswick-L&uuml;neburg&#39;s young sons
since 1638, was joined in the Duke&#39;s household  by Sigmund von Birken, a
prolific poet and friend to Georg Philipp Harsd&ouml;rffer. In 1645 Schottel
himself became the  tenth member of Harsd&ouml;rffer&#39;s Pegnesische
Blumenorden under the name &#34;Fontano,&#34; the adjectival  version of the
Latin word for &#34;fountain&#34; or &#34;spring.&#34;</p>';
      break;
case '17':
      echo '<p><span class="stammworter">__</span><p>nbsp;The security of
royal patronage, friendships with imaginative thinkers and  access to the
most extensive library in Europe, the Duke&#39;s Herzog August Library,
helped Schottel weave together  his linguistic theory, most fully developed
in the massive <i>Ausf&uuml;hrliche Arbeit von der teutschen Haubtsprache</i>
(1663). To historicize his native tongue, Schottel traces the German language
back to Ashkenaz, <a href="#">&#34;ein Altvater der  Teutschen / hat mit sich
die alte Celtische oder Teutsche Sprache von Babel gebrach&#34;<span>the
```

```
patriarch of the Germans / who brought the old Celtic or German language with
him from Babel</span></a> and from Ashkenaz all the way  back to Adam (<a
href="bibliography.html#schottel-haubtsprache" target="_blank">Schottel
<i>HaubtSprache</i><span>Schottelius, Justus Georg. <i>Ausfuehrliche Arbeit
Von Der Teutschen Haubt Sprache, Worin Enthalten Gemelter Dieser Haubt
Sprache Uhrankunft, Uhraltertuhm, Reinlichkeit, Eigenschaft, Vermoegen,
Unvergleichlichkeit, Grundrichtigkeit, Zumahl Die Sprach Kunst Und Vers Kunst
Teutsch Und Guten Theils Lateinisch Voellig Mit Eingebracht, Wie Nicht
Weniger Die Verdoppelung, Ableitung, Die Einleitung, Nahmwoerter, Authores
Vom Teutschen Wesen Und Teutscher Sprache, Von Der Verteutschung, Item Die
Stammwoerter Der Teutschen Sprache Samt Der Erklaerung Und Derogleichen Viel
Merkwuerdige Sachen ... Ausgefertiget Von Justo-Georgio Schottelio ..</i>
Braunschweig: Gedrukt und verlegt durch C. F. Zilligern, 1663.</span></a>
I.34). Thus, while the Confusion of Babel may have muddied the <i>lingua
adamica</i>, it did not completely erase it: in fact, German, Schottel
claims, has its roots in the pure vernacular of the Garden  of Eden.</p>';
      break;
case '18':
      echo '<blockquote><p class="blockquote"><span
class="stammworter">__</span><p>nbsp;<a href="#">Dem allen nach wird
gewi&szlig;lich folgen da&szlig; / gleich wie  das jtzige Teutschland annoch
dasselbe Teutschland ist / welches vor etzlichen tausend Jahren gewesen / ob
es schon  jtzo `besser bebauet / herrlicher ausgeschm&uuml;kt<span>It follows
then accordingly that / just as the present Germany is the same Germany /
that existed one thousand years ago / although it is better developed and
more magnificently adorned</span></a><a href="#"> ... Also ist gleichfals
unsere jtzige Teutsche Sprache / eben dieselbe  uhralte weltweite Teutsche
Sprache<span>So is our German language currently / the same ancient,
universal German language</span></a><a href="#> ... denn wie das Land /
Teutschland bleibet / also m&uuml;ssen die Stammw&ouml;rter /  Teutsche
W&ouml;rter bleiben die denn ihre nat&uuml;rliche Eigenschaften (davon in
dieser Sprachkunst wird gehandelt) so  lange in sich / samt jhrer Deutung /
gehabt haben / so longe sie in rerum natura gewesen.<span>thus just as the
country / remains German / also must the stem-words / remain German words
that retain their natural properties (which are traded in this linguistic
theory) / and their meaning / as they have been rerum natura.</span></a> (<a
href="bibliography.html#schottel-haubtsprache" target="_blank">Schottel
<i>HaubtSprache</i><span>Schottelius, Justus Georg. <i>Ausfuehrliche Arbeit
Von Der Teutschen Haubt Sprache, Worin Enthalten Gemelter Dieser Haubt
Sprache Uhrankunft, Uhraltertuhm, Reinlichkeit, Eigenschaft, Vermoegen,
Unvergleichlichkeit, Grundrichtigkeit, Zumahl Die Sprach Kunst Und Vers Kunst
Teutsch Und Guten Theils Lateinisch Voellig Mit Eingebracht, Wie Nicht
Weniger Die Verdoppelung, Ableitung, Die Einleitung, Nahmwoerter, Authores
Vom Teutschen Wesen Und Teutscher Sprache, Von Der Verteutschung, Item Die
Stammwoerter Der Teutschen Sprache Samt Der Erklaerung Und Derogleichen Viel
Merkwuerdige Sachen ... Ausgefertiget Von Justo-Georgio Schottelio ..</i>
Braunschweig: Gedrukt und verlegt durch C. F. Zilligern, 1663.</span></a>
I.48)</p></blockquote>';
      break;
case '19':
      echo '<p><span class="stammworter">__</span><p>nbsp;<b>Rerum
natura</b>, the nature of things: an allusion to <i>De rerum  natura</i>, a
poem by Lucretius circa the first century B.C. In it, Lucretius posits an
Epicurean, atomistic world of infinite  possible combinations &mdash; of <a
href="#">&#34;dissimiles ... formae glomeramen in unum /
conveniunt&#34;<span>globes dissimilar in form joined together as
one</span></a> (<a href="<a href="http://www.perseus.tufts.edu/cgi-
bin/ptext?doc=Perseus%3Atext%3A1999.02.0130;query=card%3D%2342;layout=;loc=2.
```

```
581" target="_blank">Lucretius<span>Lucretius, <U>De rerum
natura.</U></span></a> II.686-7).  For Lucretius and Epicurus before him,
each atom is a discrete unit of irreducible and self-evident meaning <a
href="#">&#34;quod  nusquam sine permitiali / discidio potis est seiungi
seque gregari, / pondus uti saxis, calor ignis, liquor aquai&#34;<span>since
on no occasion is one able to be separated from the group without a
destructive tearing apart, as mass from stone, heat from fire, or liquid from
water</span></a> (<a href="<a href="http://www.perseus.tufts.edu/cgi-
bin/ptext?doc=Perseus%3Atext%3A1999.02.0130;query=card%3D%2313;layout=;loc=1.
483" target="_blank">Lucretius<span>Lucretius, <U>De rerum
natura.</U></span></a> I.451-3).</p>';
      break;
case '20':
      echo '<p><span class="stammworter">__</span><p>nbsp;Although not
explicitly tied to the creation myth &mdash; in fact, Lucretius  appears to
explicitly reject Adamic theory (see <a href="bibliography.html#eco-perfect"
target="_blank">Eco<span>Eco, Umberto. <U>The Search for the Perfect
Language.</U> Cambridge, Mass.,: Blackwell, 1995.</span></a> 88-9) &mdash;
Lucretius&#39; atomism provides the groundwork  for conceptualizing
Schottel&#39;s Stammw&ouml;rter. Like atoms, Stammw&ouml;rter are
irreducible, isolatable  units of meaning, much as linguists today might
speak of morphemes. However, unlike morphemes Stammw&ouml;rter are not
rational conceptions of the human mind, but are inextricably interwoven with
the material text. For Schottel they are like  letters, having size and
weight:</p><blockquote><p class="blockquote"><a href="#">Gleich wie aber die
Teutschen letteren oder Buchstabe alle einlautend<span>Just as the German
letters or Buchstabe are all shrunken</span></a><a href="#">  / eben also
sind die ersten Surtzelen oder die Stammw&ouml;rter der Teutschen Sprache
gleichfalls einsilbig.<span> / even also are the first root or stem-words of
the German tongue likewise one syllable</span></a> (<a
href="bibliography.html#schottel-haubtsprache" target="_blank">Schottel
<i>HaubtSprache</i><span>Schottelius, Justus Georg. <i>Ausfuehrliche Arbeit
Von Der Teutschen Haubt Sprache, Worin Enthalten Gemelter Dieser Haubt
Sprache Uhrankunft, Uhraltertuhm, Reinlichkeit, Eigenschaft, Vermoegen,
Unvergleichlichkeit, Grundrichtigkeit, Zumahl Die Sprach Kunst Und Vers Kunst
Teutsch Und Guten Theils Lateinisch Voellig Mit Eingebracht, Wie Nicht
Weniger Die Verdoppelung, Ableitung, Die Einleitung, Nahmwoerter, Authores
Vom Teutschen Wesen Und Teutscher Sprache, Von Der Verteutschung, Item Die
Stammwoerter Der Teutschen Sprache Samt Der Erklaerung Und Derogleichen Viel
Merkwuerdige Sachen ... Ausgefertiget Von Justo-Georgio Schottelio ..</i>
Braunschweig: Gedrukt und verlegt durch C. F. Zilligern, 1663.</span></a>
I.61)</p></blockquote>';
      break;
case '21':
      echo '<p><span class="stammworter">__</span><p>nbsp;Like atoms and
letters, or like the primitive concepts of Leibniz&#39;s <i>lingua
characteristica</i> (<a href="bibliography.html#westerhoff"
target="_blank">Westerhoff<span>Westerhoff, Jan C. &quot;Poeta Calculans:
Harsd&ouml;rffer, Leibniz, and the <i>Mathesis Universalis</i>.&quot;
<i>Journal of the History of Ideas</i> 60.3 (1999): 449-67.</span></a> 459),
a root-word in Schottel&#39;s baroque Stammw&ouml;rter theory combines
(<i>gef&uuml;get</i>) with other words to form meaning,  <a
href="#">&#34;gewisser massen gebildet werden&#34;<span>forming a certain
bulk</span></a> (<a href="bibliography.html#schottel-haubtsprache"
target="_blank">Schottel <i>HaubtSprache</i><span>Schottelius, Justus Georg.
<i>Ausfuehrliche Arbeit Von Der Teutschen Haubt Sprache, Worin Enthalten
Gemelter Dieser Haubt Sprache Uhrankunft, Uhraltertuhm, Reinlichkeit,
Eigenschaft, Vermoegen, Unvergleichlichkeit, Grundrichtigkeit, Zumahl Die
```

Sprach Kunst Und Vers Kunst Teutsch Und Guten Theils Lateinisch Voellig Mit Eingebracht, Wie Nicht Weniger Die Verdoppelung, Ableitung, Die Einleitung, Nahmwoerter, Authores Vom Teutschen Wesen Und Teutscher Sprache, Von Der Verteutschung, Item Die Stammwoerter Der Teutschen Sprache Samt Der Erklaerung Und Derogleichen Viel Merkwuerdige Sachen ... Ausgefertiget Von Justo-Georgio Schottelio .. Braunschweig: Gedrukt und verlegt durch C. F. Zilligern, 1663. I.70). These combinations are possible because one radical — that is, one signifier — corresponds directly and immediately to the concept it signifies. This relationship is not one of mirroring, in which two distinct entities reflect one another, but, to return the materialism inherent in Stammwörter, of embodiment. Thus in some sense a root word simply is the essence a concept. ';
      break;
case '22':
      echo '<blockquote> __ nbsp;Durch die natürlich bekandte Unmüglichkeit ist es schlecht unmüglich eine leichtere / gründlichere und wundersamere Art der Letteren oder Buchstaben und Wörter / als die Teutschen sind auszubringen: Sie sind nicht allein einlaufend / die durch einen natürlichen Zufall den dehörigen Laut veruhrsachen / sonderen ihr einstimmiger Laut ist so wunderreich / und ihre Zusammenstimmung so überkünstlich / daß die Natur sich hierin v&oumlllig und aller dinges ausgearbeitet hat. Denn / ein jedes Ding / wie seine Eigenschaft und Wirkung ist / also muß es vermittelst unserer letteren / und krast derer / also zusammengefügten Teutschen Wörter / aus eines wolredenden Munde daher fliessen / und nicht anders als ob es gegenwärtig da were / durch des Zuh&oumlrers Sin und Herze dringen. Zum Exempel nehme einer nur diese Wörter: Wasser fliessen / gesäusel / sanft / stille ac wie künstlich ist es / wie gleichsam wesentlich fleust das Wasser mit stillem Besäusel von unser Zungen?By the naturally acquainted impossibility, it is bad, impossible, to spread a lighter / deeper and more miraculous art of letters or Buchstaben and words / than those in German: they are not shrinking alone / but are caused by a natural Chance, a heard sound / but the sound is more concordant than miraculous / and its together-tune so overly artificial / that Nature herself has hereby composed total and all things. Then / every thing / is as its property and effect / and so it must by means of our letters / and their force / also bring the German words together /so they flow from a well-spoken mouth / no different than if it were present as / by the listener's mind and sinking in the heart. For example take just one of these words: Water flows / purrs / soft / still as it is imitated / as, as it were important, water flows with still purrs from our tongues? (Schottel HaubtSpracheSchottelius, Justus Georg. Ausfuehrliche Arbeit Von Der Teutschen Haubt Sprache, Worin Enthalten Gemelter Dieser Haubt Sprache Uhrankunft, Uhraltertuhm, Reinlichkeit, Eigenschaft, Vermoegen, Unvergleichlichkeit, Grundrichtigkeit, Zumahl Die Sprach Kunst Und Vers Kunst Teutsch Und Guten Theils Lateinisch Voellig Mit Eingebracht, Wie Nicht Weniger Die Verdoppelung, Ableitung, Die Einleitung, Nahmwoerter, Authores Vom Teutschen Wesen Und Teutscher Sprache, Von Der Verteutschung, Item Die Stammwoerter Der Teutschen Sprache Samt Der Erklaerung Und Derogleichen Viel Merkwuerdige Sachen ... Ausgefertiget Von Justo-Georgio Schottelio .. Braunschweig: Gedrukt und verlegt durch C. F. Zilligern, 1663. I.59) </blockquote>';
      break;
case '23':
      echo ' __ nbsp;Many baroque thinkers echo Schottel's admiration for the onomatopoeic qualities of

```
the German tongue, taking it as proof of German&#39;s excellence and divine
origins.</p>';
      break;
case '24':
      echo '<p><span class="stammworter">__</span><p>nbsp;In his
<i>Frauenzimmer Gespr&auml;chspiele</i>, Harsd&ouml;rffer argues that  the
German language</p><blockquote><p class="blockquote">Speaks in the languages
of nature, quite perceptibly expressing all its sounds. ... ; it roars like
the lion, lows like the oxen, snarls like the bear, bells like the stag,
whinnies like the horse, hisses like the snake ... On all those occasions in
which nature gives things their own sound, nature speaks in our own German
tongue.  For this, many have wished to assert that the first man, Adam, would
not have been able to name the birds and all the  other beasts of the fields
in anything but our words, since he expressed, in a manner conforming to
their nature, each and  every innate property and inherent sound; and thus it
is not surprising that the roots of the larger part of our words coincide
with the sacred language. (Harsd&ouml;rffer, quoted in <a
href="bibliography.html#eco-perfect" target="_blank">Eco<span>Eco, Umberto.
<i>The Search for the Perfect Language.</i> Cambridge, Mass.,: Blackwell,
1995.</span></a> 99)</p></blockquote>';
      break;
case '25':
      echo '<p><span class="stammworter">__</span><p>nbsp;Here
Harsd&ouml;rffer brings the concept of Stammw&ouml;rter full circle, back to
the <i>lingua adamica</i>. From a finite number of radicals, infinite
meanings become possible, constructing entire linguistic universes unknown to
humans since the Confusion of Babel, perhaps even since the Fall. The more
pure these radicals are &mdash; that is, the more they embody the divine
properties of the universe, the Paths of Wisdom &mdash; the closer their
combinations come to God&#39;s creation. Thus finding and excavating
Stammw&ouml;rter in the German tongue not only legitimates the vernacular,
uplifting it from its state of corruption and neglect, but also exalts its
speakers by allowing them to enter a more direct, almost prelapsarian
communion with God.</p>';
      break;
case '26':
      echo '<p><span class="stammworter">__</span><p>nbsp;Yet, ironically, by
presenting Stammw&ouml;rter as a form of onomatopoeia, Schottel and
Harsd&ouml;rffer undermine the very theory they are attempting to support.
Once linguistic perfection is no longer anchored in the moment of Adamic
naming, any language, divinely formed or artificial, may become more perfect
simply by molding its sounds to the &#34;language of nature.&#34; Likewise,
if humans can excavate the <i>lingua adamica</i> from an ordinary vernacular,
then they possess the basic elements of creation, the building blocks from
which all meaning constructed, and hence the power to re-construct the world
in their own image &mdash; to build a new Tower of Babel.</p>';
      break;
case '27':
      echo '<p><span class="stammworter">__</span><p>nbsp;Although Schottel
takes pains to historicize his project, to present it as a descriptive
account of the German tongue, in the end it is both prescriptivist and
interventionist, wresting the power of the divine and placing it in the hands
of man.</p>';
      break;
case '28':
      echo '<p><span class="stammworter">__</span><p>nbsp;The first half of
the seventeenth century was a period of transition, when scholars and poets
and writers and publishers were struggling to develop new standards for the
```

```
changing practices of art, and for the emerging culture of experimentation in
natural philosophy (see, for instance, <a href="bibliography.html"
target="_blank">Johns<span>Johns, Adrian. <U>The Nature of the Book: Print
and Knowledge in the Making.</U> Chicago, IL: University of Chicago Press,
1998.</span></a>). Of course, the baroque era was also a period of spiritual
and social upheaval, as bitter religious wars ravaged central Europe. One
must expect that the linguistic theories of Schottel and Harsd&ouml;rffer
would reflect these competitions between the mechanical and the divine, and
natural and the artificial.</p>';
      break;
case '29':
      echo '<p><span class="stammworter">__</span><p>nbsp;As Jane Newman
points out in a clever deconstruction of New Historicism, history is not
simply the &#34;background, &#39;origin,&#39; and ultimate goal of textual
expression,&#34; and texts are not merely &#34;mirrors of historical
conditions &#39;on the outside&#39;&#34; (<a href="bibliography.html#newman"
target="_blank">Newman<span>Newman, Jane O. <i>Pastoral Conventions: Poetry,
Language, and Thought in Seventeenth-Century  Nuremberg.</i> Baltimore: Johns
Hopkins University Press, 1990.</span></a> 7, 10). In fact very few
linguistic texts of the German baroque &#34;&#39;refer&#39; or
&#39;respond&#39; to an &#39;outside&#39;&#34; history at all (<a
href="bibliography.html#newman" target="_blank">Newman<span>Newman, Jane O.
<i>Pastoral Conventions: Poetry, Language, and Thought in Seventeenth-Century
Nuremberg.</i> Baltimore: Johns Hopkins University Press, 1990.</span></a>
10). Rather, Newman argues, texts are institutions, which &#34;means, above
all, understanding them as historical events in a maximally concrete
sense&#34;: as &#34;a material phenomenon that comes into existence at a
given moment and testifies to that moment&#34; by both &#34;separating itself
(as text) from it and by embedding it in a rhetoric of transcendence&#34; (<a
href="bibliography.html#newman" target="_blank">Newman<span>Newman, Jane O.
<i>Pastoral Conventions: Poetry, Language, and Thought in Seventeenth-Century
Nuremberg.</i> Baltimore: Johns Hopkins University Press, 1990.</span></a>
20).</p>';
      break;
case '30':
      echo '<blockquote><p class="blockquote"><span
class="stammworter">__</span><p>nbsp;To study the institutional identity of a
text as well as the function of textual institutions relies, then, on
analyzing both the inaugural function of the text, its power to be
foundational and to provide a historical beginning, and its monumental
function, the strategies whereby it not only testifies to that beginning but
also transcends it by serving as a fixed model for specific future (perhaps
not only textual) behavior. These strategies gain their authority by
strategically &#39;forgetting&#39; and then recoding in a
&#39;monumental&#39; textual form the historicity of the inaugural moment as
a moment beyond and outside history. (<a href="bibliography.html#newman"
target="_blank">Newman<span>Newman, Jane O. <i>Pastoral Conventions: Poetry,
Language, and Thought in Seventeenth-Century  Nuremberg.</i> Baltimore: Johns
Hopkins University Press, 1990.</span></a> 22)</p></blockquote>';
      break;
case '31':
      echo '<p><span class="stammworter">__</span><p>nbsp;Schottel&#39;s
theory is both monument and foundation &mdash; it both historicizes the
vernacular and de-centers the very Judeo-Christian narrative that legitimizes
it. In short, like Stammw&ouml;rter themselves, it both reflects and
transcends its moment of creation.</p>';
      break;
case '32':
```

```
      echo '<blockquote><p class="blockquote"><span
class="kabbalah">__</span><p>nbsp;The great lesson that Kabbalah can teach
contemporary interpretation is that meaning in belated texts is always
wandering meaning, even as the belated Jews were a wandering people. Meaning
wanders, like human tribulation, or like error, from text to text, and within
a text, from figure to figure. What governs this wandering, this errancy, is
defense, the beautiful necessity of defense. For not just interpretation is
defense, but meaning itself is defense, and so meaning wanders to protect
itself. In its etymology, &#39;defense&#39; refers to &#39;things
forbidden&#39; and to &#39;prohibition&#39;, and we can speculate that poetic
defense rises in close alliance with the notions of trespass and
transgression, crucial for the self-presentation of any new strong poet. (<a
href="bibliography.html#bloom" target="_blank">Bloom<span>Bloom, Harold.
<i>Kabbalah and Criticism.</i> New York: Seabury Press, 1975.</span></a>
82)</p></blockquote>';
      break;
case '33':
      echo '<p><span class="kabbalah">__</span><p>nbsp;As Gershom Scholem
puts it, the <i>Sefer Yetzirah</i>, or <i>Book of Creation</i>, is &#34;small
in size but enormous in influence&#34; (<a href="bibliography.html#scholem-
kabbalah" target="_blank">Scholem<span>Scholem, Gershom Gerhard.
<i>Kabbalah.</i> New York: Quadrangle/New York Times  Book Co,
1974.</span></a> 23). Written sometime around the eighth century, it is the
earliest extant text in the Kabbalist tradition, laying out a cosmogeny of
combination and permutation. </p>';
      break;
case '34':
      echo '<blockquote><p class="blockquote"><span
class="kabbalah">__</span><p>nbsp;With 32 mystical paths of
Wisdom<br/>engraved Yah<br/>     the Lord of Hosts<br/>     the God of
Israel<br/>the living God<br/>     King of the universe<br/>El Shaddai<br/>
Merciful and Gracious<br/>     High and Exalted<br/>     Dwelling in
Eternity<br/>     Whose name is Holy &mdash; <br/>          He is lofty and
holy &mdash; <br/>And He created His universe<br/>     with three books
(Sepharim), <br/>          with text (Sepher) <br/>          with number
(Sephar) <br/>          and with communication (Sippur). (<a
href="bibliography.html#kaplan" target="_blank">Sepher Yetzirah<span>Kaplan,
Aryeh, ed. <i>Sefer Yetzirah</i>. York Beach, Me.: S. Weiser,
1997.</span></a> 1.1)</p></blockquote>';
      break;
case '35':
      echo '<p><span class="kabbalah">__</span><p>nbsp;In Kabbalah, the
thirty-two paths created by God come from 1) the ten digits, futher manifest
in the Ten Sefirot, or emanations of God, and 2) the twenty-two letters of
the Hebrew alphabet, which come together through 231 &#34;gates,&#34;
clusters of letters that form the roots of the Hebrew verb (<a
href="bibliography.html#scholem-kabbalah"
target="_blank">Scholem<span>Scholem, Gershom Gerhard. <i>Kabbalah.</i> New
York: Quadrangle/New York Times  Book Co, 1974.</span></a> 25). &#34;He hath
formed, weighed and composed with these twenty-two letters every created
thing,&#34; the <i>Sefir Yetzirah</i> reads, &#34;and the form of everything
which shall hereafter be&#34; (<a href="bibliography.html#kaplan"
target="_blank">Sepher Yetzirah<span>Kaplan, Aryeh, ed. <i>Sefer
Yetzirah</i>. York Beach, Me.: S. Weiser, 1997.</span></a> II.2). </p>';
      break;
case '36':
      echo '<p><span class="kabbalah">__</span><p>nbsp;Aryeh Kaplan explains
the basic principles of Kabbalah, which left a deep impression on
```

```
seventeenth-century European thinkers:</p><blockquote><p
class="blockquote">The letters and digits are the basis of the most basic
ingredients of creation, quality and quantity. The qualities of any given
thing can be described by words formed out of letters, while all of its
associated quantities can be expressed by numbers. (<a
href="bibliography.html#kaplan" target="_blank">Kaplan<span>Kaplan, Aryeh,
ed. <i>Sefer Yetzirah</i>. York Beach, Me.: S. Weiser, 1997.</span></a>
5)</p></blockquote>';
      break;
case '37':
      echo '<p><span class="kabbalah">__</span><p>nbsp;Much like the
Christian &#34;Book of Nature,&#34; which sees God&#39;s creation as an all-
encompassing text coextensive with the Bible, the <i>Sefir Yetzirah</i> of
the Jewish Kabbalah treats the world as fundamentally textual, a spatial
environment to be &#34;read.&#34; However, unlike the &#34;Book of
Nature,&#34; Kabbalism is interested not only in reading the world as the
end-product of God&#39;s work, but in (re)producing the mechanisms of
creation to unlock new meaning. Thus for the Kabbalist &#34;the world-process
is essentially a linguistic one, based on the unlimited combinations of
letters&#34; to form new spiritual environments (<a
href="bibliography.html#scholem-kabbalah"
target="_blank">Scholem<span>Scholem, Gershom Gerhard. <i>Kabbalah.</i> New
York: Quadrangle/New York Times  Book Co, 1974.</span></a> 1974 25).</p>';
      break;
case '38':
      echo '<blockquote><p class="blockquote"><span
class="kabbalah">__</span><p>nbsp;We are dealing with a treasure-hoard of the
second degree, one that refers to the notations of nature, which in their
turn indicate obscurely the pure gold of things themselves. The truth of all
these marks &mdash; whether they are woven into nature itself or whether they
exist in lines on parchments and in libraries &mdash; is everywhere the same:
coeval with the institution of God. (<a href="bibliography.html#foucault"
target="_blank">Foucault<span>Foucault, Michel. <i>The Order of Things: An
Archaeology of the Human Sciences.</i> New York:  Pantheon Books,
1970.</span></a> 34)</p></blockquote>';
      break;
case '39':
      echo '<p><span class="kabbalah">__</span><p>nbsp;Kabbalists extend the
idea of manipulation, combination and permutation to all sacred texts,
employing a set of three basic hermeneutic tools to uncover the secret
meanings embedded in the rolls of the Torah. The first, <b>Temurah</b>,
involves rearranging words and sentences to generate new texts from the base
elements of the old, much like anagrams. <b>Notarikon</b> similarly cobbles
together a new text from the old, but using only the first and last letter of
a series of words. And, combining the thirty-two letters and ten digits of
creation, <b>Gematria</b> assigns numerical values to letters which then
acquire quantitative significance.</p>';
      break;
case '40':
      echo '<p><span class="kabbalah">__</span><p>nbsp;Underlying each of
these practices is the belief that letters, written language, exist as
material objects with a primarily spatial, rather than temporal, existence.
Even the voice has a physical presence in the Sefir Yetzirah: it is
&#34;impressed upon the air, and audibly modified in five places; in the
throat, in the mouth, by the tongue, through the teeth, and by the lips&#34;
(<a href="bibliography.html#kaplan" target="_blank">Sepher
Yetzirah<span>Kaplan, Aryeh, ed. <i>Sefer Yetzirah</i>. York Beach, Me.: S.
Weiser, 1997.</span></a> II.3). As a physical entity, then, Scripture is not
```

```
strictly linear but contains multiple dimensions, even extralinguistic
dimensions, unlocked by cutting and anagrammatically permuting the
text.</p>';
      break;
case '41':
      echo '<p><span class="kabbalah">__</span><p>nbsp;In many ways this work
you&#39;re reading now is a Ka(nni)bbalistic reading of its source texts
&mdash; a golem, a monstrous creation that lives dispite the inanimate and
artificial nature of its constituent parts.</p>';
      break;
case '42':
      echo '<blockquote><p class="blockquote"><span
class="kabbalah">__</span><p>nbsp;Meaning, whether in modern poetry or in
Kabbalah, wanders wherever anteriority threatens to take over the whole map
of misreading, or the verbal universe, if that phrase be preferrred. Meaning
swerves, enlarges oppositely, vacates, drives down so as to rise up again,
goes outside in the wan hope of getting itself more on the inside, and at
last attempts to reverse anteriority by forsaking the evasions of mental
space for those of mental time. A poem&#39;s images or Kabbalistic hypostases
are thus types of ambivalence (not of ambiguity) that cope with the burden of
anteriority. (<a href="bibliography.html#bloom"
target="_blank">Bloom<span>Bloom, Harold. <i>Kabbalah and Criticism.</i> New
York: Seabury Press, 1975.</span></a> 89)</p></blockquote>';
      break;
case '43':
      echo '<p><span class="towerofbabel">__</span><p>nbsp;Pieter Brueghel
the Elder&#39;s <i>Tower of Babel</i> (1563) is a golem. Unlike the tall,
toddering towers of Meister der Weltenchronik or, later, Athanasius Kircher,
Brueghel&#39;s creation is a thick mound, slowly spiraling out of the sea,
consuming the surrounding countryside.</p><p>On the one hand, the Tower teams
with human life: tiny figures crawl through its crevices, hanging from
machinery, as others observe its construction from the hillside. Yet, larger
than any biological form, it seems so distinctly <i>in</i>human &mdash; an
inanimate monster composed of decaying parts.</p>';
      break;
case '44':
      echo '<p><span class="towerofbabel">__</span><p>nbsp;Part Ziggurat and
part Colosseum, Brueghel&#39;s <i>Tower</i> embodies the concept of recursion
&mdash; a fundamental principal in both the language of humans and code, the
language of computers. Each level is composed of a series of nesting blocks
that emanate from the center, forming a set of concentric wheels that point
up toward heaven.</p>';
      break;
case '45':
      echo '<p><span class="towerofbabel">__</span><p>nbsp;Yet the blocks are
not entirely planar. By constructing the elements on a slight angle, the
tower&#39;s blocked, circular design and rounded Roman arches are at tension
with its spiraling shape, causing it to crumble from the inside. Indeed,
although it stretches into the clouds, it is clear the tower can never reach
heaven; already its carefully-cut blocks are crumbling to dust, almost as
they are added to the wall. As the king stands beside his eviscerated tower,
inspecting the unused bricks strewn over the hillside, the painting
illuminates the stark difference between design and implementation, between
the dream of unity and the disparate reality of its parts.</p>';
      break;
case '46':
      echo '<p><span class="materiality">__</span><p>nbsp;While today it is
common for philosophers and linguists to conceive of language as a system of
```

```
rationally-understood syntactical relationships between larger units of
meaning, such as words or whole sentences (see, for instance, <a
href="bibliography.html" target="_blank">Chomsky<span>Chomsky, Noam.
<U>Cartesian Linguistics: A Chapter in the History of Rationalist
Thought.</U> New York: Harper and Row, 1966.</span></a>), the Cartesian
revolution in linguistics did not occur until the latter half of the
seventeenth century. During the first half, the letter still dominated
linguistic theory as the primary unit of sound and sense. </p> ';
      break;
case '4611':
      echo '<p><span class="materiality">__</span><p>nbsp;In Nuremberg during
1658, the Czech education reformer John Amos Comenius first published his
<i>Orbis Sensualium Pictus</i>, or &quot;The Visible World in Pictures,&quot;
an encyclopedia of images designed to teach children the alphabet and
vocabulary. Originating from the same theories of the <i>lingua adamica</i>
as Harsd&ouml;rffer&#39;s Denckring, Comenius&#39;s textbook &mdash; most
likely known to G. W. Leibniz and Quirinus Kuhlmann as children &mdash; was
enormously popular in Germany and went into 244 editions between 1658 and
1964 (<a href="bibliography.html" target="_blank">Crain<span>Crain, Patricia.
<U>The Story of A: The Alphabetization of America from <i>The New England
Primer</i> to <i>The Scarlet Letter</i>.</U> Stanford, CA: Stanford
University Press, 2000.</span></a> 27).</p> ';
      break;
case '4622':
      echo '<p><span class="materiality">__</span><p>nbsp;Comenius does not
depict the alphabet for mere mnemonic purposes, but as a way of naturalizing
its structure. In other words, the <i>Orbis Pictus</i> renders the
alphabet&#39;s arbitrary arrangement and strange relationship with sound as,
in fact, a reflection of the natural, divine order of the world. As Crain
writes,</p><blockquote><p class="blockquote">The difference, in this scheme,
between nature and artifice, between what the animals do and what people do,
is taxonomic; Comenian man organizes nature. Adam&#39;s task, naming the
animals, really amounted to putting them in their place; in the Comenian Eden
this amounts to ranging the creatures in alphabetical as well as generic
order. (<a href="bibliography.html" target="_blank">Crain<span>Crain,
Patricia. <U>The Story of A: The Alphabetization of America from <i>The New
England Primer</i> to <i>The Scarlet Letter</i>.</U> Stanford, CA: Stanford
University Press, 2000.</span></a> 36)</p></blockquote>';
      break;
case '4633':
      echo '<p><span class="materiality">__</span><p>nbsp;If, for Comenius,
the alphabet represents the divine natural order, then, learning the alphabet
becomes a process of learning God&#39;s creation. As Cohen writes, in the
<i>Orbis Pictus</i> &quot;the proper acquisition of language must logically
lead to the conversion of the world,&quot; since &quot;proper language
learning would lead to brotherhood, grammar to God&quot; (<a
href="bibliography.html" target="_blank">Cohen<span>Cohen, Murray.
<U>Sensible Words: Linguistic Practice in England, 1640-1785.</U> Baltimore:
Johns Hopkins University Press, 1977.</span></a> 19). Thus as with
Harsd&ouml;rffer&#39;s Denckring, to read a word simply is to write &mdash;
and right &mdash; the universe, as the material manipulation of a finite set
(the 26 letters) unlocks the infinite multitude of God&#39;s creation.</p>';
      break;
case '4644':
      echo '<p><span class="materiality">__</span><p>nbsp;Yet if, for
Comenius, learning the alphabet equates with learning the order of the
universe, then any literate person becomes, in some sense, privy to God&#39;s
secrets. Describing the episteme of recursive, symbolic reflections during
```

```
the sixteenth and seventeenth centuries in Europe, Foucault writes: &quot;To
know must therefore be to interpret: to find a way from the visible mark to
that which is being said by it and which, without that mark, would lie like
unspoken speech, dormant within things&quot; (<a
href="bibliography.html#foucault" target="_blank">Foucault<span>Foucault,
Michel. <u>The Order of Things: An Archaeology of the Human Sciences.</u> New
York:  Pantheon Books, 1970.</span></a> 32). Likewise, the process of reading
a text within this system also writes its physical equivalent into existence,
as the site of literacy enables both the production and consumption of a
universal language.</p> ';
      break;
case '4655':
      echo '<blockquote><p class="blockquote"><span
class="materiality">__</span><p>nbsp;The verbal and visual tropes that
surround the alphabet cloak the fact that the unit of textual meaning &mdash;
the letter &mdash; lacks meaning itself. The alphabet&#39;s semantic vacuum
represents a threat to orthodoxy, for into this space competing meaning
systems may rush. The features of the alphabet&#39;s mask change over time,
drawn as they are from a cultural moment&#39;s fund of meaningfulness. Where
a culture goes to make sense of itself is where the alphabet too scavenges.
While one might be used to thinking in an archeological fashion about
excavating layers of meaning, the student of the alphabetic text digs down
only to reveal a null. It is the content of those layers that yields
interest. (<a href="bibliography.html" target="_blank">Crain<span>Crain,
Patricia. <U>The Story of A: The Alphabetization of America from <i>The New
England Primer</i> to <i>The Scarlet Letter</i>.</U> Stanford, CA: Stanford
University Press, 2000.</span></a> 18)</p></blockquote>';
      break;
case '47':
      echo '<blockquote><p class="blockquote"><span
class="materiality">__</span><p>nbsp;Language was envisioned as an aggregate
of discrete signs, each denoting a mental image, with in turn mirrored a
natural world of separate physical objects. As the syntactical relations
between signs could not be easily visualized, they tended to be neglected.
(<a href="bibliography.html#hudson" target="_blank">Hudson<span>Hudson,
Nicholas. <i>Writing and European Thought, 1600-1830.</i> Cambridge ; New
York, NY, USA:  Cambridge University Press, 1994.</span></a>
43)</p></blockquote>';
      break;
case '48':
      echo '<p><span class="materiality">__</span><p>nbsp;Like many other
baroque thinkers, Harsd&ouml;rffer identified the material form of letters in
objects of the world &mdash; or, rather, found the world in the materiality
of letters. In a visual alphabet from the <i>Delitiae mathematicae et
physicae &ndash; Der mathematischen und philosophischen Erquickstunden,
Dritter Theil</i> (1653, 44-5), Harsd&ouml;rffer plays with signifying nature
of <i>Buchstaben</i>, lending his writing (as Juliet Fleming puts it,
describing early modern graffiti) &#34;meaning in excess both of its
signified content, and of its easily recognized aesthetic dimensions &mdash;
a meaning that has to do with the fact of its appearance in matter&#34; (<a
href="bibliography.html#fleming" target="_blank">Fleming<span>Fleming,
Juliet. <i>Graffiti and the Writing Arts of Early Modern England.</i>
Philadelphia: University of  Pennsylvania Press, 2001.</span></a> 115).</p>';
      break;
case '49':
      echo '<blockquote><p class="blockquote"><span
class="materiality">__</span><p>nbsp;If words represent reality to our
understanding, then, like existing things, they must consist of assembled
```

```
elements. Reducing language to physical properties visually organized on the
page justifies operating on words as objects. If the written and spoken
language can be touched, tabulated, and visualized, then it can be secured,
improved, perfected and rationally taught. (<a href="bibliography.html#cohen"
target="_blank">Cohen<span>Cohen, Murray. <i>Sensible Words: Linguistic
Practice in England, 1640-1785.</i> Baltimore: Johns  Hopkins University
Press, 1977.</span></a> 8)</p></blockquote>';
      break;
case '53':
      echo '<p><span class="hand">__</span><p>nbsp;The connection between the
Denckring and the hand is strong. Four hands populate the edges of the
mechanism, positioned as if they were the reader&#39;s own. The hands in the
upper left and lower right hold instruments &mdash; a compass and a quill,
respectively &mdash; while the other two display symbols, a heavy ring and a
small wreath.</p>';
      break;
case '54':
      echo '<p><span class="hand">__</span><p>nbsp;The hands holding
instruments represent writing as a material, artifactual practice mediated by
instruments, tools and technologies, linking it visually to the emerging
practice of experimental philosophy. In fact, the <i>Delici&aelig; Physico-
Mathematic&aelig;</i> (one of the texts in which the Denckring appears) is
peppered with illustrations of disembodied hands holding tools to measure,
slice, or organize matter &mdash; instruments that, like written language,
help us to understand the surrounding world by rendering it in human
terms.</p>';
      break;
case '55':
      echo '<p><span class="hand">__</span><p>nbsp;The hands in the upper
right and lower left corners of the Denckring depict language as symbolic
&mdash; an atemporal monument that materializes and therefore preserves the
abstract. In Western culture, a wreath represents both eternity and death,
the fate of language upon inscription, while the hand in the top right corner
gestures outward, palm facing up, over the word &#34;unvergessen,&#34; or
&#34;unforgotten.&#34;</p>';
      break;
case '5511':
      echo '<p><span class="hand">__</span><p>nbsp;Thus two hands around the
Denckring illustrate the transient, performative act of writing &mdash; the
quill sits poised, as if preparing to set pen to paper &mdash; while the
other two symbolize its material product: the written text.</p><p>As an
abstracted form known to all humans, then, the hand becomes a monument
&mdash; a structure, much like Harsd&ouml;rffer&#39;s Denckring, that stores
data indefinitely. That is, until the human hand comes to retrieve it through
the gesture of writing.</p>';
      break;
case '5522':
      echo '<p><span class="hand">__</span><p>nbsp;Hands are often found
around volvelles in early printed books, usually in the form of
&#34;pointers.&#34; For example, a woodcut calendar from 1466 calculates the
date of easter, as well as the golden number of any given year, by spinning a
set of nesting wheels against each other. The innermost wheel depicts an
angel pointing with hand and foot at a golden number (<a
href="bibliography.html" target="_blank">Sherman<span>Sherman, Claire Richter
et al. <U>Writing on Hands: Memory and Knowledge in Early Modern Europe.</u>
Chicago, IL: University of Chicago Press, 2002.</span></a> 164-5, Cat. 41).
Similarly, Giambattista della Porta&#39;s popular cryptography manual
contains several decoder dials &mdash; volvelles used to decode encrypted
```

```
texts mechanically &mdash; which feature God&#39;s pointing finger in the
center, poised to decrypt the surrounding symbols (<a
href="bibliography.html" target="_blank">Schmidt<span>Schmidt, Suzanna Karr.
&quot;Constructions both Sacred and Profane: Serpents, Angels and Pointing
Fingers in Renaissance Books with Moving Parts.&quot;</span></a> <a
href="http://www.robertsabuda.com/everythingpopup/suzannekarr.asp"
target="_blank">online</a>).</p>';
      break;
case '56':
      echo '<blockquote><p class="blockquote"><span
class="hand">__</span><p>nbsp;There is then a history of technology that is
also the history of &#34;man,&#34; the programmed/programming machine: the
human written. The human cannot simply be returned to the divine/oral origin;
the hand is there from the start, as the locus of retroactive
redetermination. Thus, from the start, the written being and the writing
being are coincident and differential, opening and enclosing at one and the
same time interiority and exteriority, the human and the technological, the
mind and the body, speech and writing in their narrow sense; from the start,
the relationships between these and all other differential terms exist within
the possibility of protention and retention and the differential possibilties
of rhythms of emergence. (<a href="bibliography.html#goldberg"
target="_blank">Goldberg<span>Goldberg, Jonathan. <i>Writing Matter: From the
Hands of the English Renaissance.</i> Stanford,  Calif.: Stanford University
Press, 1990.</span></a> 24)</p></blockquote>';
      break;
case '5611':
      echo '<p><span class="hand">__</span><p>nbsp;In fact, in the visual
culture of early modern books, the image of the hand often straddles the
boundary between performance and product. Sign languages like George
Dalgarno&#39;s <i>A Manual Alphabet</i> (1680) and John Bulwer&#39;s
<i>Chirologia or the Naturall Language of the Hand</i> (1644) link gestures
with the production of speech, treating hand motions as a form of universal
language, while earlier finger-reckoning systems, such as Luca Pacioli&#39;s,
use digits of bone and sinew to calculate digits of numerical abstraction.
Finger alphabets also proliferated during the late sixteenth century,
facilitating memory through the rehearsed gestures of individual letters and
even, in Giambattista della Porta&#39;s cryptology manual <i>De furtivis
literarum notis</i> (1563), encrypting texts through a gestural performance
that links touched body parts to individual letters (see <a
href="bibliography.html" target="_blank">Sherman<span>Sherman, Claire Richter
et al. <U>Writing on Hands: Memory and Knowledge in Early Modern Europe.</u>
Chicago, IL: University of Chicago Press, 2002.</span></a> 186).</p>';
      break;
case '5622':
      echo '<p><span class="hand">__</span><p>nbsp;Even as the hand performs
language, it is itself a surface for writing during the early modern period.
In thoughtful and extensive exhibit on images of inscribed hands in early
printed books, Claire Richter Sherman points to the many ways that marks on
the hand operated as a form of memory palace, helping humans &#34;to
understand, order and recall abstract concepts&#34; through &#34;iconic
metaphors, bodily mnemonics, and cognitive maps encompassing processes of
association, memory, and recollection&#34; (<a href="bibliography.html"
target="_blank">Sherman<span>Sherman, Claire Richter et al. <U>Writing on
Hands: Memory and Knowledge in Early Modern Europe.</u> Chicago, IL:
University of Chicago Press, 2002.</span></a> 13; see Cat. 56, 64).</p>';
      break;
case '5633':
```

```
      echo '<p><span class="hand">__</span><p>nbsp;In Stephan Fridolin&#39;s
<i>Schatzbehalter</i> (1491), a series of woodcuts by Michael Wohlgemut
(Albrecht D&uuml;rer&#39;s instructor) depict the human hand as a database
storing a set of 50 roman numerals, each value linking back to a meditation
in the text. In addition to using the hand as a storage device, Fridolin
encourages his readers to cut out large paper hands and tag them with
keywords that will help the penitent call up associated text and images (see
<a href="bibliography.html" target="_blank">Sherman<span>Sherman, Claire
Richter et al. <U>Writing on Hands: Memory and Knowledge in Early Modern
Europe.</u> Chicago, IL: University of Chicago Press, 2002.</span></a>- 66).
Thus the hand becomes a technology of inscription that not only triggers the
reader&#39;s memory, but actually forces the reader to compose her own texts
through a meditative form of cut-ups and recombinations.</p>';
      break;
case '5644':
      echo '<p><span class="hand">__</span><p>nbsp;Manicules &mdash; the
&#34;pointing hands&#34; that annotate the margins of so many books across
the ages &mdash; slide along the same sharp edge of performance and monument,
motion and storage. As Clanchy points out, these hand-drawn symbols might be
seen as a form of <i>ars memorativa</i>, &#34;a simple way of facilitating
the retrieval of information&#34; (<a href="bibliography.html"
target="_blank">Clanchy<span>Clanchy, M. T. <U>From Memory to Written Record:
England 1066-1307.</U> Cambridge, MA: Harvard University Press,
1979.</span></a> 172). Yet, as Will Sherman argues, the manicule also
&#34;has a gestural function that extends beyond its straightforwardly
indexical one&#34; (<a href="bibliography.html"
target="_blank">Sherman<span>Sherman, William H. <U>Used Books: Marking
Readers in Renaissance England.</U> Philadelphia, PA: University of
Pennsylvania Press, 1008.</span></a> 51). Operating within a dense network of
inscription, the manicule originates in a moment of reading that becomes the
act of writing a hand that points at, and thereby &#34;stores,&#34; a slice
of text. Like Harsd&ouml;rffer&#39;s Denckring, like Burroughs&#39; cut-ups,
the manicule forces us to ride the rim between reading and writing, between
textual consumption and textual production.</p>';
      break;
case '57':
      echo '<p><span
class="volvelles">__</span><p>nbsp;Harsd&ouml;rffer&#39;s Denckring collapses
the boundaries between speaking, reading and writing, presenting the totality
of a language that nonetheless does not exist outside the combinatory
potential of its mechanism. Roland Barthes asserts that the &#34;text&#34;
&mdash; an authorless, &#34;irreducible ... plural,&#34; an &#34;explosion, a
dissemination&#34; &mdash; is a relatively recent phenomenon compared to the
&#34;work,&#34; which is a authored and pedigreed piece of Literature (see <a
href="bibliography.html" target="_blank">Barthes<span>Barthes, Roland.
&quote;From Work to Text.&quot; <U>Textual Strategies.</u> Ed. J Harari.
Ithaca, NY: Cornell University Press, 1979.</span></a>), 73ñ81); but in fact
the Denckring shows that the ergodic text is an important part of baroque
culture. In Harsd&ouml;rffer&#39;s poetry generator, the author-function is
algorithmic and abstract, constructing a structure or code which the reader
then must assemble through the motion of the hand &mdash; a motion connected
with that of writing, of measuring.</p>';
      break;
case '58':
      echo '<blockquote><p class="blockquote"><span
class="hand">__</span><p>nbsp;Oral and written are derivative, then, from the
handedness of the human. As Raymond Williams says, there are
&#34;relationships embodied in writing.&#34; For Derrida, that is literally
```

```
true; the writing being is also being written, whether at the instinctive
level or while writing at the computer. In the relationship between writing
in the general and in the specific sense &mdash; in relationships which are
not merely at the order of conceptuality, but which are also social and
historical &mdash; the field of a differential inscription of human being
comes into view. (<a href="bibliography.html#goldberg"
target="_blank">Goldberg<span>Goldberg, Jonathan. <i>Writing Matter: From the
Hands of the English Renaissance.</i> Stanford,  Calif.: Stanford University
Press, 1990.</span></a> 24)</p></blockquote>';
      break;
case '59':
      echo '<p><span class="thirtyyearswar">__</span><p>nbsp;The sixteenth
and early seventeenth centuries find central Europe weak and disorganized.
Around three hundred different princes, prelates, counts and knights ruled
over a loosely-affiliated smattering of states, provinces, towns and
fiefdoms, all grouped under the central governance of the Holy Roman Empire.
Foreign leaders such as the King of Spain held fiefs within the Burgundian
Circle, while German rulers held foreign territories, causing them to be both
directly under and outside the authority of the Holy Roman Empire. Emperor
Charles V himself, a Habsburg, ruled the Spanish realms of Castile and Aragon
and often appealed to the King of Spain to help suppress internal divisions
&mdash; an alliance which threatened the independence of the German princes,
who retaliated by cementing their own alliances with the King of France, the
Emperor&#39;s enemy.</p>';
      break;
case '60':
      echo '<p><span class="thirtyyearswar">__</span> Religious conflict
further strained the already tenuous ties between local rulers and the
Emperor. In 1555, after a series of Protestant uprisings in the first half of
the sixteenth century, Emperor Ferdinand I, brother and successor to Charles
V, signed a peace treaty with the Schmalkaldic League, a group of Lutheran
princes, dukes and rulers. The treaty, called the Peace of Augsburg,
officially (though not effectively) ended the religious conflict by the
principle of <i>cuius regio, eius religio</i>, or &#34;whose region, his
religion&#34;: that is, whatever religion the leader of the territory
believed in, whether Catholic or Protestant, so must all subjects
believe.</p>';
      break;
case '61':
      echo '<p><span class="thirtyyearswar">__</span> Although intended
to curb violence, in fact, by cementing religious divisions through a policy
of mutual intolerance, the Augsburg Peace agreement ironically destroyed any
lingering sense of unity amongst the scattered states of the Holy Roman
Empire and further weakened the authority of state leaders; for, as C. V.
Wedgwood writes, the Lutheran princes and dukes &#34;were demanding from the
Emperor what they refused to their own people&#34; (<a
href="bibiography.html#wedgwood" target="_blank">Wedgwood<span>Wedgwood, C.
V. <i>The Thirty Years War.</i> New York: New York Review Books,
2005.</span></a> 46). Those who refused to convert to the state&#39;s
religion were forced to emigrate, leading to large populations of refugees
spreading of disease and famine across an already unstable region. By 1618,
tensions were so high that when Ferdinand II, another Habsburg and a staunch
Catholic, became King of the largely Protestant Bohemia, the nobility threw
his representatives out a window, starting a religious war that would last
the next thirty years.</p>';
      break;
case '62':
```

```
      echo '<p><span class="thirtyyearswar">__</span> As Ferdinand&#39;s
imperial governors were falling (quite luckily) onto a pile of manure in
Prague, Martin Opitz, a young Protestant student from Lower Silesia, was
traveling north along the Oder toward Frankfurt an der Oder, a mid-size
trading town nestled in the heart of Brandenburg. When Opitz arrived in 1618
to study law, the state of Brandenburg, like most of the Empire, was cut deep
with religious divisions. Although resistant to the Reformation in the early
decades of the sixteenth century, its rulers had since &#34;jerked from
Lutheranism to Calvinism and back, blazing a trail of dispossession, exile
and violence&#34; that threatened to destroy what little religious harmony
remained within the state (<a href="bibiography.html#wedgwood"
target="_blank">Wedgwood<span>Wedgwood, C. V. <i>The Thirty Years War.</i>
New York: New York Review Books, 2005.</span></a> 46).</p>';
      break;
case '63':
      echo '<blockquote><p class="blockquote"><span
class="thirtyyearswar">__</span> In Brandenburg the Elector declared
that he would rather burn his only University than allow one Calvinist
doctrine to appear in it. Nevertheless his successor became a Calvinist and
introduced a pastor at Berlin, whereat the Lutheran mob broke into the
newcomer&#39;s house and plundered it so effectively that he had to preach on
the following Good Friday in a bright green undergarment, which was all that
the rioters had left him. (<a href="bibliography.html#wedgwood"
target="_blank">Wedgwood<span>Wedgwood, C. V. <i>The Thirty Years War.</i>
New York: New York Review Books, 2005.</span></a> 47)</p></blockquote>';
      break;
case '64':
      echo '<p><span class="thirtyyearswar">__</span> It was in this
environment that young Opitz entered the University of Frankfurt an der Oder,
where he published his first essay, a polemic on behalf of the German
language entitled <i>Aristarchus; sive, de contemptu linguae
Teutonicae</i>.</p>';
      break;
case '65':
      echo '<p><span class="thirtyyearswar">__</span> <a
href="#">&#34;Quotiescunq; majores nostros Germanos, viros fortes ac
invictos, cogito,&#34;<span>Whenever I think about our native German
ancestors, strong men and indeed invincible</span></a> Opitz begins, <a
href="#">&#34;religione quadam tacita ac horrore ingenti
percellor&#34;<span>I am knocked down by a certain quiet religiousness and a
powerful reverent awe</span></a>; for the German tongue is &#34;a charming
tongue, a decent tongue, a serious tongue&#34; well-suited to alexandrine and
Neoclassical verse. Yet, to express the artistic potential of his
&#34;charming&#34; native tongue, Opitz chooses Latin, the &#34;corrupt&#34;
language of Catholicism, switching occasionally to German when citing his own
original poetry. Thus reflecting the culture of the Holy Roman Empire itself,
the essay begins by presenting columns of clean, Roman typeface &mdash;</p>';
      break;
case '66':
      echo '<p><span class="thirtyyearswar">__</span> &mdash;- but then
slowly becomes colonized by pockets of Gothic font that playfully bursts
across the page in expressive curls:</p>';
      break;
case '67':
      echo '<p><span class="thirtyyearswar">__</span> While the essay
and even the lines of the pages attempt to contain these Gothic outbursts
through an imposed aesthetic harmony, the forced unity only leads to greater
discord as the exiled, dispossessed tongue attempts to reclaim, even to re-
```

```
colonize, the page. In fact, on this page Latin is quite literally pushed to
the edge, squashed between the enclosing ranks of German.</p>';
      break;
case '68':
      echo '<p><span class="thirtyyearswar">__</span> Thus not only does
the text reflect the kind of political battles being fought at the time, it
actually becomes a soldier in the war, re-drawing territorial boundaries.
Seen from this angle, <i>Aristarchus</i> does not ironically reinforce the
need to use Latin but actually undermines it, forcing Latin toward the
margins as German stakes a claim to its own space.</p><p>The verses may be
Neoclassical in form but, as Opitz makes clear, they belong to German.</p>';
      break;
case '100':
      echo '<p><span class="cutups">__</span> In a text-generating
volvelle, space is a spiral, a nest, a knot. Spinning the wheel unrolls a
series of combinations, much like the mesmerizing list of As, Cs, Ds and Gs
produced by DNA sequencers. Yet the disc itself consumes only a small portion
of the page, perhaps a few inches square. By condensing the art of
combination into an interlocking mechanism, volvelles produce the seemingly
infinite from a finite space.</p><p>By contrast, the cut-up method dismembers
the text. Hands crack back the book&#39;s spine, ripping the sinews of
binding; scissors slice the page and words fall, disfigured, from the
text&#39;s corpus. Unlike the tightly-wound spiral spinning imagined clouds
of combinations, cut-ups expose the text&#39;s spread to violence; they
explode the thick brick of the book into ashen wisps of words.</p>';
      break;
case '101':
      echo '<p><span class="cutups">__</span> Like the four hands
floating around Harsd&ouml;rffer&#39;s Denckring &mdash; like manicules
&mdash; like the reader&#39;s hand spinning the discs &mdash; like the finger
that points and clicks &mdash; the hand will cut up the hand that recombines.
The coherent abstraction of a textual icon, dismembered from its body,
reconnects through the process of an embodied hand that moves, spinning a web
of combinations.</p><p>As William Burroughs wrote, cut-ups are experimental
&mdash; &quot;experimental in the sense of being something to do.&quot;</p>';
      break;
case '102':
      echo '<p><span class="cutups">__</span> When Thomas Jefferson took
scissors to his Bible in the Spring of 1804 &mdash; three years into his
Presidency and less than twenty years after the federation of the United
States of America &mdash; he was profoundly dissatisfied with corrupted,
anti-democratic state of the Christian religion. &quot;In extracting the pure
principles which [Jesus] taught,&quot; he later wrote to his friend John
Adams, &quot;we should have to strip off the artificial vestments in which
they have been muffled by priests, who have travestied them into various
forms, as instruments of riches and power to them&quot; (<a
href="bibliography.html" target="_blank">Jefferson<span>Jefferson, Thomas.
<U>Jefferson&#39;s Extracts from the Gospels: <i>The Philosophy of Jesus</i>
and <i>The Life and Morals of Jesus</i>.</u> Ed. Dickinson W. Adams.
Princeton, NJ: Princeton University Press, 1983.</span></a> 352).</p>';
      break;
case '103':
      echo '<blockquote><p class="blockquote"><span
class="cutups">__</span> We must reduce our volume to the simple
evangelists, select, even from them, the very words only of Jesus, paring off
the Amphibologisms into which they have been led by forgetting often, or not
understanding, what had fallen from him, by giving their own misconceptions
as his dicta, and expressing unintelligibly for others what they had not
```

```
understood themselves. There will be found remaining the most sublime and
benevolent code of morals which has ever been offered to man. I have
performed this operation for my own use, by cutting verse by verse out of the
printed book, and arranging, the matter which is evidently [Jesus&#39;], and
which is as easily distinguishable as diamonds in a dunghill. (<a
href="bibliography.html" target="_blank">Jefferson<span>Jefferson, Thomas.
<U>Jefferson&#39;s Extracts from the Gospels: <i>The Philosophy of Jesus</i>
and <i>The Life and Morals of Jesus</i>.</u> Ed. Dickinson W. Adams.
Princeton, NJ: Princeton University Press, 1983.</span></a>
352)</p></blockquote>';
      break;
case '104':
      echo '<p><span class="cutups">__</span> To purify the Word of God,
Jefferson ordered two clean copies of the New Testament, then proceeded to
dissect each corpus, extracting only those verses that spoke in the
&quot;true&quot; voice of Jesus (<a href="bibliography.html"
target="_blank">Sheridan<span>Sheridan, Eugene R. &quot;Introduction&quot;
<U>Jefferson&#39;s Extracts from the Gospels: <i>The Philosophy of Jesus</i>
and <i>The Life and Morals of Jesus</i>.</u> Ed. Dickinson W. Adams.
Princeton, NJ: Princeton University Press, 1983.</span></a> 27) &mdash;
verses, as Jefferson put it, &quot;as easily distinguishable as diamonds in a
dunghill&quot; (<a href="bibliography.html"
target="_blank">Jefferson<span>Jefferson, Thomas. <U>Jefferson&#39;s Extracts
from the Gospels: <i>The Philosophy of Jesus</i> and <i>The Life and Morals
of Jesus</i>.</u> Ed. Dickinson W. Adams. Princeton, NJ: Princeton University
Press, 1983.</span></a> 352). As he wrote, &quot;the result is an 8vo. of 46.
pages of pure and unsophisticated doctrines, such as were professed and acted
on by the <i>unlettered</i> apostles, the Apostolic fathers, and the
Christians of the 1st. century&quot; (<a href="bibliography.html"
target="_blank">Jefferson<span>Jefferson, Thomas. <U>Jefferson&#39;s Extracts
from the Gospels: <i>The Philosophy of Jesus</i> and <i>The Life and Morals
of Jesus</i>.</u> Ed. Dickinson W. Adams. Princeton, NJ: Princeton University
Press, 1983.</span></a> 352).</p><p>Jefferson titled his work <i>The
Philosophy of Jesus of Nazareth, extracted from the account of his life and
doctrines as given by Mathew, Mark, Luke, & John. being an abidgement of the
New Testament for the use of the Indians unembarrassed with matters of fact
or faith beyond the level of their comprehension.</i></p>';
      break;
case '105':
      echo '<p><span class="cutups">__</span> Thomas Jefferson took his
scissors to a set of Bibles for a second time around 1820, nearly seventeen
years after completing <i>The Philosophy of Jesus</i>. The result was a
multilingual reconstruction entitled <i>The Life and Morals of Jesus of
Nazareth Extracted textually from the Gospels in Greek, Latin, French &amp;
English.</i></p><p>Each spread in the book contains four columns, one for
each language, and includes passages pertaining not only to Jesus&#39;
teachings (as with <i>The Philosophy of Jesus</i>) but also to his life. As
Sheridan points out, whereas Jefferson cut up and recombined his first Bible
&quot;in response to his personal religious needs and his concern with the
problem of maintaining social harmony in a republication nation,&quot; his
second &quot;was strictly a product of his private search for religious
truth&quot; (<a href="bibliography.html"
target="_blank">Sheridan<span>Sheridan, Eugene R. &quot;Introduction&quot;
<U>Jefferson&#39;s Extracts from the Gospels: <i>The Philosophy of Jesus</i>
and <i>The Life and Morals of Jesus</i>.</u> Ed. Dickinson W. Adams.
Princeton, NJ: Princeton University Press, 1983.</span></a> 30).</p>';
      break;
case '106':
```

```
      echo '<p><span class="cutups">__</span> I have used the language
of disfigurement and dissection to describe the cut-up method, as if the body
of the book must remain an inviolable, unimpregnated closure of text. The
cut-up, I have implied, is a kind of Frankenstein, a Golem, a Tower of Babel
that sucks life from its elements as soon as their attacker dislocates and
recombines them into something frighteningly, unnaturally new. </p><p>Yet it
is clear that Jefferson and the women of Little Gidding saw their scissors as
swords of peace, not pillage. His cut-ups did not, he thought, disfigure the
Word but clipped away the blemishes that had accumulated, plucking diamonds
from the dung. In short, the selective recombination of the old gave birth to
a primary, not secondary, text &mdash; a New Son brought to wash the Wor(l)d
of its sins.</p>';
      break;
case '107':
      echo '<p><span class="cutups">__</span> For Jefferson, this new
text was, in the words of Burroughs, a poetry &quot;for everyone!&quot; (<a
href="bibliography.html" target="_blank">Burroughs<span>Burroughs, William.
&quot;The Cut Up Method.&quot; <U>The Moderns: An Anthology of New Writing in
America.</U Ed. Leroi Jones. New York: Corinth Books, 1963. 345-8.</span></a>
346).</p>';
      break;
case '108':
      echo '<blockquote><p class="blockquote"><span
class="cutups">__</span> Nicholas Ferrar&#39;s response to the
increasing acrimony and division in political and religious life [during the
last sixteenth century] was not to join in the fray but to create an
alternative that he hoped would stand outside and rise above that discord.
(<a href="bibliography.html" target="_blank">Ransome<span>Ransome, Joyce.
&quot;Monotessaron: The Harmonies of Little Gidding.&quot; <U>The Seventeenth
Century</U 20.1 (April 2005): 22-52.</span></a> 22)</p></blockquote>';
      break;
case '109':
      echo '<p><span class="cutups">__</span> The cut-up method is an
accident.</p>';
      break;
case '110':
      echo '<p><span class="cutups">__</span> In 1626 &mdash; the same
year England crowned the controversial King Charles I &mdash; Nicholas Ferrar
moved his family from London to Little Gidding, nestled just north of
Cambridge. There, as religious wars were being waged both at home and abroad,
Ferrar&#39;s family prayed, lived a strict routine of self-reliance and, what
concerns us here, constructed &quot;Harmonies&quot; &mdash; cut up and
reassembled religious texts and images.</p>';
      break;
case '111':
      echo '<p><span class="cutups">__</span> Like sewing, weaving and
other forms of text/ile production, assembling the &quot;Harmonies&quot; was
women&#39;s hand-i-work at Little Gidding (<a href="bibliography.html"
target="_blank">Dyck<span>Dyck, Paul. &quot;&#39;So rare a use&#39;:
Scissors, Reading and Devotion at Little Gidding.&quot; <U>George Herbert
Journal</U> 27.1&2 (2003/2004): 67-81.</span></a> 68). The patriarch Nicholas
directed his nieces on the form, structure and verses to compile; then the
women extracted the relevant sections from the text, laid them out on the
table, and recombined them to form a new narrative.</p><p>Thus with
appropriately-phallic &quot;Knives & Cizers&quot; in hand, these nameless
nieces tore into the body of the Bible &mdash; the &quot;Heads&quot; &mdash;
to breathe life into a new text. Like strands of DNA splicing in the womb,
discrete elements must disassemble themselves before reassembling into a new
```

```
organism &mdash; a creature paradoxically the same and yet wholly different
from its parents.</p>';
      break;
case '112':
      echo '<p><span class="cutups">__</span> As Dyck points out, the
source texts often contained marginal commentary, so that &quot;the Little
Gidding books literally sunder much of the biblical textual markup ... from
the textual apparatus that gives it meaning&quot; (Dyck &quot;Marking the
Bible&quot; 2-3). Ensconced in a bedding of commentary, words are woven into
an intellectual tradition, acquiring sense through abstract syntactical
relationships in the same way individual threads form an image when woven
into a tapestry. Sliced from the page, though, they become material objects,
suddenly strange &mdash; all form and no content.</p>';
      break;
case '113':
      echo '<p><span class="cutups">__</span> Although seventeenth-
century women were not often given their own pens and bid write, they were at
times given their own scissors and bid <i>cut, paste</i>. Disallowed from
forming their own, they fed off the words of others.</p>';
      break;
case '114':
      echo '<p><span class="cutups">__</span> Biblical harmonies were
widely known during the sixteenth and seventeenth centuries. In England, for
instance, Thomas Middleton, James Bentley, Hugh Broughton, John Huid, Henry
Garthwait, John Lightfoot and William Gould all produced some variant of a
&quot;Harmony&quot; that attempted to reconcile the books of the Bible into
one coherent narrative, thereby participating in a long tradition of biblical
exegesis that extends beyond the scope of the present work.</p><p>What is
fascinating about the Little Gidding &quot;Harmonies,&quot; then, is not
their drive to narrativize (to harmonize) religious teachings, but the cut-
and-paste methods they employ.</p>';
      break;
case '115':
      echo '<p><span class="cutups">__</span> Like Ramon Llull&#39;s ars
magna &mdash; itself a form of biblical exegesis &mdash; each of the four
Gospels in the Little Gidding books is color-coded and labeled with a letter.
Similarly, variations in type and interspersed images keep each chunk of text
separate from the others.</p><p>Thus whereas most other Harmonies homogenize
a re-ordered narrative into a singular new printing, the Little Gidding books
maintain the unique character of each disparate element, facilitating not
only a form of writing, but a cut-and-paste form of reading. Through visual
homogeneity, the Harmony&#39;s user is empowered to combine different texts
and images in new ways &mdash; that is, to read against the forward-facing
grain of a typical printed book.</p>';
      break;
case '116':
      echo '<p><span class="cutups">__</span> Yet this process does not
individualize the reader (as we might imagine) but actually links her to a
long tradition of  &quot;harmonizing&quot; the Bible through selective
reading. Although we, conditioned by the hat tricks of Tristan Tzara and the
radical literary experiments of William Burroughs, theorize cut-ups and
combinations as a way to &quot;lose the undesired past, to cut [our] way out
of an old identity, if not out of identity itself&quot; (<a
href="bibliography.html" target="_blank">Harris<span>Harris, Oliver.
<U>William Burroughs and the Secret of Fascination.</u> Carbondale: Southern
Illinois University Press, 2003.</span></a> 8), the Little Gidding readers
used the Harmonies to construct a desired past in the present, connecting
their isolated commune to a transhistorical community of readers. To
```

```
individualize the text was to participate in a shared religious practice. In
fact, this collective form of reading cut-ups was rooted in the routine of
the commune. As Stewart points out, the Harmonies were read aloud three times
daily at Little Gidding in a special room called the Concordance Room (<a
href="bibliography.html" target="_blank">Stewart<span>Stewart, Stanley.
<U>George Herbert.</U> Boston: Twayne Publisher, 1986.</span></a> 66).</p>';
      break;
case '117':
      echo '<p><span class="cutups">__</span> The Bolognese naturalist
Ulisse Aldrovandi (1522-1605) collected nature, accruing over 7,000 diverse
specimens throughout his life.</p><p>Francesca Fontana, Aldrovandi&#39;s
<i>uxor dilectissima</i> (as he called her), collected notes. She cut and
collated pieces of Aldrovandi&#39;s manuscripts; compiled scraps from his
library for his <i>Pandechion Epistemonicon</i>; and upon his death, edited
and published his writings (<a href="bibliograph.html"
target="_blank">Blair<span>Blair, Ann. &quot;Reading Strategies for Coping
with Information Overload ca.1550-1700.&quot; <U>Journal of the History of
Ideas</U> 64.1 (2003): 11-28.</span></a> 26-7, <a href="bibliography.html"
target="_blank">Findlen<span>Findlen, Paula. &quot;Masculine Prerogatives:
Gender, Space, and Knowledge in the Early Modern Museum.&quot; <U>The
Architecture of Science.</U> Ed. Peter Louis Galison, Emily Ann Thompson.
Cambridge, MA: MIT Press, 1999.</span></a> 44-5). By remediating and remixing
Aldrovandi&#39;s objects, Fontana rendered his cabinet of curiosities &mdash;
his thesaurus, his treasure-hoard &mdash;  as text, then quite literally
positioned these scraps within the pages of the existing world of
print.</p><p>While Aldrovandi brought the world to Bologna, his wife,
wielding scissors, brought his words to the world.</p>';
      break;
case '118':
      echo '<blockquote><p class="blockquote"><span
class="cutups">__</span> As much as this arrangement [of the Harmonies]
makes possible a reading of one combined account, it also foregrounds the
overlap, or the imbrication, of the texts, drawing attention not to
monolithic unity, but to harmonic difference; the book suggests a unity that
is necessarily expressed exactly through difference. The arrangement suggests
distinct kinds of text and thus of reading: a liturgical and immediate story
of Christ that invites a receptive response, and a scholarly and mediated
story of Christ that exposes the multiple versions that constitute it, and
that invites a critical awareness of the relations between the versions.
While naming the differences between these kinds of reading is helpful, one
must also note how tightly the two relate: receptivity and inquiry are
mutually constituting conditions of the devoted reader. (<a
href="bibliography.html" target="_blank">Dyck<span>Dyck, Paul. &quot;&#39;So
rare a use&#39;: Scissors, Reading and Devotion at Little Gidding.&quot;
<U>George Herbert Journal</U> 27.1&2 (2003/2004): 67-81.</span></a>
70)</p></blockquote>';
      break;
case '119':
      echo '<p><span class="cutups">__</span> It is perhaps not
surprising that Nicholas Ferrar, founder of the Little Gidding community, was
good friends with the visual poet George Herbert. Throughout his poetry,
Herbert cuts, copies and recombines disparate verses and phrases to create
not a set of individual, self-sufficient poems, but a network of verse that,
like the Harmonies, is dependent on patterns, iconography and echoes of the
Scripture.</p>';
      break;
case '11911':
```

```
      echo '<p><span class="cutups">__</span> As Stewart
writes,</p></p><blockquote><p class="blockquote">it is a mistake to overlook
the way in which these lines [from The Temple] &quot;bring in&quot; others in
the collection. As Herbert conceives the interpretation of Scripture, so he
presents his own lines as part of a composite expression, which in turn
reflects a continuing process of the speaker&#39;s struggle to grasp the
hidden meaning (&quot;Such are thy secrets&quot;) of God&#39;s Word and, so,
his intentions for one particular &quot;Christian.&quot; This process
transforms shards of experience that seem unrelated and confused into a new
and refreshing recognition or &quot;story.&quot; (<a href="bibliography.html"
target="_blank">Stewart<span>Stewart, Stanley. <U>George Herbert.</U> Boston:
Twayne Publisher, 1986.</span></a> 67)</p></blockquote>';
      break;
case '120':
      echo '<p><span class="cutups">__</span> In other words, both the
Little Gidding &quot;Harmonies&quot; and Herbert&#39;s pattern explore the
sometimes slippery juncture between written language and its material medium,
reflecting a faithful interpretation of scripture through typography and the
spatial layout of the page. Or, to borrow Katherine Hayles&#39; term, both
the Harmonies and Herbert&#39;s poems operate through material metaphor,
&quot;foreground[ing] the traffic between words and physical artifacts&quot;
(Hayles 22).</p>';
      break;
case '12011':
      echo '<blockquote><p class="blockquote"><span
class="materiality">__</span><p>nbsp;When a literary work interrogates the
inscription technology that produces it, it mobilizes reflexive loops between
its imaginative world and the material apparatus embodying that creation as a
physical presence. Not all literary works make this move, of course, but even
for those that do not, my claim is that <i>the physical form of the literary
artifact always affects what the words (and other semiotic components)
mean</i>. Literary works that strengthen, foreground, an thematize the
connections between themselves as material artifacts and the imaginative
realm of verbal/semiotic signifiers they instantiate open a window on the
larger connections that unite literature as a verbal art to its material
forms. (<a href="bibliography.html" target="_blank">Hayles<span>Hayles, N.
Katherine. <u>Writing Machines.</u> Cambridge, Mass.: MIT Press,
2002.</span></a> 25)</p></blockquote>';
      break;
case '121':
      echo '<p><span class="cutups">__</span> In 1634, the women of
Little Gidding compiled a Harmony for King Charles I from several copies of
the New Testament, two copies of Henry Garthwait&#39;s gospel harmony
<i>Montoessaron</i> (1634), pages from a folio Bible and an assembly of
biblical illustrations (<a href="bibliography.html" target="_blank">Dyck  and
Williams<span>Dyck, Paul and Stuart Williams. &quot;Toward an Electronic
Edition of an Early Modern Assembled Book.&quot;<U>CHWP</U> A.44 (July
2008).</span></a> <a
href="http://www.chass.utoronto.ca/epc/chwp/CHC2007/Dyck_Williams/Dyck_Willia
ms.htm" target="_blank">online</a>). As John Ferrar recorded after his
brother Nicholas&#39;s death, King Charles I considered the book to be
&quot;a rare jewel&quot; (<a href="bibliography.html" target="_blank">Dyck
and Williams<span>Dyck, Paul and Stuart Williams. &quot;Toward an Electronic
Edition of an Early Modern Assembled Book.&quot;<U>CHWP</U> A.44 (July
2008).</span></a> <a
href="http://www.chass.utoronto.ca/epc/chwp/CHC2007/Dyck_Williams/Dyck_Willia
ms.htm" target="_blank">online</a>). Thus the same King who sparked anger by
marrying a Catholic princess; who would soon be embroiled in a bitter civil
```

war with the Long Parliament; and who, within twenty years of receiving the Little Gidding Harmony, would lose his head to the fervently anti-Catholic Oliver Cromwell found solace in a text reassembled from the extracted parts of books both Protestant and Catholic. From discord, the Little Gidding women constructed a Harmony. ';
      break;
case '122':
      echo '<blockquote> __ N.F. having first spent Some time in the contrivance of the Work (wch was comonly an hour every Day) and having given his Nieces directions How & in what Manner they should do it, They with their Cizers cut out of each Evangelist such & such Verses, & layd them together, to make & perfect such & such a Head, or chapter, which when they had first roughly done, then with their Knives & Cizers they neatly fitted each Verse So cutt out, to be pasted downe upon sheets of Paper, & So artifically they performed this new-found-out-way, as it were a new kind of Printing: For all that saw the Bookes when they were done, tooke them to be printed on ye ordinary Way, So finely were ye verses joyned together and with great Presses for that purpose pressed down upon ye white sheets of paper. (quoted in DyckDyck, Paul. "'So rare a use': Scissors, Reading and Devotion at Little Gidding." <U>George Herbert Journal</U> 27.1&2 (2003/2004): 67-81. 69) </blockquote>';
      break;
case '123':
      echo ' __ After the Civil War, Sir John Gibson (1606-1665) — a wealthy Royalist who had been unwaveringly loyal to King Charles I, even guarding York during its siege in 1644 — found himself imprisoned at Durham Castle by 1653, unable to repay his debts. There, with his own life in tatters, he began cutting and pasting a multimedia, multidimensional commonplace book. ';
      break;
case '124':
      echo ' __ Adam Smyth describes Sir John Gibson's manuscript as "breathless, chaotic, fragmented," suggesting "little sense of coherence; even less of a linear order" (SmythSmyth, Adam. "'Rend and teare in peeces': Textual Fragmentation in Seventeenth-Century England." <U>The Seventeenth Century</U> 19.1 (april 2004): 36-52. 38). Handwritten sententiae surround emblematic images; pre-Christian poems and devotional texts mingle; and cuttings are interleaved throughout the book (SmythSmyth, Adam. "'Rend and teare in peeces': Textual Fragmentation in Seventeenth-Century England." <U>The Seventeenth Century</U> 19.1 (april 2004): 36-52. 37). Sometimes Gibson remixes the printed medium in simple ways, as when he appropriates a printed border to frame a verse (see SmythSmyth, Adam. "'Rend and teare in peeces': Textual Fragmentation in Seventeenth-Century England." <U>The Seventeenth Century</U> 19.1 (april 2004): 36-52. 42); other times, the remediated layers of copied verse, printed images and commentary radically reinterpret the source materials, as in (Smyth argues) the collage of texts related to Charles I's execution (see SmythSmyth, Adam. "'Rend and teare in peeces': Textual Fragmentation in Seventeenth-Century England." <U>The Seventeenth Century</U> 19.1 (april 2004): 36-52. 46). And, like Harsdörffer, Kuhlmann, and many other German figures during the

```
same period, Gibson shows a deep interest in anagrams, punctuating his cut-up
collages with wordplays and anagrammatic juxtapositions.</p>';
      break;
case '126':
      echo '<p><span class="cutups">__</span> It is true what you&#39;re
thinking &mdash; that cut-ups take us rather far afield from <i>ars
combinatoria</i> proper. Text generation through permutation as represented
in Harsd&ouml;rffer&#39;s Denckring is systematic and computational; cutting
and pasting, while combinatory in a general sense, is more of an <i>ars
compilationis</i>. In other words, whereas the former selectively breeds one
organism at a time from an existing genome, the latter dissects organs from
already-formed potentials, sewing them back together to form something wholly
new.</p><p>That is: <i>ars combinatoria</i> is precise and conservative,
operating within a closed network, whereas <i>ars compilationis</i> is open,
incessant and unwieldy, as the scholar carves back the shoots and brambles
that constantly threaten to engulf her work.</p>';
      break;
case '127':
      echo '<p><span class="cutups">__</span> Yet as writing practices,
both <i>ars combinatoria</i> and cut-ups operate through selection, through
picking out discrete elements &mdash; as in <i>legere</i>, to collect,
gather, choose, pick out, read.</p>';
      break;
case '128':
      echo '<p><span class="cutups">__</span> Both <i>ars
combinatoria</i> and what we might call <i>ars compilationis</i> &mdash;
perhaps better situated in the growing histories on commonplacing, notetaking
and marginalia (see, for example, <a href="bibliography.html"
target="_blank">Sherman<span>Sherman, William H. <U>Used Books: Marking
Readers in Renaissance England.</U> Philadelphia, PA: University of
Pennsylvania Press, 1008.</span></a>) &mdash; are literacies: strategies for
consuming, producing and using texts in a language-rich media
environment.</p>';
      break;
case '129':
      echo '<p><span class="computers">__</span><p>nbsp;A growing literature
is exploring how we read and write across the dense network of twenty-first-
century communications technologies (see <a href="bibliography.html"
target="_blank">Coiro et al.<span>Coiro, Julie and Michele Knobel, Colin
Lankshear, and Donald J. Leu, eds. <U>Handbook of Research on New
Literacies.</U> Lawrence Erlbaum Associates/Taylor & Francis Group,
2008.</span></a>). Perhaps not surprisingly, our &quot;new&quot; media
literacies echo rather &quot;old&quot; strategies for manipulating
information. Just as Sir John Gibson cut and copied passages, notes and
references into notebook, we tumble and tweet the flood of net ephemera that
washes over our screens, cutting and pasting found materials into webforms
housed on servers we&#39;ll likely never see. In some ways, the structures we
use, whether a printed commonplace book or a WordPress account, constrain
how, when and where we compile our data; yet, just as Caramuel and
Harsd&ouml;rffer radically altered Ramist word tables into circular databases
to suit their programmatic epistemologies, we tweak Blogger&#39;s codes,
write opensource plug-ins and create our own tags, our own folksonomies.
</p>';
      break;
case '12911':
      echo '<p><span class="cutups">__</span> Cutting, copying, pasting
and combining are literacies: strategies for consuming, producing, and using
texts in a language-rich media environment. It perhaps not surprising that we
```

```
find deep similarities in the literacies demanded of readers at the turn of
the seventeenth century and those at the turn of the twenty-first. Both eras
were and are learning to organize and understand vast amounts of
information.</p>';
      break;
case '130':
      echo '<blockquote><p class="blockquote"><span
class="cutups">__</span> All writing is in fact cut-ups. A collage of
words read heard overheard. What else? Use of scissors renders the process
explicit and subject to extension and variation. (<a href="bibliography.html"
target="_blank">Burroughs<span>Burroughs, William. &quot;The Cut Up
Method.&quot; <U>The Moderns: An Anthology of New Writing in America.</U Ed.
Leroi Jones. New York: Corinth Books, 1963. 345-8.</span></a>
347)</p></blockquote>';
      break;
case '131':
      echo '<blockquote><p class="blockquote"><span
class="cutups">__</span> ALL WRITING IS IN FACT CUT-UPS OF GAMES AND
ECONOMIC BEHAVIOR OVERHEARD? WHAT ELSE? (<a href="bibliography.html"
target="_blank">Burroughs<span>Burroughs, William. &quot;The Cut Up
Method.&quot; <U>The Moderns: An Anthology of New Writing in America.</U Ed.
Leroi Jones. New York: Corinth Books, 1963. 345-8.</span></a>
347)</p></blockquote>';
      break;
case '13111':
      echo '<p><span class="cutups">__</span> Perhaps the most well-
known cut-up artist of the twentieth-century is William S. Burroughs, who
enacted Gysin&#39;s methodologies on his own oeuvre. Slicing and folding his
way through the Word Hoard, a trunk stuffed with his typewritten work,
Burroughs produced a set of three cut-up novels: <i>The Soft Machine</i>
(1961), <i>The Ticket That Exploded</i> (1962) and <i>Nova Express</i>
(1964).</p>';
      break;
case '13122':
      echo '<p><span class="cutups">__</span> Through the physical
manipulation of the page, Burroughs&#39; cut-ups argue that &quot;the Word is
literally a virus&quot; inhabiting its &quot;human host,&quot; serving
&quot;no internal function other than to replicate itself&quot; (<a
href="bibliography.html" target="_blank">Burroughs<span>Burroughs, William S.
<U>The Adding Machine.</U> Arcade Publishing, 1993.</span></a> 48). This is,
Burroughs emphasizes, &quot;not an allegorical comparison&quot; (<a
href="bibliography.html" target="_blank">Burroughs<span>Burroughs, William S.
<U>The Adding Machine.</U> Arcade Publishing, 1993.</span></a> 59): language
physically inhabits the body, forcing us to experience the world through an
ongoing, neverending internal monologue (see <a href="bibliography.html"
target="_blank">Land<span>Land, Christopher. &quot;Apomorphine Silence:
Cutting-up Burroughs&#39; Theory of Language and Control.&quot;
<U>Ephemera</U> 5.3 (2005):450-470.</span></a> 453-456). </p>';
      break;
case '132':
      echo '<blockquote><p class="blockquote"><span
class="cutups">__</span> Quite simply, Burroughs&#39; cut-up project of
the 1960s began as a way to systematize the drive to lose the undesired past,
to cut his way out of an old identity, if not out of identity itself. ... The
ambition to make a complete break, to cut off history and dispossess the
self, to kick the habit of what Walter Benjamin once called &#39;that most
terrible drug &mdash; ourselves &mdash; which we take in solitude&#39;, this
ambition demands discontinuity as a historical as well as a formal principle.
```

```
... The disjunctive and metamorphic juxtapositions that mark his cut-up texts
of the 1960s undo the illusion of fixed identity, and this sabotage
coincided, fully and quite deliberately, with the introduction of a radical
discontinuity in the narrative of his own biographical and literary history.
(<a href="bibliography.html" target="_blank">Harris<span>Harris, Oliver.
<U>William Burroughs and the Secret of Fascination.</u> Carbondale: Southern
Illinois University Press, 2003.</span></a> 8-9)</p></blockquote>';
      break;
case '133':
      echo '<blockquote><p class="blockquote"><span
class="cutups">__</span> Later Burroughs sought a self-abolishing
structure, and tried to defeat our codes of continuity, cultural and
temporal, by shuffling his prose into random order. &quot;Writers until the
cut-up method was made explicit,&quot; he says, &quot;had no way to produce
the accident of spontaneity.&quot; But it seems that in the logic of the
situation we shall find such accidents happy only when we see in them some
allusion, direct or ironical, to our inherited notions of linguistic and
narrative structure ... If Burroughs is a satirist, and he is, then that also
presupposes a past significantly altered. (<a href="bibliography.html"
target="_blank">Kermode<span>Kermode, Frank. <U>The Sense of an Ending:
Studies in the Theory of Fiction.</U> New York: Oxford University Press,
1967.</span></a> 117-8)</p></blockquote>';
      break;
case '134':
      echo '<p><span class="cutups">__</span> The story has been written
so many times; why struggle for new words?</p><blockquote><p
class="blockquote">At the surrealist rally in the 1920s, Tristan Tzara the
man from nowhere proposed to create a poem on the spot by pulling words out
of a hat. A riot ensued wrecked the theatre. Andr&eacute; Breton expelled
Tristan Tzara from the movement and grounded the cut up on the Freudian
couch. (<a href="bibliography.html" target="_blank">Burroughs<span>Burroughs,
William. &quot;The Cut Up Method.&quot; <U>The Moderns: An Anthology of New
Writing in America.</U Ed. Leroi Jones. New York: Corinth Books, 1963. 345-
8.</span></a> 345)</p></blockquote>';
      break;
case '135':
      echo '<p><span class="cutups">__</span> Here&#39;s how Gysin told
the story in 1979:</p><blockquote><p class="blockquote">I always refrained
from ever even reminding Tzara of this painful incidence in the Closerie des
Lilas, where a traditional versifier named Saint-Pol-Roux was being feted as
the Prince of Poets. The loosely amalgamated avant-garde of the time
organized a commando night raid on the celebrated cafe and was routed by the
riot squad. Breton loudly proclaimed that Tzara was a fink: he called the
cops. When the lights went out, Breton heard the voice of Tzara calling the
commissariat on the telephone, distinctly. The commandos chose to believe
him. They went in together: they came out apart. Dada was dead. (Gysin in <a
href="bibliography.html" target="_blank">Zurbrugg<span>Zurbrugg, Nicholas,
ed. <U>Art, Performance, Media: 31 Interviews.</U> Twin Cities: University of
Minnesota Press, 2004.</span></a> 190)</p></blockquote>';
      break;
case '136':
      echo '<blockquote><p class="blockquote"><span
class="cutups">__</span> Tristan Tzara and I used to bump into one
another sometimes in the late &#39;50s around about midnight, for a standup
steak and a beer at the circular zinc counter of the old Royal Saint Germain,
now transmogrified into the monstrous Le Drugstore, where no poets meet who
can help it. Every time we met, Tzara would whine, &quot;Would you be kind
enough to tell me just why your young friends insist on going back over the
```

```
ground we covered in 1920?&amp;quot; What could I say, except, &amp;quot;Perhaps they
feel you did not cover it thoroughly enough.&amp;quot; (Gysin in <a
href="bibliography.html" target="_blank">Zurbrugg<span>Zurbrugg, Nicholas,
ed. <U>Art, Performance, Media: 31 Interviews.</U> Twin Cities: University of
Minnesota Press, 2004.</span></a> 190)</p></blockquote>';
      break;
case '137':
      echo '<p><span class="cutups">__</span>&amp;nbsp;There is yet another
beginning to this story.</p><blockquote><p class="blockquote">One day in late
September 1959, Brion Gysin was in his hotel room, mounting some drawings,
slicing through the boards with his Stanley knife and simultaneously slicing
through the pile of old New York Herald Tribunes he was using to protect his
table. When he finished, he noticed that where a strip of a page was cut
away, the newsprint on the next page lined up and could be read across,
combining stories from different pages, often with hilarious results. Some of
the combinations amused him so much that his neighbors in the hotel knocked
on his door, thinking that his laughter was a hysteria attack. (<a
href="bibliography.html" target="_blank">Miles<span>Miles, Barry. <U>William
Burroughs: el hombre invisible.</U> New York: Hyperion, 1993.</span></a>
111)</p></blockquote>';
      break;
case '138':
      echo '<p><span class="cutups">__</span>&amp;nbsp;In the early 1970s, Tom
Stoppard writes this action into the first scene of <i>Travesties</i>, a play
starring James Joyce, Lenin, Henry Carr and Tristan Tzara. Together, they sit
in a library &amp;quot;occupied with books, papers, pencils . .
.&amp;quot;:</p><blockquote><p class="blockquote">TZARA is writing as the play
begins. On his table are a hat and a large pair of scissors. TZARA finishes
writing, then takes up the scissors and cuts the paper, word by word, into
his hat. When all the words are in the hat he shakes the hat and empties it
on the table. He rapidly separates the bits of paper into random lines,
turning a few over, etc., and then reads the result in a loud voice. (<a
href="bibliography.html" target="_blank">Stoppard<span>Stoppard, Tom.
<U>Travesties.</U> New York: Grove Press, 1975.</span></a> 1-
2)</p></blockquote>';
      break;
case '139':
      echo '<p><span class="cutups">__</span>&amp;nbsp;The hush of the library,
with its categorized tomes tucked carefully on their shelves, shatters with
the sound of Tzara&amp;#39;s voice, chanting gibberish from a loose pile of paper
scraps. Tzara&amp;#39;s act of cut-and-combine is more than a dissembling of the
paper archive: it&amp;#39;s a recombination, compiled from material scraps like
Golem, like the Tower of Babel. Its very existence as such exposes the cut-up
and combinatory nature of all around it, including what is presumed, in the
wagging finger of the librarian, to be sacred.</p><p>No longer a natural or
self-evident system, the archive is exposed for the randomness of its rules
and its arbitrary organization. If the Biblical cut-ups of Jefferson or the
Little Gidding community gave birth to a new, more perfect text, Tzara&amp;#39;s
hat trick is a form of linguistic death &amp;mdash; an exquisite corpse.</p>';
      break;
case '140':
      echo '<blockquote><p class="blockquote"><span
class="cutups">__</span>&amp;nbsp;Like Dali&amp;#39;s, Tzara&amp;#39;s hats are of felt,
with their softly sloping crowns creased along their summits to produce the
parallel lips of the fashionable fedora. A labial crease. A genital smile.
(<a href="bibliography.html" target="_blank">Krauss<span>Krauss, Rosalind.
<U>The Optical Unconscious.</U> Cambridge, MA: MIT Press, 1993.</span></a>
162)</p></blockquote><p>The women of Little Gidding splice together texts
```

```
like DNA in the womb; Tzara reaches into a hat shaped like female genitals to
extract lines of letters.</p><p>Can we gender the act of recombination? Do
women, so often deprived of the pen, make their mark with the cut-ups of
<i>ars combinatoria</i>?</p><p>And if so, what does Tzara&#39;s distinctly
male act &mdash; described as the moment that gave birth to Dada &mdash;
signal for a movement so distinctly homosocial and masculine (see <a
href="bibliography.html" target="_blank">Hopkins<span>Hopkins, David.
<U>Dada&#39;s Boys: Masculinity after Duchamp.</U> New Haven, CT: Yale
University Press, 2007.</span></a>)?</p>';
      break;
case '141':
      echo '<p><span class="cutups">__</span> Like his friend Tzara
before him, Brion Gysin &mdash; William Burroughs&#39; colleague and, in
Burroughs&#39; words, &quot;the only man that I&#39;ve ever respected&quot;
(Burroughs in <a href="bibliography.html"
target="_blank">Zurbrugg<span>Zurbrugg, Nicholas, ed. <U>Art, Performance,
Media: 31 Interviews.</U> Twin Cities: University of Minnesota Press,
2004.</span></a> 86) &mdash; experimented with cut-ups, slicing not only
pages but magnetic tape and film.</p><p>Many of his poems operate through
permutation, as in: &quot;Come to Free the Words,&quot; &quot;Too Free Come
the Words,&quot; &quot;The Words Come Too Free&quot; (quoted in <a
href="bibliography.html" target="_blank">Zurbrugg<span>Zurbrugg, Nicholas,
ed. <U>Art, Performance, Media: 31 Interviews.</U> Twin Cities: University of
Minnesota Press, 2004.</span></a> 191); others juxtapose sound, text and
image in a spontaneous, multimodal performance. He was, as he later described
it, &quot;attempting to show that these techniques could be decanted back and
forth through different media&quot; (Gysin in <a href="bibliography.html"
target="_blank">Zurbrugg<span>Zurbrugg, Nicholas, ed. <U>Art, Performance,
Media: 31 Interviews.</U> Twin Cities: University of Minnesota Press,
2004.</span></a> 192).</p>';
      break;
case '142':
      echo '<blockquote><p class="blockquote"><span
class="cutups">__</span> There was a whole muggy area of doubt as to
what to call this monstrosity I helped bring about. Because of the machines
involved, the electronics, I would have called it &quot;machine poetry,&quot;
but everyone shied away from that. I felt good about creating through the
machines, and they did not. I wanted to make language work in a new way, to
surprise its secrets by using it as the material one passed through the
available electronics to amplify the voices of poetry. ... I wanted to get as
far away as possible from &quot;inspiration.&quot; I wanted expiration
instead, to breathe out rather than in. (Gysin quoted in <a
href="bibliography.html" target="_blank">Zurbrugg<span>Zurbrugg, Nicholas,
ed. <U>Art, Performance, Media: 31 Interviews.</U> Twin Cities: University of
Minnesota Press, 2004.</span></a> 194)</p></blockquote>';
      break;
case '143':
      echo '<p><span class="computers">__</span><p>nbsp;In 1960,
mathematician Ian Sommerville and Brion Gysin collaborated on automating the
process of permuting the phrase &quot;I AM THAT I AM.&quot; Using a computer,
Gysin and Sommerville programmed a simple algorithm that would print out all
120 permutations of the phrase organized into four text blocks (see <a
href="bibliography.html" target="_blank">Funkhouser<span>Funkhouser, Chris.
<U>Prehistoric Digital Poetry: An Archaeology of Forms, 1959-1995.</U>
Tuscaloosa: University of Alabama Press, 2007.</span></a> 39, <a
href="bibliography.html" target="_blank">Kuri<span>Kuri, Jos&eacute;
F&eacute;rez, ed. <U>Brion Gysin: Tuning in to the Multimedia Age.</U>
London: Thames & Hudson, 2003.</span></a> 96-7).</p>';
```

```
      break;
case '144':
      echo '<p><span class="computers">__</span><p>nbsp;Florian Cramer links
computer programming and the cut-up experiments of Gysin, Burroughs and
Sommerville to mysticism and magic, all of which rely on the formal execution
of linguistic commands. &quot;Spoken by the author on the tape
recording,&quot; Cramer writes, Gysin&#39;s permutations of &#39;IN THE
BEGINNING WAS THE WORD&#39; &quot;were not solely mathematical computations,
but also incantations&quot; (<a href="bibliography.html"
target="_target">Cramer<span>Cramer, Florian. <U>Words Made Flesh: Code,
Culture, Imagination.</U> Piet Zwart Institute, 2005.</span></a> 17). Thus
just as ritual provides a structure to order the seemingly random chaos of
the world, programming ritualizes language itself, creating a controlled and
yet infinitely expandable environment for contemplating our experience as
linguistically embodied selves.</p>';
      break;
case '146':
      echo '<p><span class="computers">__</span><p>nbsp;Importantly and
helpfully, Cramer&#39;s cultural history of computing embeds material
technologies in an immaterial context, teasing out the subtle connections
between software and the imagination. Yet the practice of cut-ups and <i>ars
combinatoria</i> is more than the manipulation of symbols through matter; it
is a complete materializing of language itself.</p>';
      break;
case '147':
      echo '<p><span class="cutups">__</span> By embodying itself as a
running mental monologue, language &mdash; particularly after the advent of
written text &mdash; frames existence as a linear narrative for Burroughs.
Land explains:</p><blockquote><p class="blockquote">Burroughs is suggesting
that the word-virus operates as an external &#39;other&#39; that colonizes
the body, forcing it to sub-vocalize and thereby reproducing itself. It is
this internal monologue, all but impossible to shut off and expressly non-
human, which produces an all-too-human sense of identity and self-continuity;
generating a linear, narrative time along which experience is distributed and
through which identity is assured. (<a href="bibliography.html"
target="_blank">Land<span>Land, Christopher. &quot;Apomorphine Silence:
Cutting-up Burroughs&#39; Theory of Language and Control.&quot;
<U>Ephemera</U> 5.3 (2005):450-470.</span></a> 455)</p></blockquote>';
      break;
case '14711':
      echo '<p><span class="cutups">__</span> Which is to say: cut-ups
cut-up control. Cut-ups slice through the flat sheet of a linear reality to
give the page new dimensions, new depths. In doing so, cut-ups also free the
poet &mdash; a necessarily linguistic being always/already inhabited by the
word virus &mdash; from the parasite of narrative language, helping her to
develop new linguistic sense perceptions.</p>';
      break;
case '14722':
      echo '<p><span class="cutups">__</span> Yet written language can
never be wholly freed from its material state. Ironically, just as Burroughs
disembodies the word virus, it attaches itself again to the page, clinging to
the space where pools of ink sink into the fibers of the paper. In short,
cut-ups can only extract obsessive bookishness from the human body by drawing
attention to embodied state of any book, any text, which only exists through
the page itself.</p><p>Thus being an active reader frees the writer to write;
while merely&quot;writing&quot; &mdash; that is, manipulating the text as
inscribed matter &mdash; allows the reader access to the text as a
creation.</p>';
```

```
      break;
case '148':
      echo '<p><span class="cutups">__</span> Thus just as
Harsd&ouml;rffer mechanizes a materialist view of language in his Denckring,
so Burroughs&#39; theory of cut-ups treats language as always/already
embodied, refusing to lock it in an immaterial mind.</p><p>From this angle,
Burroughs&#39; linguistic theory &mdash; like the history of <i>ars
combinatoria</i> in general &mdash; might be read as a reaction against
contemporary movements to scientize the study of language. As the cognitive
revolution began locating linguistic meaning in syntactic relationships and
mental processes, cut-up artists refused to dematerialize and disembody the
word. For them, the relationship between human and text was not that of
process and product, but a palimpsest of media forms that inscribed
themselves onto and through the body.</p>';
      break;
case '14811':
      echo '<blockquote><p class="blockquote"><span
class="cutups">__</span> We began to find out a whole lot of things
about the real nature of words and writing. What are words and what are they
doing? The cut-up method treats words as the painter treats his paint, raw
material with rules and reasons of its own . . . if you want to challenge and
change fate . . . cut up words, make them a new world. (Gysin, quoted in <a
href="bibliography.html" target="_blank">Miles<span>Miles, Barry. <U>William
Burroughs: el hombre invisible.</U> New York: Hyperion, 1993.</span></a>
130)</p></blockquote>';
      break;
case '14822':
      echo '<blockquote><p class="blockquote"><span
class="cutups">__</span> My ambition was to destroy the assumed natural
links of language, that in the end are but expressions of Power, the
favourite weapon of control or even the essence of control. (Gysin, quoted in
<a href="bibliography.html" target="_blank">Kuri<span>Kuri, Jos&eacute;
F&eacute;rez, ed. <U>Brion Gysin: Tuning in to the Multimedia Age.</U>
London: Thames & Hudson, 2003.</span></a> 164)</p></blockquote>';
      break;
case '14833':
      echo '<blockquote><p class="blockquote"><span
class="permutation">__</span> The permutated poems set the words
spinning of on their own; echoing out as the words of a potent phrase are
permutated into an expanding ripple of meanings which they did not seem to be
capable of when they were struck and then stuck into that phrase. (Gysin in
<a href="bibliography.html" target="_blank">Kuri<span>Kuri, Jos&eacute;
F&eacute;rez, ed. <U>Brion Gysin: Tuning in to the Multimedia Age.</U>
London: Thames & Hudson, 2003.</span></a> 154)</p></blockquote>';
      break;
case '149':
      echo '<p><span class="cutups">__</span> Of course, it would be
disingenuous not to point out Burroughs&#39; later boredom with the whole
project. As he said of cut-ups in a 1983 interview with Nicholas
Zurbrugg,</p><blockquote><p class="blockquote">Well, I haven&#39;t done any
of those in years, but they were interesting. They explored the whole matter
of synchronicity. You can record in the street, and you can take something
that you&#39;ve recorded and play it back in the street, and you observe all
sorts of synchronicities. It just makes you aware of certain things going on
all the time anyway. (Burroughs in <a href="bibliography.html"
target="_blank">Zurbrugg<span>Zurbrugg, Nicholas, ed. <U>Art, Performance,
Media: 31 Interviews.</U> Twin Cities: University of Minnesota Press,
```

```
2004.</span></a> 70)</p></blockquote><p>And thus the cutting edge becomes
&quot;interesting&quot; &mdash; simply something to do.</p>';
      break;
case '150':
      echo '<p><span class="computers">__</span><p>nbsp;As Funkhouser points
out, Gysin and Sommerville&#39;s experiment was not the first to use
computers to automatically generate lines of verse. In 1959, Theo Lutz used a
Zuse Z22 computer to &quot;take over the laborious production of stochastic
texts,&quot; substituting the effort of, for instance, throwing dice with a
&quot;program-controlled data processor&quot; to generate random numbers (<a
href="bibliography.html" target="_blank">Lutz<span>Lutz, Theo.
&quot;Stochastische Texte.&quot; <U>augenblick</u> 4 (1959): 3-9.</span></a>
<a href="http://www.stuttgarter-schule.de/lutz_schule_en.htm">online</a>).
Here&#39;s how it works:</p><blockquote><p class="blockquote">A new number is
formed from an initial number by an arithmetic operation, and from this
number digits are taken by intersection, which are then considered to be a
random number. The number generated by this operation is the initial number
to determine the next random number. By continuing this process, a sequence
of numbers is obtained. (<a href="bibliography.html"
target="_blank">Lutz<span>Lutz, Theo. &quot;Stochastische Texte.&quot;
<U>augenblick</u> 4 (1959): 3-9.</span></a> <a href="http://www.stuttgarter-
schule.de/lutz_schule_en.htm">online</a>)</p></blockquote>';
      break;
case '151':
      echo '<blockquote><p class="blockquote"><span
class="computers">__</span><p>nbsp;The machine stores a certain number of
subjects, predicates, logical operators, logical constants and the word
&quot;IST&quot; (engl.: &quot;IS&quot;), coded as binary numbers. Using the
first random number the machine forms the address (i.e. the position number
in the store) of a subject by adding a constant which the machine now has at
its disposal. In the one-following memory cell, the program locates a code
number which it evaluates as gender of the subject in question, e.g. 0=
masculine, 1= feminine and 2 = neutral. The machine then determines a logical
operator using a new random number and coordinates this with the gender of
the subject, using the located code number. At this stage a print-out is done
for the first time e.g. the teleprinter prints:</p><p>NICHT JEDER BLICK<br
/>(engl.: NOT EVERY LOOK) (<a href="bibliography.html"
target="_blank">Lutz<span>Lutz, Theo. &quot;Stochastische Texte.&quot;
<U>augenblick</u> 4 (1959): 3-9.</span></a> <a href="http://www.stuttgarter-
schule.de/lutz_schule_en.htm">online</a>)</blockquote>';
      break;
case '152':
      echo '<p><span class="computers">__</span><p>nbsp;To computationally
compose a poem, Lutz took Franz Kafka&#39;s <i>The Castle</i> as his source
text, cutting the text into sixteen titles and subjects that were then stored
in a database. Much like the roll of a dice, the numbers randomly generated
by the program called the titles and subjects from the database, then
organized them into a syntax according to predetermined logical constraints.
The result is, as Funkhouser describes it, full of &quot;discursive leaps and
quirky, unusual semantic connections&quot; that force the reader to
&quot;connect and interpret abstractions in the poem ... and derive meaning
from the verbal associations while reading the text in and against its
context&quot; (<a href="bibliography.html"
target="_blank">Funkhouser<span>Funkhouser, Chris. <U>Prehistoric Digital
Poetry: An Archaeology of Forms, 1959-1995.</U> Tuscaloosa: University of
Alabama Press, 2007.</span></a> 37-8).</p>';
      break;
case '153':
```

```
      echo '<blockquote><p class="blockquote"><span
class="computers">__</span><p>nbsp;Lutz&#39;s selection of words, combined
with his programming method, enables a speculative, self-reflexive,
unconventional style of expression; the programming method consists of about
fifty commands and could theoretically generate more than four million
different sentences. (<a href="bibliography.html"
target="_blank">Funkhouser<span>Funkhouser, Chris. <U>Prehistoric Digital
Poetry: An Archaeology of Forms, 1959-1995.</U> Tuscaloosa: University of
Alabama Press, 2007.</span></a> 38)</p></blockquote>';
      break;
case '154':
      echo '<p><span class="computers">__</span><p>nbsp;More recently, the
cut-up method has been re-packaged as Flarf, a form of poetry composed by
(among other things) plucking snatches of text from Google searches to
combine into a poem.</p><p>Flarf is <a
href="http://www.brooklynrail.org/2009/02/books/flarf-from-glory-days-to-
glory-hole" target="_blank">energetic</a>. Flarf is <a
href="http://possumego.blogspot.com/2009/01/flarf-and-shit-and-fake-
shit.html" target="_blank">shit</a>. Flarf is <a
href="http://jacketmagazine.com/29/hoy-flarf.html" target="_blank">naive</a>.
Flarf is <a href="http://lime-tree.blogspot.com/2009/02/dale-smith-deals-
death-blow-to-flarf.html" target="_blank">dead</a>.</p><p>And perhaps the
only objectively true statement: Flarf is controversial.</p>';
      break;
case '155':
      echo '<p><span class="computers">__</span><p>nbsp;In an April 2006
edition of Jacket magazine, Dan Hoy chides Flarfists for not recognizing that
&quot;there is always a power determining orthodoxy in any &#39;random&#39;
generator&quot; &mdash; particularly in a corporate entity like Google. Hoy
also accuses Flarf of ignoring its Dadaist predecessors (<a
href="bibliography.html" target="_blank">Hoy<span>Hoy, Dan. &quot;The Virtual
Dependency of the Post-Avant and the Problematics of Flarf: What Happens when
Poets Spend Too Much Time Fucking Around on the Internet.&quot; <U>Jacket</U>
29 (April 2006): online.</span></a> <a
href="http://jacketmagazine.com/29/hoy-flarf.html"
target="_blank">online</a>).</p>';
      break;
case '156':
      echo '<p><span class="computers">__</span><p>nbsp;I&#39;m not
interested in debating the relative merits of Flarf. For our purposes,
it&#39;s enough to recognize how Flarf has reinvigorated debate around the
issue of randomness and chance in poetry &mdash; and reintroduced the
question of code from a new angle. Thus far, most of the poems or volvelles
or cut-ups I have discussed emerge directly from some form of code practice
&mdash; whether a procedure such as the exquisite corpse, a paper mechanism
or a text-generating Perl poem. Yet, as Hoy argues, Google&#39;s obscure,
closely-guarded search algorithms may influence the underlying ideology of a
poetic practice without an artist even knowing how or why. What writer with a
computer and access to the internet hasn&#39;t used a search engine to skim
across the surface of the web&#39;s &quot;bottomless ocean &mdash; all
potentiality&quot; (<b>Sokei-an</b>)? Whose &quot;formless mind&quot; has
been left untouched by the new tools that surround us?</p>';
      break;
case '157':
      echo '<p><span class="computers">__</span><p>nbsp;Just as Tzaras&#39;s
hat trick forces Stoppard to rethink the archive; just as
Harsd&ouml;rffer&#39;s Denckring opens a new relationship between poetry and
the vernacular; just as Jefferson cuts up his Bible to reconstitute religion,
```

```
so Flarf invites to consider how algorithmic media infiltrate the poetic
mechanism. Maybe this is the ideological redemption Flarf&#39;s critics keep
hoping to find behind its patent absurdism; but most likely not. Like other
acts of <i>ars combinatoria</i>, Flarf&#39;s selective cut-and-recombine
methods tell us more about the information age in which we live than it does
about poetry. When overload impedes imagination, spin the wheel, search the
web.</p>';
      break;
case '158':
      echo '<p><span class="computers">__</span><p>nbsp;In <a
href="http://pangrammaticon.blogspot.com/2006/02/against-generative-grammar-
of-flarf.html" target="_blank">a comment appended to Thomas
Basb&oslash;ll&#39;s response</a>, Anne Boyer writes that &quot;Hoy misses
the mark &mdash; the progenitor of Flarf isn&#39;t Cage, or Queneau, but
Baroness von Else Freytag Loringhoven.&quot; Although written out of most
histories of modernism, Freytag-Loringhoven was a well-known female avant-
garde poet and subject of a recent feminist re-reading of Dada by Amelia
Jones.</p><p>If Dada began with Tzara pulling cut-up words from a yonic hat,
what does it mean that the most hated cut-up poetry movement of the twenty-
first century identifies with the Baroness, rather than Cage or
Queneau?</p>';
      break;
case '159':
      echo '<blockquote><p class="blockquote"><span
class="computers">__</span><p>nbsp;To use and/or promote a search engine
without question is an implicit acceptance of it as an arbiter of relevance.
This is like compiling a list of what&#39;s going on in the world from only
the U.S. network TV news shows, without acknowledging the biases inherent in
their selection processes. Google is not the zeitgeist, nor is it an
indifferent and all-inclusive database of it. Google, as a generator, is a
corporate algorithm that ranks webpage relevancy (from a limited cross-
section &mdash; i.e. what it bothers to index &mdash; of an already limited
segment of the zeitgeist &mdash; i.e. Internet users and web content) based
on its own idiosyncratic definition of &#39;relevancy&#39; and, in tandem
with corporate web designers, manipulation of the results. (<a
href="bibliography.html" target="_blank">Hoy<span>Hoy, Dan. &quot;The Virtual
Dependency of the Post-Avant and the Problematics of Flarf: What Happens when
Poets Spend Too Much Time Fucking Around on the Internet.&quot; <U>Jacket</U>
29 (April 2006): online.</span></a> <a
href="http://jacketmagazine.com/29/hoy-flarf.html"
target="_blank">online</a>)</p></blockquote>';
      break;
case '161':
      echo '<blockquote><p class="blockquote"><span
class="permutation">__</span> In the beginning was the word. Everything
seems to be wrong with what was produced from those beginnings, so let&#39;s
rub out the word and start afresh ... if the whole thing began with the word,
well then, if we don&#39;t like what was produced, and we don&#39;t,
let&#39;s get right to the root of the matter and radically alter it. (Gysin,
quoted in <a href="bibliography.html" target="_blank">Kuri<span>Kuri,
Jos&eacute; F&eacute;rez, ed. <U>Brion Gysin: Tuning in to the Multimedia
Age.</U> London: Thames & Hudson, 2003.</span></a> 130)</p>';
      break;
case '162':
      echo '<p><span class="permutation">__</span> <b>To rub out</b>: a
verb phrase meaning both to erase, eradicate, remove the word. By permuting
the phrase to the point of absurdity in his programmed permutation poem
&#39;RUB OUT THE WORD&#39;, Gysin extracts the (as Burroughs puts it)
```

```
&quot;word virus&quot; from the human, rendering meaning as nonsense, and
written language as no more than set of arbitrary symbols.</p><p>Yet rub out
also carries a sense of creation, construction &mdash; as in to &quot;rub the
text out of a slab of marble,&quot; removing the negative space around
letters to bring a text to the surface. Thus to rub out the word is both to
raze it and to raise it, wiping the surface clean even as one constructs a
new reality.</p>';
      break;
case '163':
      echo '<p><span class="permutation">__</span> Thus Gysin permutes
The Word to break outside an all-encompassing monotheistic system &mdash;
indeed, to break outside of Meaning itself &mdash; but ends up reinstating
another form of mystical creation: producing the infinite from the finite, or
(to return to Leibniz&#39;s binary system) 1 from 0.</p><p>In this way, like
Llull&#39;s machines of conversion, the material manipulation of symbols
spins, and spins, and spins until language takes flight, no longer anchored
to any totalizing linguistic system but becomes pure sound &mdash; a pure,
pre-Babel utterance. Writing is not a process of making meaning, then, but of
destroying meaning; and reading is precisely what we typically conceive of as
writing &mdash; an attempt to find sense in the chards.</p>';
      break;
case '164':
      echo '<p><span class="permutation">__</span> Perhaps mystic
Quirinus Kuhlmann and Brion Gysin share more than either might have
imagined.</p>';
      break;
 case '170':
      echo '<p><span class="computers">__</span><p>nbsp;As Lutz&#39;s work
shows, even the earliest use of electronic digital computers treated writing
as, to borrow a term from Kenneth Goldsmith, a fundamentally
&quot;uncreative&quot; act of cutting up and recombining existing texts..
Indeed, the very act of programming &mdash; of building a model that (at
least in much digital poetry) calls from and manipulates data stored in a
database &mdash; engages writing as a form of algorithmic generation (see <a
href="bibliography.html" target="_blank">Kirschenbaum<span>Kirschenbaum,
Matthew. &quot;Hello Worlds.&quot; <U>The Chronicle Review.</u> January 23,
2009: online.</span></a>). Likewise, from their earliest conception digital
poems questioned the traditional role of the reader. As Charles O. Hartman
writes, as the reader becomes more important, poets begin to &quot;think of
moving him or her away from that end-of-the-road box in the diagrams and back
into the process somewhere,&quot; thereby making &quot;the reader&#39;s
constructive role in the poem more conscious&quot; (<a
href="bibliography.html" target="_blank">Hartman<span>Hartman, Charles O.
<U>Virtual Muse: Experiments in Computer Poetry</U>. Hanover, N.H.: Weleyan
University Press, 1996.</span></a> 105).</p>';
      break;
 case '171':
      echo '<blockquote><p class="blockquote"><span
class="computers">__</span><p>nbsp;The phenomenology of reading that emerges
from them shifts away from the sense of reading and writing as the only two
directions of a highway into and out of the soul, and adds a very tangible
third person plural. Constraints of keyboard, mouse, and screen are
experienced as resistances from a machine intelligence governed by a social
network whose horizon is out of sight, and these resistances are felt as both
gateways and interfaces between body and machine. Text no longer projects as
a bounded material object; it is more like a landscape of which only part is
visible due to the limits of the gaze. Text unrolls down the screen in a
manner not unlike the road or track unrolling before the gaze in a computer
```

```
simulation of driving, and that is based on the framing of landscape by a
windshield, so we might be tempted to wonder if the experience of reading has
more affinity with driving than before. (<a href="bibliography.html"
target="_blank">Middleton<span>Middleton, Peter. <U>Distant Reading:
Performance, Readership, and Consumption in Contemporary Poetry.</u>
Tuscaloosa: University of Alabama Press, 2005.</span></a> 145-
6)</p></blockquote>';
      break;
 case '172':
      echo '<p><span class="computers">__</span><p>nbsp;Funkhouser has
identified a number of other early experiments in digital poetry: Auto-
Beatnik, a computer programmed to spit out beatnik-y verse (1962); Jackson
Mac Low&#39;s programmed film reader which randomly permuted messages (1969);
Angel Carmona&#39;s &quot;V2 Poems&quot; and Alan Sondheim&#39;s TI59
calculator poems (both 1970s). As should be evident, many of the early works
were generative and combinatory. Thus, just as Lull&#39;s arguments exist in
the nexus between his combinatory Figures and their constants, the poetry in
these experiments lies in the interaction between a designed program and a
word list that, together, produce what appears to be a traditional
poem.</p>';
      break;
 case '173':
      echo '<p><span class="computers">__</span><p>nbsp;Here, it&#39;s worth
pausing over one oft-cited early computer program: TRAVESTY, a text generator
written in Pascal and developed in 1984 by literary critic Hugh Kenner and
computer scientist Joseph O&#39;Rourke. Designed as an algorithm for
analyzing, manipulating, and (re)producing text, TRAVESTY takes a source
text, divides it into character strings (or letters) of length <i>n</i>, then
produces a statistical frequency table for any given set of <i>n</i>
characters. This table is  used to produce another text with the same
statistical properties. Or, to use a noncomputational metaphor, TRAVESTY
eats, digests and regurgitates any source text provided by its user.</p>';
      break;
 case '174':
      echo '<p><span class="computers">__</span><p>nbsp;When <i>n</i> = 1,
the output text is almost unreadable, each individual letter having been
scrambled. However, when <i>n</i> = 9, the longest string possible in the
program, the output text appears almost entirely the same as its input.
Here&#39;s an example from Hartman before and after an <i>n</i> = 9
travesty:</p><blockquote><p class="blockquote">Dead flies cause the ointment
of the apothecary to send forth a stinking savor: so doth a little folly him
that is in reputation for wisdom and honour.<br/>Dead flies cause the
ointment of the rule: folly is set in great dignity, and the end of his mout
is foolishness: and the end of his talk is mischievous madness. (<a
href="bibliography.html" target="_blank">Hartman<span>Hartman, Charles O.
<U>Virtual Muse: Experiments in Computer Poetry</U>. Hanover, N.H.: Weleyan
University Press, 1996.</span></a> 55)</p></blockquote>';
      break;
 case '175':
      echo '<blockquote><p class="blockquote"><span
class="computers">__</span><p>nbsp;Travesty [an early text-generating
computer program] offers ... the wickedness of exploding revered literary
scripture into babble. We can reduce Dr. Johnson to inarticulate imbecility,
make Shakespeare talk very thickly through his hat, or exhibit Francis Bacon
laying waste his own edifice of logic amid the pratfalls of <i>n</i> = 9. Yet
the other side of the coin is a kind of awe. Here is language creating itself
out of nothing, out of mere statistical noise. (<a href="bibliography.html"
target="_blank">Hartman<span>Hartman, Charles O. <U>Virtual Muse: Experiments
```

```
in Computer Poetry</U>. Hanover, N.H.: Weleyan University Press,
1996.</span></a> 57)</p></blockquote>';
      break;
 case '176':
      echo '<p><span class="computers">__</span><p>nbsp;By conceptualizing
writing as a process of analysis and manipulation, TRAVESTY reduces meaning
to a user-determined statistic, the numerical primitive long-sought by
Leibniz. At the same time, the program&#39;s output continually slides along
the rim between sense and nonsense, between language-as-meaning and language-
as-data; in this way, much like the cut-ups of Tristan Tzara, TRAVESTY forces
the always already literate reader to question the text&#39;s
&quot;authenticity.&quot; Thus TRAVESTY represents a return to an unfulfilled
promise of the seventeenth-century programmatic epistemology even as it
participates in the postmodern politics of twentieth-century procedural art,
finding wholeness from plurality while exploding any sense of the
&quot;natural&quot; in &quot;natural language.&quot;</p>';
      break;
 case '177':
      echo '<p><span class="computers">__</span><p>nbsp;At about <i>n</i> =
3, the early text-generating computer program TRAVESTY produces phrases like:
&quot;the thy hedge, afte se the whatter&quot; &mdash; in other words,
nonsense, but, as Hartman puts it, &quot;clearly <i>English</i>
nonsense&quot; (<a href="bibliography.html"
target="_blank">Hartman<span>Hartman, Charles O. <U>Virtual Muse: Experiments
in Computer Poetry</U>. Hanover, N.H.: Weleyan University Press,
1996.</span></a> 56). According any standard lexicon (including the <i>Oxford
English Dictionary</i>), &#39;whatter&#39; is not a word; yet it follows
standard English rules for word formation &mdash; the &#39;h&#39; follows the
&#39;w&#39;, &#39;er&#39; is a common ending &mdash; and so seems like it
<i>could</i> be.</p><p>Thus, much like Leibniz&#39;s alphabet or
Harsd&ouml;rffer&#39;s Denckring, the TRAVESTY program treats written
language as a set of rule-driven primitives and, in doing so, transforms a
closed lexicon of isolated words into an open, infinitely-regenerative
algorithm of linguistic possibilities.</p>';
      break;
 case '178':
      echo '<p><span class="computers">__</span><p>nbsp;Hartman subsequently
used TRAVESTY to compose poems. He began by writing several of his own
verses, then feeding them to the program using 8 different <i>n</i> values
(all but <i>n</i> = 1). These eight outputs he organized into a loose
structure, then manipulated according to his own intuitive sense of the most
&quot;superior&quot; output (see <a href="bibliography.html"
target="_blank">Hartman<span>Hartman, Charles O. <U>Virtual Muse: Experiments
in Computer Poetry</U>. Hanover, N.H.: Weleyan University Press,
1996.</span></a> 62-4). The result is his &quot;Monologues of Soul and
Body.&quot;</p>';
      break;
 case '179':
      echo '<blockquote><p class="blockquote"><span
class="computers">__</span><p>nbsp;Thinking up ways to &#39;do computer
poetry&#39; makes us look at language and poetry from an unfamiliar angle.
This seems appropriate if one of poetry&#39;s functions is to make us aware,
with a fresh intensity, of our relation to the language that constitutes so
much of human life &mdash; or if you like, of how language constitutes so
much of our relation to the world. How do words mean when we put them into
new contexts? Under what conditions does the meaning web tear apart? What
meanings can words make (or can we make of them) when we disturb their normal
relation to each other? These are questions that any poet can be seen as
```

```
asking; and from this point of view, someone experimenting with computer
poetry is continuing an age-old project. (<a href="bibliography.html"
target="_blank">Hartman<span>Hartman, Charles O. <U>Virtual Muse: Experiments
in Computer Poetry</U>. Hanover, N.H.: Weleyan University Press,
1996.</span></a> 104)</p></blockquote>';
      break;
 case '180':
      echo '<p><span class="computers">__</span><p>nbsp;As procedural works,
early digital poems are often historicized &mdash; if they are historicized
at all &mdash; against the modernist movements of the twentieth century,
particularly dada, Fluxus, the Oulipo and cut-up methods. Thus for Glazier,
the &quot;post-typographic and nonlinear disunion&quot; of digital poetry
emerges from &quot;Futurism, Gertrude Stein, and Dada,&quot; among others (<a
href="bibliography.html" target="_blank">Glazier<span>Glazier, Loss Peque
      &ntilde;o. <U>Digital Poetics: The Making of E-Poetries.</U>
Tuscaloosa, AL: University of Alabama Press, 2002.</span></a> 35); for
McGann, Mallarm&eacute;&#39;s <i>Un coup de d&eacute;s</i> comes to
&quot;forecas[t] key terms by which we now characterize digital media&quot;
(<a href="bibliography.html" target="_blank">McGann<span>McGann, Jerome J.
<U>Radiant Textuality: Literature After the World Wide Web.</U> New York:
Palgrave, 2001.</span></a> 210); and electronic literatures are theorized
against their &quot;print precursors&quot; (<a href="bibliography.html"
target="_blank">Sloane<span>Sloane, Sarah. <U>Digital Fictions: Storytelling
in a Material World.</U> New York: Greenwood Publishing Group,
2000.</span></a> 44).</p>';
      break;
 case '181':
      echo '<p><span class="computers">__</span><p>nbsp;Yet  modeling digital
poetry on modernist movements presents an interesting dilemma &mdash; what
Sandy Baldwin calls the &quot;paradox of innovation.&quot; For under this
historical regime, &quot;the &#39;new&#39;-ness of literary innovation occurs
against the background of a tradition that novelty ends up reinforcing&quot;
(<a href="bibliography.html" target="_blank">Baldwin<span>Baldwin, Sandy.
&quot;A Poem is a Machine to Think With: Digital Poetry and the Paradox of
Innovation.&quot; Review of Loss Peque&ntilde;o Glazier, <U>Digital Poetics:
The Making of E-Poetries.</U> <U>Postmodern Culture</U> 13:2 (January 2003).
Online.</span></a> <a href="http://www.iath.virginia.edu/pmc/text-
only/issue.103/13.2baldwin.txt" target="_blank">online</a>). In other words,
situating digital poetry within an always already avant-garde politics
immediately renders it stale, uninteresting &mdash; merely an electronic
iteration of procedural poems already produced on paper. In this model, the
only thing &quot;new&quot; about digital poetry is the technology
itself.</p>';
      break;
 case '182':
      echo '<p><span class="computers">__</span><p>nbsp;On the flip side,
digital poetry is sometimes imagined as a radical rupture from past practices
precisely <i>because</i> of its technological newness. Thus Alan Sondheim
writes that &quot;for thousands of years, writers have, again in general,
taking their tools &mdash; taken writing itself &mdash; for granted. Even
Sterne and Carroll work within traditional means. The computer and the
Internet, however, have opened up a whole (and indefinable) world of
possibilities&quot; (<a href="bibliography.html"
target="_blank">Sondheim<span>Sondheim, Alan. &quot;Introduction:
Codework.&quot; <U>ABR</U> 22.6 (September/October 2001).</span></a>
1).</p>';
      break;
 case '183':
```

```
      echo '<p><span class="computers">__</span><p>nbsp;Yet by focusing on
technological rather than aesthetic &quot;newness,&quot; Sondheim goes on to
describes these &quot;vast uncharted domains ... of new and future
literatures&quot; (<a href="bibliography.html"
target="_blank">Sondheim<span>Sondheim, Alan. &quot;Introduction:
Codework.&quot; <U>ABR</U> 22.6 (September/October 2001).</span></a> 2)
&mdash; what he calls <i>codework</i> &mdash; in terms that could easily
apply to Harsd&ouml;rffer&#39;s Denckring or Caramuel&#39;s Maria Stella .
Just as the paradox of innovation destroys the avant-garde, so does an
obsession with newness reduce all literature to an outcome of its
technological functionalities &mdash; in many cases, functionalities
ironically found across multiple platforms.</p>';
      break;
 case '184':
      echo '<p><span class="computers">__</span><p>nbsp;Importantly, my
critique is not of these histories themselves, but of their deployment.
Certainly dada deeply influenced many digital artists and programmars;
likewise it would be facile to posit too deep of a connection between, say,
Caramuel&#39;s Maria Stella and the TRAVESTY program. Rather, presenting
digital poetry as radically new or even procedurally modernist overemphasizes
media technologies at the expense of what Erkki Huhtamo, following Ernst
Robert Curtius, calls <i>media topoi</i>, or &quot;cyclically recurring
elements and motives underlying and guiding the development of media
culture&quot; and, in particular, literacy (<a href="bibliography.html"
target="_blank">Huhtamo<span>Huhtamo, Erkki. &quot;From kaleidoscomaniac to
cybernerd: Notes toward an archaeology of the media.&quot; <U>Electronic
Culture: Technology and Visual Representation.</U> Ed. by Timothy Druckrey.
London: Aperture Foundation, 1996.</span></a> 301).</p><p>What if, instead of
linking &quot;digital poems&quot; to modernist &quot;procedural poems,&quot;
we broadened our investigation to combinatory reading and writing practices
across multiple material and aesthetic platforms?</p>';
      break;
 case '185':
      echo '<p><span class="computers">__</span><p>nbsp;To understand
Harsd&ouml;rffer&#39;s Denckring in relation to an electronically-mediated
work, we must take a radically anti-evolutionary approach to poetry.
Literatures do not grow and then die; nor do they compete among each other in
a race for the best environmental fit. The process of historicizing a set of
practices demands that we explore their underlying epistemologies, both
poetic and philosophical, as they have been embedded in very different media
machines. </p>';
      break;
 case '186':
      echo '<p>In short, this machine acts as what Hayles calls a
&quot;technotext&quot;: it constantly &quot;interrogates the inscription
technology that produces it&quot; and, in doing so, &quot;mobilizes reflexive
loops between its imaginative world and the material apparatus embodying that
creation as a physical presence&quot; (Hayles 25). Hayles distinguishes her
term from Espen Aarseth&#39;s widely-used &quot;cybertext&quot; &mdash;
defined by Aarseth as a &quot;machine for the production of variety of
expression&quot; (Aarseth 3) &mdash; by noting that it overemphasizes user
interaction, &quot;connot[ing] a functional and semiotic approach that
emphasizes a computational perspective&quot; (Hayles 29). By contrast,
technotexts engages its own materiality in a variety of ways, both reader-
and author-centered.</p>';
      break;
case '190':
```

```
      echo '<blockquote><p class="blockquote"><span
class="volvelles">__</span><p>nbsp;A mechanism is faulty not for being too
artificial to account for living matter, but for not being mechanical enough,
for not being adquately machined. Our mechanisms are in fact organized into
parts that are not in themselves machines, while the organism is infinitely
machined, a machine whose every part or piece is a machine, but only
&quot;transformed by different folds that it receives.&quot; (<a
href="bibliography.html" target="_blank">Deleuze<span>Deleuze, Gilles. <U>The
Fold: Leibniz and the Baroque.</u> Tran. by Tom Conley. Minneapolis:
University of Minnesota Press, 1993.</span></a> 8)</p></blockquote>';
      break;
 case '200':
      echo '<p><span class="leibniz">__</span> As the final battles of
what would become known as the Thirty Years War were being waged in central
Europe, Gottfried Wilhelm Leibniz was born in Leipzig, a Saxony trading town
in the northeast corner of the Holy Roman Empire. His father Friedrich, a
professor of moral philosophy, died when Leibniz was only six, leaving a
large library of books. At the urging of a neighbor, Leibniz&#39;s mother
opened her husband&#39;s library to the curious young boy, who spent his
formative years wandering through its diverse collections (see <a
href="bibliography.html" target="_blank">Antognazza<span>Antognazza, Maria
Rose. <U>Leibniz: An Intellectual Biography.</U>New York, NY: Cambridge
University Press, 2009.</span></a> 33-7).</p>';
      break;
case '201':
      echo '<p><span class="leibniz">__</span><p>nbsp;In his father&#39;s
library, young Leibniz encountered German baroque poetry, moral philosophy,
legal books, and perhaps even a &quot;mass of unbound material deriving from
the father of Friedrich Leubnitz&#39;s second wife, the bookseller and
publisher, Bartholom&auml;us Voigt&quot; (<a href="bibliography.html"
target="_blank">Antognazza<span>Antognazza, Maria Rose. <U>Leibniz: An
Intellectual Biography.</U>New York, NY: Cambridge University Press,
2009.</span></a> 34). Thus even as his formal education drilled him the
strict German semi-Ramist tradition of logic, his informal wandering
introduced him to an eclectic mix of religious and aesthetic belief systems,
later influencing his notion of a <i>mathesis universalis</i> &mdash; a
method of generating, with mathematical certainty, the answer to any
philosophical question.</p>';
      break;
case '202':
      echo '<p><span class="leibniz">__</span><p>nbsp;One book in particular
had a deep influence on young Leibniz: Daniel Schwenter and G. P.
Harsd&ouml;rffer&#39;s <i>Delici&aelig; Physico-Mathematiac&aelig;</i> (1636-
1653), an imaginative collection of mathematical and poetic games and
inventions, including the first known design for a fountain pen as well as a
text-generating volvelle, the Denckring (<a href="bibliography.html"
target="_blank">Antognazza<span>Antognazza, Maria Rose. <U>Leibniz: An
Intellectual Biography.</U>New York, NY: Cambridge University Press,
2009.</span></a> 62-3, <a href="bibliography.html"
target="_blank">Reed<span>Reed, Eugene E. &quot;Leibniz, Wieland, and the
Combinatory Principle.&quot; <U>The Modern Language Review</U> 56.4 (1961):
529-37.</span></a> 529).</p><p>In fact, Leibniz&#39;s &quot;alphabet of human
thoughts&quot; contains traces of Stammw&ouml;rter theory, while his
<i>mathesis universalis</i> &mdash; an algorithm for permuting his alphabet
into universal knowledge &mdash; works much like the Denckring. Both operate
through a computational imaginary that condenses meaning into discrete
variables before exploding it exponentially, mechanically.</p>';
      break;
```

```
case '203':
      echo '<p><span class="leibniz">__</span><p>nbsp;In 1666, at the age of
twenty, Leibniz defended his <i>Dissertatio de Arte Combinatoria</i>.
Although now difficult to find and little studied &mdash; in a typical
dismissal, Reed states it &quot;contained nothing worthy of the Leibnizian
genius&quot; (<a href="bibliography.html" target="_blank">Reed<span>Reed,
Eugene E. &quot;Leibniz, Wieland, and the Combinatory Principle.&quot; <U>The
Modern Language Review</U> 56.4 (1961): 529-37.</span></a> 529) &mdash; the
work was published  immediately and became surprisingly successful for a
student dissertation (<a href="bibliography.html"
target="_blank">Antognazza<span>Antognazza, Maria Rose. <U>Leibniz: An
Intellectual Biography.</U>New York, NY: Cambridge University Press,
2009.</span></a> 62).</p><p>In it, Leibniz explores Ramon Llull&#39;s <i>ars
magna</i>, or his combinatorial art &mdash; a subject tackled by many
thinkers of the period, including Johann Heinrich Alsted and Athanasius
Kircher. Like Llull himself, Leibniz explores the art of combination not
simply as mechanical means for permuting discrete elements, but as system for
logical discovery.</p>';
      break;
case '204':
      echo '<blockquote><p class="blockquote"><span
class="leibniz">__</span><p>nbsp;It was fresh from the reading of
Harsd&ouml;rffer&#39;s work that Leibniz proceeded to demonstrate the
enormous potential of combination inherent in but a single poetic line -- an
application simple and obvious to the product of an age seeking new
combinations within the framework of the classical tradition. (<a
href="bibliography.html" target="_blank">Reed<span>Reed, Eugene E.
&quot;Leibniz, Wieland, and the Combinatory Principle.&quot; <U>The Modern
Language Review</U> 56.4 (1961): 529-37.</span></a> 529)</p></blockquote>';
      break;
case '205':
      echo '<p><span class="leibniz">__</span><p>nbsp;For young Leibniz,
knowledge emerges from an &quot;alphabet of human thoughts&quot; &mdash; that
is, a set of primitives that constitute the most basic units of philosophical
concepts. Combining these primitives through a logically-valid algorithm
will, by necessity, produce true arguments, thus contributing to a universal
science.</p><p>As Leibniz later argued, &quot;a kind of alphabet of human
thoughts can be worked out,&quot; and &quot;everything can be discovered and
judged by a comparison of the letters of this alphabet and an analysis of the
words made from them&quot; (<a href="bibliography.html"
target="_blank">Leibniz<span>Leibniz, G. W. &quot;On the General
Characteristic.&quot; In <U>Philosophical Papers and Letters, 2nd ed.</U>
Trans. by Leroy E. Loemker. Boston: D. Holland Publishing Company,
1956.</span></a> 222).</p>';
      break;
case '206':
      echo '<blockquote><p class="blockquote"><span
class="leibniz">__</span><p>nbsp;Leibniz&#39;s real proposal is not for a
device that relies on success in working out the correspondences of this
language, but for one that brings its syntax into effect prior even to the
discovery of its semantics. Even in the absence of a ready-made lexicon of
primitive concepts, the first stunning effect of Leibniz&#39;s clarity-
machine would be to give an a priori proof for its own possibility. (<a
href="bibliography.html" target="_blank">Selcer<span>Selcer, Daniel.
&quot;The Uninterrupted Ocean: Leibniz and the Encyclopedic
Imagination.&quot; <U>Representations</U> 98.1 (2007): 25-50.</span></a>
26)</p></blockquote>';
      break;
```

```
case '207':
      echo '<p><span class="leibniz">__</span><p>nbsp;Like the baroque theory
of Stammw&ouml;rter &mdash; still popular when Leibniz was writing his
Dissertatio &mdash; Leibniz&#39;s alphabet first reduces what is known before
producing what is true, setting up a closed system that a priori proves
itself (see <a href="bibliography.html" target="_blank">Selcer<span>Selcer,
Daniel. &quot;The Uninterrupted Ocean: Leibniz and the Encyclopedic
Imagination.&quot; <U>Representations</U> 98.1 (2007): 25-50.</span></a> 26).
Thus just as any word or line of poetry produced by Harsd&ouml;rffer&#39;s
Denckring always/already mirrors the <i>lingua adamica</i>, the originary and
perfect vernacular, so too does any knowledge produced through Leibniz&#39;s
alphabet of thoughts automatically mirror some universal truth. Both poetry
and philosophy become not product but process &mdash; a system, mechanized in
a set of nesting paper wheels, whose very existence instantiates its
combinatory possibilities.</p>';
      break;
case '208':
      echo '<p><span class="leibniz">__</span><p>nbsp;It is useful to frame
Leibniz&#39;s project not only through his baroque contemporaries and
forebears, but also through contrasting philosophical movements across
Europe.</p><p>A year after Leibniz published his <i>Dissertatio</i>, the
Englishman John Wilkins published his famous <i>Essay Towards a Real
Character and Philosophical Language</i>, which lays out his plan for a
philosophical language, immutable in both orthography and pronunciation.
Wilkins&#39; language (like that of Dalgarno and other seventeenth-century
thinkers) relies on a Ramist taxonomy that hierarchizes the world into a
system of branching nodes that move from the general to the specific,
accreting characteristics through distinctions. Thus a &quot;Tiger&quot; is a
viviparous, cloven-footed, rapacious cat-kind beast with shorter legs but a
larger body than other cats, characters by its spots (<a
href="bibliography.html" target="_blank">Wilkins<span>Wilkins, John. <U>An
Essay towards a Real Character and a Philosophical Language.</U> London:
1668.</span></a> 159).</p>';
      break;
case '209':
      echo '<p><span class="leibniz">__</span><p>nbsp;By contrast,
Leibniz&#39;s alphabet of human thoughts is radically reductive. Rather than
pursue the static, cumulative logic of tables and taxonomies, Leibniz
advocates stripping down philosophy to its most basic concepts, then devising
a system for combining these primitives to &quot;discover&quot; (i.e.,
generate) all truth. In this way, the philosopher becomes an engineer,
designing machines, and philosophy itself becomes the exponential power of
<i>ars combinatoria</i>.</p>';
      break;
case '210':
      echo '<blockquote><p class="blockquote"><span
class="leibniz">__</span><p>nbsp;This encyclopedia was to be both a
comprehensive and generative text, its categorical structure not merely
recording scientific, intellectual, historical and literary achievements but
functioning as a philosophical machine for the production and organization of
knowledge. (<a href="bibliography.html" target="_blank">Selcer<span>Selcer,
Daniel. &quot;The Uninterrupted Ocean: Leibniz and the Encyclopedic
Imagination.&quot; <U>Representations</U> 98.1 (2007): 25-50.</span></a>
29)</p></blockquote>';
      break;
case '211':
      echo '<p><span class="leibniz">__</span><p>nbsp;In short, whereas the
Ramist tables and taxonomies of a late seventeenth-century thinker like John
```

```
Wilkins enclose the world within a (somewhat arbitrary) structure,
Leibniz&#39;s program writes a structure into the world. The former orders
knowledge; the latter generates it.</p><p>The former reflects reality; the
latter <i>is</i> reality.</p>';
      break;
case '212':
      echo '<blockquote><p class="blockquote"><span
class="leibniz">__</span><p>nbsp;By the time he had completed his formal
education, Leibniz&#39;s vision had been born. He had decided what to do with
his life. Born into a fragmented world lacerated by religious, political, and
intellectual crises, &quot;Wilhelmus Pacidius&quot;, alia Gottfried Wilhelm
Leibniz, was going to put the pieces together to achieve a universal
synthesis for the glory of God and the happiness of mankind. This synthesis
would be designed to restore unity in multiplicity, unveiling the universal
harmony which, despite apparently unbridgeable divisions, governed reality at
both the metaphysical and the epistemological level[.] (<a
href="bibliography.html" target="_blank">Antognazza<span>Antognazza, Maria
Rose. <U>Leibniz: An Intellectual Biography.</U>New York, NY: Cambridge
University Press, 2009.</span></a> 66-7)</p></blockquote>';
      break;
case '213':
      echo '<p><span class="leibniz">__</span><p>nbsp;This &quot;alphabet of
human thoughts&quot; remained an undercurrent in Leibniz&#39;s philosophy
throughout his life, manifesting itself in a number of different plans: his
dream for a networked encyclopedia in which, through linking, every entry was
a microcosm of the human macrocosm (see <a href="bibliography.html"
target="_blank">Selcer<span>Selcer, Daniel. &quot;The Uninterrupted Ocean:
Leibniz and the Encyclopedic Imagination.&quot; <U>Representations</U> 98.1
(2007): 25-50.</span></a> 29); his lingua characteristica, or notation system
for concepts &quot;whose signs or characters serve the same purpose that
arithmetical signs serve for numbers&quot; (<a href="bibliography.html"
target="_blank">Leibniz<span>Leibniz, G. W. &quot;On the General
Characteristic.&quot; In <U>Philosophical Papers and Letters, 2nd ed.</U>
Trans. by Leroy E. Loemker. Boston: D. Holland Publishing Company,
1956.</span></a> 222); even his notion of &quot;monads&quot; &mdash;
discrete, irreducible primitives that nonetheless reflect the infinity of the
spiritual cosmos. More specifically, Leibniz develops his
&quot;alphabet&quot; through an account of a <i>mathesis universalis</i>, a
universal system for storing and generating knowledge.</p>';
      break;
case '214':
      echo '<blockquote><p class="blockquote"><span
class="leibniz">__</span><p>nbsp;Leibniz, who had subjected (and, if
necessary, corrected) all mathematical signs to Gutenberg&#39;s place-value
logic, recognized in Zero the nothingness preceding creation, and in One,
divine creation itself. No wonder, then, that his binary system was said to
have been able to describe the whole of being. (<a href="bibliography.html"
target="_blank">Kittler<span>Kittler, Friedrich A. &quot;The Perspective of
Print.&quot; <U>Configurations</U> 10.1 (Winter 2002): 37-50.</span></a>
48)</p></blockquote>';
      break;
case '215':
      echo '<blockquote><p class="blockquote"><span
class="leibniz">__</span><p>nbsp;[T]here are two crucial notions that
determine Leibniz&#39;s account of the <i>mathesis universalis</i>. The first
is the idea of the combinatorial nature of his system of notation: the
primitive concepts in the <i>lingua characteristica</i>, which provide the
foundation for the whole system, are conceived as being the basis of a
```

```
combinatorial process of the development of this language. The second idea is
that the whole system is open to a mechanical treatment, which is guaranteed
by the arithmetical nature of the calculus ratiocinator. The whole process of
generation, transformation, and decision in this language could be in
principle performed by a machine. (<a href="bibliography.html#westerhoff"
target="_blank">Westerhoff<span>Westerhoff, Jan C. &quot;Poeta Calculans:
Harsd&ouml;rffer, Leibniz, and the <i>Mathesis Universalis</i>.&quot;
<i>Journal of the History of Ideas</i> 60.3 (1999): 449-67.</span></a> 451;
for Leibniz&#39;s own account of his thinking on the subject, see &quot;On
the General Characteristic&quot; (1679))</p></blockquote>';
      break;
case '216':
      echo '<p><span class="leibniz">__</span><p>nbsp;Moreover, Leibniz
explicitly describes his &quot;alphabet of human thoughts,&quot; his
<i>mathesis universalis</i>, as a new technology that extends the human
organism.</p><blockquote><p class="blockquote">Once the characteristic
numbers for most concepts have been set up, however, the human race will have
a new kind of instrument which will increase the power of the mind much more
than optical lenses strengthen the eyes and which will be as far superior to
microscopes or telescopes as reason is superior to sign. The magnetic needle
has brought no more help to sailors than this lodestar will bring to those
who navigate the sea of experiments. (<a href="bibliography.html"
target="_blank">Leibniz<span>Leibniz, G. W. &quot;On the General
Characteristic.&quot; In <U>Philosophical Papers and Letters, 2nd ed.</U>
Trans. by Leroy E. Loemker. Boston: D. Holland Publishing Company,
1956.</span></a> 224)</p></blockquote>';
      break;
case '219':
      echo '<p><span class="leibniz">__</span><p>nbsp;Of course, in addition
seeking a <i>mathesis universalis</i> encompassing the sum total of human
knowledge, Leibniz also invented a binary system that represents each number
as a six-character string of 1s and 0s. Instead of proceeding in units of
ten, then &mdash; from 10 to 20, 20 to 30, and so on &mdash; Leibniz&#39;s
binary system moves by multiples of two, with 0 == 000000, 1 == 000001, 2 ==
000010, 3 == 000011. In this way, each string is calculated by permuting the
previous string. </p>';
      break;
case '220':
      echo '<p><span class="leibniz">__</span><p>nbsp;For Leibniz, his binary
system presents more than mathematical possibilities: in fact, it embodies
the world. As he wrote in a letter,</p><blockquote><p class="blockquote">For
0 can symbolize the void which preceded the creation of heaven and earth ....
At the beginning of the first day 1 existed, that is to say God. At the
beginning of the second day, heaven and earth, being created on the first
[that is, the end of the first day]. Finally at the beginning of the seventh
day everything already existed. This is why the last [day] is the most
perfect and the Sabbath, for all is created and complete. Thus 7 is written
as 111 without 0. And it is only in this way of writing by 0&#39;s and
1&#39;s that we see the perfection of the seventh [day] which is considered
holy. And it is even more remarkable that its character has some relation to
the Trinity. (quoted in <a href="bibliography.html"
target="_blank">Swetz<span>Swetz, F. J. &quot;Leibniz, the Yijing, and the
Religious Conversion of the Chinese.&quot; <U>Mathematics Magazine</U> 76.4
(2003): 276--291.</span></a> 285-6)</p></blockquote>';
      break;
case '221':
```

```
      echo '<p><span class="leibniz">__</span><p>nbsp;Of course, the binary
system is now embedded in digital computing through dyadic data-carrying
signals that pulse as 1 (present) or 0 (absent).</b></p>';
      break;
case '222':
      echo '<p><span class="leibniz">__</span><p>nbsp;Leibniz invented a
calculating machine to rival Blaise Pascal&#39;s &quot;Pascaline,&quot;
invented in 1645, the year before Leibniz&#39;s birth. Capable not only of
adding and subtracting, Leibniz&#39;s calculator was also intended to
multiply, divide, and extract root numbers (<a href="bibliography.html"
target="_blank">Antognazza<span>Antognazza, Maria Rose. <U>Leibniz: An
Intellectual Biography.</U>New York, NY: Cambridge University Press,
2009.</span></a> 144), and is now seen as a precursor to the modern computer.
Thus Leibniz not only imagined a <i>calculus ratiocinator</i> but, in at
least one sense, constructed one.   </p>';
      break;
case '223':
      echo '<p><span class="leibniz">__</span><p>nbsp;Although Leibniz did
not believe his binary system was the universal characteristic he was seeking
(<a href="bibliography.html" target="_blank">Cook and Rosemont<span>Cook,
Daniel J. and Henry Rosemont, Jr. &quot;The Pre-established Harmony between
Leibniz and Chinese Thought.&quot; <U>Journal of the History of Ideas</U 42.2
(April-June 1981): 253-267.</span></a> 264), he often returns to numbers
(specifically, Arabic numerals) as a model for scientific inquiry. He writes:
&quot;if we could explain by them abstract ideas, that would be the most
scientific method ... of unimaginable usefulness (quoted in <a
href="bibliography.html" target="_blank">Swetz<span>Swetz, F. J.
&quot;Leibniz, the Yijing, and the Religious Conversion of the Chinese.&quot;
<U>Mathematics Magazine</U> 76.4 (2003): 276--291.</span></a> 283).</p>';
      break;
case '224':
      echo '<blockquote><p class="blockquote"><span
class="leibniz">__</span><p>nbsp;But there can be no doubt that the general
art of combinations or characteristics contains much greater things than
algebra has given, for by its use all our thoughts can be pictures and as it
were, fixed, abridged, and ordered; pictured to others in teaching them,
fixed for ourselves in order to remember them; abridged so that they may be
reduced to a few; ordered so that all of them can be present in our thinking.
... I believe that when you examine the matter more seriously, you will agree
that this general characteristic will be of unbelievable value, since a
spoken and written language can also be developed with its aid which can be
learned in a few days and will be adequate to express everything that occurs
in everyday practice, and of astonishing value in criticism and discovery,
after the model of numeral characters. (<a href="bibliography.html"
target="_blank">Leibniz to von Tschirnhaus<span>Leibniz, G. W. &quot;Letter
to Walter von Tschirnhaus.&quot; In <U>Philosophical Papers and Letters, 2nd
ed.</U> Trans. by Leroy E. Loemker. Boston: D. Holland Publishing Company,
1956.</span></a>, May 1678; 193)</p></blockquote>';
      break;
case '225':
      echo '<p><span class="leibniz">__</span><p>nbsp;When, then, the French
Jesuit Joachim Bouvet wrote to Leibniz of the Chinese <i>I Ching</i> &mdash;
an ancient divination system of 64 hexagrams composed of six lines each
&mdash; both thinkers immediately saw its connection to Leibniz&#39;s binary
system (28 February 1698). Both use a base-2 system (1s and 0s for Leibniz,
broken and solid lines in the <i>I Ching</i>); both permute these two values
across 6 different slots; and both connect these permuted primitives to a
broad cosmology that encompasses the creation and ordering of the universe
```

(see AntognazzaAntognazza, Maria Rose. <U>Leibniz: An Intellectual Biography.</U>New York, NY: Cambridge University Press, 2009. 433-6, CammannCammann, Schuyler. &Chinese Hexagrams, Trigrams, and the Binary System. <U>Proceedings fo the American Philosophical Society</U> 135.4 (December 1991): 576-589., Cook and RosemontCook, Daniel J. and Henry Rosemont, Jr. "The Pre-established Harmony between Leibniz and Chinese Thought." <U>Journal of the History of Ideas</U 42.2 (April-June 1981): 253-267., RyanRyan, James A. "Leibniz' Binary System and Shao Yong's Yijing." <U>Philosophy East & West</U> 46 (1996)., SwetzSwetz, F. J. "Leibniz, the Yijing, and the Religious Conversion of the Chinese." <U>Mathematics Magazine</U> 76.4 (2003): 276--291., SwiderskiSwiderski, Richard M. "Bouvet and Leibniz: A Scholarly Correspondence." <U>Eighteenth-Century Studies</U> 14.2 (Winter 1980-1): 135-50. for further discussion). ';
      break;
case '226':
      echo ' __ nbsp;In a subsequent letter to Leibniz, written on November 1, 1701, Bouchet included a diagram containing all 64 hexagrams laid out in a circle. As in Harsdörffer's text-generating volvelles, or the wheels of Ramon Llull's ars magna, or even the units of the Kabbalah — another mystical system which fascinated Leibniz (see CoudertCoudert, Alison. <U>Leibniz and the Kabbalah.</U> New York: Springer, 1995.) — the diagram represents the infinite through the systematic permutation and combination of discrete elements. Therefore, much as Leibniz's dyads embody a programmatic cosmology, so the I Ching uses minimalist abstractions to produce an entire world — both yin and yang, in all their combinations. ';
      break;
case '227':
      echo '<blockquote> __ nbsp;What is amazing in this reckoning is that this arithmetic by 0 and 1 is found to contain the mystery of the lines of an ancient King and philosopher named Fuxi, who is believed to have lived more than 4000 years ago, and whom the Chinese regard as the founder of their empire and their sciences. There are several linear figures attributed to him, all of which come back to this arithmetic, but it is sufficient to give here the Figure of the Eight Cova, as it is called, which is said to be fundamental, and to join to them the explanation which is obvious, provided that one notices, firstly, that a whole line — means unity, or 1, and secondly, that a broken line — means zero, or 0. (LeibnizLeibniz, G. W. "Explanation of Binary Arithmetic." online) </blockquote>';
      break;
case '228':
      echo ' __ nbsp;More than the obvious similarity between Leibniz's binary system and the hexagrams of the I Ching, their historical significance excited Joachim Bouvet. As a Jesuit missionary living in China, Bouvet read any cultural syncretism as a confirmation of his ecumenical belief system. For instance, he speculated

that Chinese characters were "the writing used among the learned before the Flood," much like Schottel's theory of Stammwörtter (quoted in SwiderskiSwiderski, Richard M. "Bouvet and Leibniz: A Scholarly Correspondence." <U>Eighteenth-Century Studies</U> 14.2 (Winter 1980-1): 135-50. 138), and sought Christian symbols in the pre-Christian people, places and narratives of the Chinese (RyanRyan, James A. "Leibniz' Binary System and Shao Yong's Yijing." <U>Philosophy East & West</U> 46 (1996). 61, SwiderskiSwiderski, Richard M. "Bouvet and Leibniz: A Scholarly Correspondence." <U>Eighteenth-Century Studies</U> 14.2 (Winter 1980-1): 135-50. 149). Thus, for Bouvet, the fact that one of the foremost European thinkers had independently devised a system that was (in his mind) nearly identical to the I Ching only proved Christianity to be an all-encompassing Truth undergirding all knowledge at all times, even before the coming of Christ. ';
	break;
case '229':
	echo ' __ nbsp;In this way, the syncronicity between Leibniz's binary system and the I Ching transformed both from culturally- and socially-embedded media — the former, a product of the scientific discourse congealing at the end of the seventeenth century in Europe; the latter, an ancient form of communication between individuals, their present and their past — into machines of conversion, re-fitting the Chinese hexagrams to mechanical Christian worldview. Like the systems themselves, the Leibniz-Bouvet correspondence isolates discrete cultural systems from each other, levels them into abstract variables, and recombines them to write a new text, a new narrative of Chinese and Christian history. In short, Bouvet seeks a new Lullian wheel — an ars combinatoria to programmatically convert the Chinese through the mechanical permutation of symbols. ';
	break;
case '230':
	echo ' __ nbsp;As with baroque Stammwörtter theory (which inspired young Leibniz), Leibniz's binary system searches for origins as a path to the future. In other words, it seeks to disappear knowledge into universal characters, even as the seemingly transparent algorithms and abstractions it uses can never escape their own ideology. ';
	break;
case '231':
	echo ' __ nbsp;Since his death, Leibniz has been transformed from polymathic, baroque thinker with an almost mystical worldview to, as Cammann describes him, one "too rational-minded to pay any attention to astrology or other aspects of the occult" (CammannCammann, Schuyler. &Chinese Hexagrams, Trigrams, and the Binary System. <U>Proceedings fo the American Philosophical Society</U> 135.4 (December 1991): 576-589. 588). Even Florian Cramer positions Leibniz "opposite to Kuhlmann's Kabbalist cosmology," writing that he "on the grounds of rational science transforms Lullism into symbolic logic and an early mechanical calculation device" (CramerCramer, Florian. <U>Words Made Flesh: Code, Culture, Imagination.</U> Piet Zwart Institute, 2005. 48). Yet from his search for a mathesis universalis, to his "alphabet of human thoughts," to his binary cosmology and its relationship to the Chinese I Ching, Leibniz shows a deep sympathy systems that seem now

```
to us &quot;non-scientific&quot; or &quot;non-rational.&quot; In fact, he
bases many of his epistemological claims on an <i>ars combinatoria</i> rooted
in the very Lullism and Kabbalah Cramer sees as antithetical. The divine and
the programmatic, or the aesthetic and the rational, do not always undermine
each other but, in Leibniz&#39;s world, converge.</p>';
      break;
case '232':
      echo '<p><span class="leibniz">__</span><p>nbsp;Proteus poetry becomes
a major source for much of Leibniz&#39;s work in the <i>Dissertatio</i>, for
obvious reasons. By reducing language (i.e. meaning) to a set of primitives
that can combine to generate new meaning, proteus verse is a model for
Leibniz&#39;s &quot;alphabet of human thoughts.&quot; </p>';
      break;
case '233':
      echo '<blockquote><p class="blockquote"><span
class="leibniz">__</span><p>nbsp;It is not always the case that the
variations of a proteus verse have the same meaning, nor is a proteus verse
always defined in this way. Leibniz, nevertheless, considers it part of the
meaning of &quot;proteus verse&quot; that its variations have the same sense.
Regardless of the way the elements of a proteus verse are recombined, the
meaning will be left unchanged. This is particularly interesting because this
feature of a synthesis of elements is mirrored in one of Leibniz&#39;s ideas
on analysis. In constructing the <i>lingua characteristica</i> as a part of
the <i>mathesis universalis</i>, Leibniz considers it as necessary to analyze
concepts into their most simple components, the primitive concepts or
&quot;alphabet of human thoughts." (<a href="bibliography.html#westerhoff"
target="_blank">Westerhoff<span>Westerhoff, Jan C. &quot;Poeta Calculans:
Harsd&ouml;rffer, Leibniz, and the <i>Mathesis Universalis</i>.&quot;
<i>Journal of the History of Ideas</i> 60.3 (1999): 449-67.</span></a>
458)</p></blockquote>';
      break;
case '234':
      echo '<p><span class="leibniz">__</span><p>nbsp;Leibniz&#39;s
<i>Dissertatio de arte combinatoria</i> is deeply influenced by Llull&#39;s
<i>ars</i>, such that it could be described as a fundamentally Lullian work.
As in the <i>ars</i>, Leibniz breaks down triads of constants, or
&quot;wholes,&quot; into a set of three dyads, such that two types of
variation are possible from every whole composed of situated parts (see <a
href="bibliography.html" target="_blank">Leibniz<span>Leibniz, G. W.
&quot;Dissertation on the Art of Combination.&quot; In <U>Philosophical
Papers and Letters, 2nd ed.</U> Trans. by Leroy E. Loemker. Boston: D.
Holland Publishing Company, 1956.</span></a> 77). And, as in the Lullian Art,
Leibniz seeks to use this relationship between wholes and parts, in which
combination provides the mechanism for traversing all different
possibilities, as a way of explaining the world.</p>';
      break;
case '235':
      echo '<blockquote><p class="blockquote"><span
class="leibniz">__</span><p>nbsp;That is what Leibniz explains in an
extraordinary piece of writing: a flexible or an elastic body still has
cohering parts that form a fold, such that they are not separated into parts
of parts but are rather divided to infinity in smaller and smaller folds that
always retain a certain cohesion. Thus a continuous labyrinth is not a line
dissolving into independent points, as flowing sand might dissolve into
grains, but resembles a sheet of paper divided into infinite folds or
separated into bending movements, each one determined by the consistent or
conspiring surroundings. (<a href="bibliography.html"
target="_blank">Deleuze<span>Deleuze, Gilles. <U>The Fold: Leibniz and the
```

```
Baroque.</u> Tran. by Tom Conley. Minneapolis: University of Minnesota Press,
1993.</span></a> 6)</p></blockquote>';
      break;
case '250':
      echo '<p><span class="permutation">__</span> In 1651, five years
after Leibniz was born in Saxony, Quirinus Kuhlmann was born several hundred
miles to the southeast in Brauslau (Wroclaw), Lower Silesia. Also like
Leibniz, Kuhlmann&#39;s father died when he was very young, leaving his pious
Lutheran mother to raise the sickly, stuttering young Quirinus alone.</p>';
      break;
case '251':
      echo '<p><span class="permutation">__</span> Growing up in Lower
Silesia, one of the largest influences on Kuhlmann&#39;s early life was
Martin Opitz (1597-1639), a celebrated baroque poet of G. P
Harsd&ouml;rffer&#39;s generation, also born in Lower Silesia. As Beare
writes, &quot;the students of the two Breslau schools recited Opitz poems and
panegyrics to Opitz in the market place&quot; as part of a yearly celebration
that attracted a parade of well-known poets, including Andreas Gryphius and
Daniel Caspar (<a href="bibliography.html" target="_blank">Beare<span>Beare,
Robert L. &quot;Quirinus Kuhlmann: The Religious Apprenticeship.&quot;
<U>PMLA</U> 68.4 (1953): 828-62.</span></a> 834). By age seventeen, Kuhlmann
had published his first book of verse, <i>Unsterbliche Sterblichkeit</i>
(1668).</p>';
      break;
case '252':
      echo '<p><span class="permutation">__</span> In 1669, a year after
publishing his first (and highly acclaimed) book of poems, illness struck
Quirinus Kuhlmann. Later, he claimed to have had prophetic visions during
this period &mdash; images of Christ, the Devil, and two angels in the guise
of flashing lights which flared in his left peripheral vision for many years
(see <a href="bibliography.html" target="_blank">Beare<span>Beare, Robert L.
&quot;Quirinus Kuhlmann: The Religious Apprenticeship.&quot; <U>PMLA</U> 68.4
(1953): 828-62.</span></a> 838). Famously, he continued to excite similar
visions by covering the walls of his room with reflective papers.</p>';
      break;
case '253':
      echo '<p><span class="permutation">__</span> From an early age,
Athanasius Kircher&#39;s work fascinated Quirinus Kuhlmann &mdash;
particularly his treatise on Ramon Llull&#39;s <i>ars magna</i>, or art of
combinations. Like Leibniz, Kuhlmann</p><blockquote><p
class="blockquote">planned, using this system, to write a series of
encyclopedic studies, some in several volumes, covering nineteen fields of
knowledge, the sum total of human knowledge at the time. Nothing came of this
plan, but his failure to carry it out is lamented by later writers, although
they distrusted his reliance upon divine impulse. (<a
href="bibliography.html" target="_blank">Beare<span>Beare, Robert L.
&quot;Quirinus Kuhlmann: The Religious Apprenticeship.&quot; <U>PMLA</U> 68.4
(1953): 828-62.</span></a> 840)</p></blockquote>';
      break;
case '254':
      echo '<p><span class="permutation">__</span> Like Leibniz,
Kuhlmann also encountered Llull through the work of Georg Philipp
Harsd&ouml;rffer and was very likely familiar with his Denckring, a mechanism
for generating correct German words, rhymes and poems.</p>';
      break;
case '255':
      echo '<p><span class="permutation">__</span> Perhaps Quirinus
Kuhlmann&#39;s most fascinating work is in his <i>Himmlische Libes-
```

```
k&uuml;sse, </i>or Heavenly Love-kisses, published two years after his
prophetic visions of Christ. Within this collection of fifty religious-themed
sonnets, Kuhlmann includes one permutation poem: XLI, the 41st, entitled
&quot;Vom Wechsel menschlicher Sachen,&quot; or &quot;On the Variations of
Human Matters.&quot;</p><p>The first twelve lines are composed of sixteen
monosyllabic words each. The first and last words in these lines, as well as
the last couplet in the sonnet, must remain stable to preserve the rhyme
scheme; the middle words, totaling 169, are permutable.</p>';
      break;
case '256':
      echo '<p><span class="permutation">__</span> In his commentary on
the sonnet, Kuhlmann stops counting its permutations at fifty. However, as
Florian Cramer points out, the number of possible sonnets generated by
&quot;On the Variations of Human Matters&quot; is not 50 but
3.399*10<sup>117</sup> (<a href="bibliography.html"
target="_target">Cramer<span>Cramer, Florian. <U>Words Made Flesh: Code,
Culture, Imagination.</U> Piet Zwart Institute, 2005.</span></a> 47). </p>';
      break;
case '257':
      echo '<blockquote><p class="blockquote"><span
class="permutation">__</span> It functions as a world machine,
permuting, both in the meaning of its words and in its combinatory mechanics,
an inventory of the micro- and macrocosm.  Its own couplet, and its
afterword, claims that to grasp the principle of the permutation of things
means to have wisdom of the world.  This alludes to the both to the use of
Solomon&#39;s proverbs in the poem and adaptions of Solomon&#39;s Song
Celestial in the volume in which it appeared. Through this intertextuality,
the poem renders itself a Solomonic machine. It is a computational reverse
engineering of Solomon&#39;s wisdom, considering the proverbs as they are
written in the Bible the fragmentary output of an occult machine. (<a
href="bibliography.html" target="_target">Cramer<span>Cramer, Florian.
<U>Words Made Flesh: Code, Culture, Imagination.</U> Piet Zwart Institute,
2005.</span></a> 47)</p></blockquote>';
      break;
case '258':
      echo '<p><span class="permutation">__</span> Quirinus
Kuhlmann&#39;s permutable sonnet participates in the mechanical, programmatic
epistemology of the German baroque, so well articulated by his contemporary
Leibniz. That is: it reduces its verse to a finite set of primitives, and
algorithmically permutes those primitives to generate new texts, new poems,
and thereby new meaning. For Kuhlmann, to know is to accrete and accumulate,
not to reduce and refine.</p>';
      break;
case '259':
      echo '<p><span class="permutation">__</span> Yet, unlike
Harsd&ouml;rffer&#39;s Denckring, Kuhlmann&#39;s sonnet XLI is legible, in
both the obvious and etymological sense of the word. I can reproduce a line
from his poem on the screen for you, because that line exists, regardless of
how one might manipulate it after the moment of reading.</p><p>What would I
reproduce of the Denckring? Or the Maria Stella? Or August de Campos&#39;
&quot;SOS&quot;? What is the text one has &quot;read&quot; &mdash; and what
has been already &quot;written&quot;?</p>';
      break;
case '260':
      echo '<p><span class="permutation">__</span> In short, Quirinus
Kuhlmann does not instruct the reader to physically cut up his poem, as
Burroughs and Gysin do; nor has he already cut it up, as the women of Little
Gidding have done to the Bible. For Kuhlmann, the manipulation &mdash; the
```

handling of the text — is mental, not material — locked in the mind, not on the page. And the Word is not matter, but spirit, spinning off into the void. ';
	break;
case '261':
	echo ' __ History now monumentalizes Leibniz as the great seventeenth-century rationalist, too mathematically-inclined "to pay any attention to astrology or other aspects of the occult" (CammannCammann, Schuyler. &Chinese Hexagrams, Trigrams, and the Binary System. <U>Proceedings fo the American Philosophical Society</U> 135.4 (December 1991): 576-589. 588), while brushing away Quirinus Kuhlmann as a delusional and possibly mentally deranged mystic. (Worse still, Kuhlmann's role in seventeenth-century German poetry is often altogether forgotten; see, for instance, BeareBeare, Robert L. "Quirinus Kuhlmann: The Religious Apprenticeship." <U>PMLA</U> 68.4 (1953): 828-62. 828.) Yet, born only five years and several hundred miles apart, both thinkers sought ars combinatoria as a computer for organizing — and, in doing so, generating — knowledge in a society newly overwhelmed with printed texts. ';
	break;
case '262':
	echo '<blockquote> __ In 1674, three years after his permutational sonnet, Quirinus Kuhlmann published his correspondence with Athanasius Kircher in a book epistolae duae. It documents an early debate about automatically generated art and its cognitive limitations. In his letters, Kuhlmann rejects a purely technical application of Llull's combinatorics, claiming that "knowing Lullus does not mean to have knowledge in the alphabetum of his Ars in order to build syllogisms with it, but to grasp the true power from the universal book of nature that is hidden underneath it, and apply it to everything." (CramerCramer, Florian. <U>Words Made Flesh: Code, Culture, Imagination.</U> Piet Zwart Institute, 2005. 105) </blockquote>';
	break;
case '263':
	echo ' __ Quirinus Kuhlmann was not the first to write what has become known as proteus poetry. Optatianus Porfyrius, writing in the fourth century AD, composed twenty stanzas by permuting the elements of one (see HallynHallyn, Fernand, and Roxanne Lapidus. "'A Light-Weight Artifice': Experimental Poetry in the 17th Century." <U>SubStance</U> 22.2/3, Issue 71/72: Special Issue: EpistÈmocritique (1993): 289-305. 292, CramerCramer, Florian. <U>Words Made Flesh: Code, Culture, Imagination.</U> Piet Zwart Institute, 2005. 44), and there is a long history of centos — from the Latin word cento, an overcoat stitched together from patches of cloth — which combine phrases from difference source texts to produce Latin verse. In 1561, Julius Caesar Scalinger, looking to the Latin and Greek traditions of permutation poetry, created his own combinatory verse by permuting elements of the line Perfide sperasti divos falleri Proteu. Published in his Poetices Libri Septem (II.30), he called his poem "Proteus" after the shape-shifting sea god (<a href="bibliography.html" target="_target">CramerCramer, Florian. <U>Words Made Flesh: Code, Culture, Imagination.</U> Piet Zwart Institute,

```
2005.</span></a> 44, <a href="bibliography.html"
target="_blank">Higgins<span>Higgins, Dick. <U>Pattern Poetry: Guide to an
Unknown Literature.</U> Albany: State University of New York Press,
1987.</span></a> 183).</p>';
      break;
case '264':
      echo '<p><span class="permutation">__</span> In 1626, Thomas
Lansius composed a couplet of monosyllabic words, each one permutable except
<i>bona</i> and <i>mala</i>:</p><blockquote><p class="blockquote"><i>Les,
Rex, Grex, Res, Spes, Jus, Tus, Sal, Sol, (bona) Lux, Laus:<br/>Mars, Mors,
Sors, LIs Vis, Styx, Pus, Nox, Fex, (mala) Crux,
Fraus.</i></p></blockquote><p>Lansius&#39; poem &mdash; containing 39,916
possible variations (<a href="bibliography.html"
target="_blank">Bernoulli<span>Bernoulli, Jacob. <U>The art of conjecturing,
together with Letter to a friend on sets in court tennis.</U> Trans. by Edith
Dudley Sylla. Baltimore, MD: Johns Hopkins University Press,
2006.</span></a>197) &mdash; is reprinted in Johann Heinrich Alsted&#39;s
famous encyclopedia, influenced by the Lullian Art (<a
href="bibliography.html" target="_target">Cramer<span>Cramer, Florian.
<U>Words Made Flesh: Code, Culture, Imagination.</U> Piet Zwart Institute,
2005.</span></a> 45).</p>';
      break;
case '265':
      echo '<p><span class="permutation">__</span> In 1616, Bernard
Bauhuis published a book of Latin epigrams, <i>Jesu epigrammatum selectorum
libri</i>, containing two permutational lines, the first having (as Bauhuis
writes) &quot;as many faces, in fact, as there are stars in the heavens
(since it can be transformed one thousand and twenty-two times, while its
meaning is retained and the rules of epic poetry are observed)&quot;
(Bauhuis, quoted in <a href="bibliography.html"
target="_blank">Hallyn<span>Hallyn, Fernand, and Roxanne Lapidus.
&quot;&#39;A Light-Weight Artifice&#39;: Experimental Poetry in the 17th
Century.&quot; <U>SubStance</U> 22.2/3, Issue 71/72: Special Issue:
EpistÈmocritique (1993): 289-305.</span></a> 290).</p><blockquote><p
class="blockquote"><i>Tot tibi sunt dotes, Virgo, quot sidera
caelo.</i></p></blockquote><p>As Hallyn describes, the next year Erycius
Puteanus (Henri Van de Putte) published the <i>Pietatis Thaumata</i>, in
which he writes out all 1,022 variations, filling thirty-six pages (<a
href="bibliography.html" target="_blank">Hallyn<span>Hallyn, Fernand, and
Roxanne Lapidus. &quot;&#39;A Light-Weight Artifice&#39;: Experimental Poetry
in the 17th Century.&quot; <U>SubStance</U> 22.2/3, Issue 71/72: Special
Issue: EpistÈmocritique (1993): 289-305.</span></a> 291).</p>';
      break;
case '266':
      echo '<blockquote><p class="blockquote"><span
class="permutation">__</span> Puteanus does not fail to play up the
mythological theme of Proteus. He  recalls that because of  the multiplicity
of Proteus&#39;s changing appearances, Heraclides identifies him with the
original Chaos, after which Providence separated the elements and assembled
the distinct forms of the world.  But the symbolism of Proteus is itself
protean: Nature itself, in its variety, is a Proteus, as is Christ, as well
as Bauhusius, who is equally a new  Pythagoras and an Orpheus. (<a
href="bibliography.html" target="_blank">Hallyn<span>Hallyn, Fernand, and
Roxanne Lapidus. &quot;&#39;A Light-Weight Artifice&#39;: Experimental Poetry
in the 17th Century.&quot; <U>SubStance</U> 22.2/3, Issue 71/72: Special
Issue: EpistÈmocritique (1993): 289-305.</span></a> 293)</p></blockquote>';
      break;
case '267':
```

```
      echo '<p><span class="permutation">__</span> Many others generated
proteic verse during the seventeenth century, including Georg Keppisius (<a
href="bibliography.html" target="_blank">Hallyn<span>Hallyn, Fernand, and
Roxanne Lapidus. &quot;&#39;A Light-Weight Artifice&#39;: Experimental Poetry
in the 17th Century.&quot; <U>SubStance</U> 22.2/3, Issue 71/72: Special
Issue: EpistÈmocritique (1993): 289-305.</span></a> 297), Pietro Francesco
Passerini (<a href="bibliography.html" target="_blank">Higgins<span>Higgins,
Dick. <U>Pattern Poetry: Guide to an Unknown Literature.</U> Albany: State
University of New York Press, 1987.</span></a> 183), Carolus of Goldsten and
Johan-Philipp Ebel (<a href="bibliography.html"
target="_blank">Hallyn<span>Hallyn, Fernand, and Roxanne Lapidus.
&quot;&#39;A Light-Weight Artifice&#39;: Experimental Poetry in the 17th
Century.&quot; <U>SubStance</U> 22.2/3, Issue 71/72: Special Issue:
EpistÈmocritique (1993): 289-305.</span></a> 298). Georg Philipp
Harsd&ouml;rffer also experimented with permutational verse.</p>';
      break;
case '300':
      echo '<p><span class="volvelles">__</span><p>nbsp;For the thirteenth-
century Majorcan monk Ramon Llull, as for Quirinus Kuhlmann, the art of
combination began with a vision.</p><blockquote><p class="blockquote">Ramon
went up a certain mountain not far from his home, in order to contemplate God
in greater tranquility. When he had been there scarcely a full week, it
happened that one day while he was gazing intently heavenward the Lord
suddenly illuminated his mind, giving him the form and method for writing the
aforementioned book against the errors of the unbelievers.</p><p
class="blockquote">Giving thanks to the Almighty, he came down from the
mountain and returned at once to the above-mentioned abbey, where he began to
plan and write the book in question, calling it at first the <i>Ars
major</i>, and later on the <i>Ars generalis</i>. (from the <i>Vita
coaetanea</i>, in <a href="bibliography.html"
target="_blank">Llull<span>Llull, Ramon. <u>Selected Works of Ramon
Llull.</u> Vol. 1. Ed. and trans. by Anthony Bonner. Princeton, NJ: Princeton
University Press, 1985.</span></a> 22)</p></blockquote>';
      break;
case '301':
      echo '<p><span class="volvelles">__</span><p>nbsp;Just as
Harsd&ouml;rffer&#39;s Denckring and Leibniz&#39;s <i>mathesis
universalis</i> reflect the political milieu of the Thirty  Years War, Ramon
Llull&#39;s <i>ars</i> is, in some sense, a microcosm of Majorcan culture
during the thirteenth century, representing a fundamental paradox
&mdash;</p><blockquote><p class="blockquote">&mdash; that of a figure coming
from a small island in the western Mediterranean and from what nowadays is
considered a minority culture, developing one of the most universalist of
systems, one which he presented to popes, kings, sultans, and universities in
Spain, France, Italy, and North Africa during his lifetime, and which brought
him extraordinary fame throughout Europe in the Renaissance. (<a
href="bibliography.html" target="_blank">Bonner<span>Bonner, Anthony.
&quot;Introduction.&quot; <u>Selected Works of Ramon Llull.</u> Vol. 1. Ed.
and trans. by Anthony Bonner. Princeton, NJ: Princeton University Press,
1985.</span></a> 3)</p></blockquote>';
      break;
case '302':
      echo '<p><span class="volvelles">__</span><p>nbsp;As Bonner points out,
during the thirteenth century an imperialist push from Catalonia into Greece
and Turkey &quot;gave the Catalans a sense of their own identity&quot; that
was &quot;tied up not only with a feeling that they were carrying out a
divine destiny, but also with a feeling for their language as an expression
of that destiny&quot; (<a href="bibliography.html"
```

```
target="_blank">Bonner<span>Bonner, Anthony. &quot;Introduction.&quot;
<u>Selected Works of Ramon Llull.</u> Vol. 1. Ed. and trans. by Anthony
Bonner. Princeton, NJ: Princeton University Press, 1985.</span></a> 7-8).
Thus Ramon Llull &mdash; born in Palma, Majorca &mdash; was writing his ars
in Catalan during a time in which his native tongue represented a divine
right &mdash; in fact, when it itself was a kind of machine for
conversion.</p>';
      break;
case '303':
      echo '<p><span class="volvelles">__</span><p>nbsp;The combinatory
system of what would become known as the Lullian Art, or the <i>ars magna</i>
of Ramon Llull, is rooted in his earlier manuscripts, particularly the <i>Ars
demonstrativa</i> (1283). In it, Llull takes five letters: A, S, T, V, and X,
each representing some Figure (in addition to other Figures regarding, for
instance, the Principles of Law, which we&#39;ll put aside for the sake of
simplicity). Y and Z represent, respectively, truth and falsehood, while the
letters B through R are copied as a principle of each figure A, S, T, V and
X, constructing binary combinations such as AB, AC, AD; SB, SC, SD; and so
on. Thus through the constrained permutation of all twenty-three letters of
the medieval alphabet, Llull demonstrates how to &quot;know and love A,&quot;
or God (<a href="bibliography.html" target="_blank">Llull<span>Llull, Ramon.
<u>Selected Works of Ramon Llull.</u> Vol. 1. Ed. and trans. by Anthony
Bonner. Princeton, NJ: Princeton University Press, 1985.</span></a> <i>Ars
demonstrativa</i> 415).</p>';
      break;
case '304':
      echo '<p><span class="volvelles">__</span><p>nbsp;The Figure S,
representing the intelligent soul, takes on a special role as, in the words
of Anthony Bonner, &quot;a connective or relational figure&quot; that
&quot;works externally, connecting [the Art&#39;s] components to the outside
world, to the &#39;artist&#39; or to those to whom the Art is directed&quot;
(<a href="bibliography.html" target="_blank">Bonner<span>Bonner, Anthony.
&quot;Introduction.&quot; <u>Selected Works of Ramon Llull.</u> Vol. 1. Ed.
and trans. by Anthony Bonner. Princeton, NJ: Princeton University Press,
1985.</span></a> 309). Figure T, then, acts as the mechanism that allows S to
relate to the external world. As Llull writes, &quot;just as the artisan
fashions the artificial form by means of his tools, so S does what it does in
this Art by means of T, introducing it objectively into its powers, and
introducing it into the compartments of the figures, insofar was it can be
admitted into them&quot; (<a href="bibliography.html"
target="_blank">Llull<span>Llull, Ramon. <u>Selected Works of Ramon
Llull.</u> Vol. 1. Ed. and trans. by Anthony Bonner. Princeton, NJ: Princeton
University Press, 1985.</span></a> <i>Ars demonstrativa</i> 327). </p>';
      break;
case '305':
      echo '<blockquote><p class="blockquote"><span
class="volvelles">__</span><p>nbsp;The roles of the figures can perhaps be
clarified by a very loose analogy with a computer: if A, V, X, Y, Z, the
Elemental Figure, and the three Figures of Theology, Philosophy, and Law
contain the basic data, Figure T provides the processing unit, and Figure S
constitutes the terminal or control unit assuring correct access to the data
and processing. (<a href="bibliography.html"
target="_blank">Bonner<span>Bonner, Anthony. &quot;Introduction.&quot;
<u>Selected Works of Ramon Llull.</u> Vol. 1. Ed. and trans. by Anthony
Bonner. Princeton, NJ: Princeton University Press, 1985.</span></a>
310)</p></blockquote>';
      break;
case '306':
```

```
      echo '<p><span class="volvelles">__</span><p>nbsp;In <i>Ars
demonstrativa</i>, Llull goes on to develop other Figures that specify
particular relationships between different elements. Only the Ninth Figure,
however, concerns us. </p><blockquote><p class="blockquote">It is composed of
six revolving circles along with a wheel situated in the middle in which
there is a triangle representing the five triangles of T. It is made up of
circles, each inside the other, like weights inserted one in another, with a
pin in the middle, which keeps the circles in place. ... This figure contains
and includes all the other figures of this Art, as is explained above;
moreover, all the compartments of the other figures can be formed by
revolving the circles of this figure in the right way[.] (<a
href="bibliography.html" target="_blank">Llull<span>Llull, Ramon. <u>Selected
Works of Ramon Llull.</u> Vol. 1. Ed. and trans. by Anthony Bonner.
Princeton, NJ: Princeton University Press, 1985.</span></a> <i>Ars
demonstrativa</i> 333-34)</p></blockquote>';
      break;
case '307':
      echo '<p><span class="volvelles">__</span><p>nbsp;This figure, the
Ninth in Llull&#39;s <i>Ars demonstrativa</i>, represents an early attempt to
mechanize the ars. Here, instead of tracing Llull&#39;s logic through
dispersed rules and tables of combinations, the user may operate a single
machine whose laws simply are the physical relationships between the spinning
paper discs. In this way, reading a set of interlocking letters writes a
representation of the world, just as writing their combinations becomes an
act of exploration, a voyage, a reading. </p>';
      break;
case '308':
      echo '<p><span class="volvelles">__</span><p>nbsp;The system outlined
in the <i>Ars demonstrativa</i> proved too complicated and was derided by the
intellectual community in Paris, where Llull was teaching. After returning to
Montpellier, he set to work reducing his system to only four figures, thereby
(as later described in his dictated autobiography) &quot;eliminating &mdash;
or rather disguising, because of the weakness of human intellect which he had
witness in Paris &mdash; twelve of the sixteen figures that had formerly
appeared in his Art&quot; (from the <i>Vita coaetanea</i>, in <a
href="bibliography.html" target="_blank">Llull<span>Llull, Ramon. <u>Selected
Works of Ramon Llull.</u> Vol. 1. Ed. and trans. by Anthony Bonner.
Princeton, NJ: Princeton University Press, 1985.</span></a> <i>Ars
demonstrativa</i>  22). The vision passed to him from God proved too
complicated for humans; so he simplified it to an <i>Ars inventiva
veritatis</i>, leading to his two most influential works: the <i>Ars
brevis</i> (1308) and the <i>Ars generalis ultima</i> (1305-1308).</p>';
      break;
case '309':
      echo '<p><span class="volvelles">__</span><p>nbsp;In the <i>Ars
brevis</i>, Ramon Llull reduces the twelve figures of his earlier system to
four, and the twenty-three letters to nine, B through K.</p><p>The First
Figure, A (or God), takes letters B through K &mdash; representative of the
nine Divine Dignities (or attributes), such as goodness, greatness, eternity,
and so on &mdash; and inscribes them around the edge of a circle &quot;to
show that any subject can become a predicate and vice versa, as when one
says, &#39;goodness is great,&#39; &#39;greatness is good,&#39; and so
on&quot; (<a href="bibliography.html" target="_blank">Llull<span>Llull,
Ramon. <u>Selected Works of Ramon Llull.</u> Vol. 1. Ed. and trans. by
Anthony Bonner. Princeton, NJ: Princeton University Press, 1985.</span></a>
<i>Ars demonstrativa</i>  <i>Ars brevis</i> 582). As Llull argues, since
&quot;everything that exists is implicit in the principles of this
figure,&quot; then &quot;whatever exists is reducible to the above-mentioned
```

```
principles&quot; (<a href="bibliography.html"
target="_blank">Llull<span>Llull, Ramon. <u>Selected Works of Ramon
Llull.</u> Vol. 1. Ed. and trans. by Anthony Bonner. Princeton, NJ: Princeton
University Press, 1985.</span></a> <i>Ars brevis</i> 583).</p>';
      break;
case '310':
      echo '<p><span class="volvelles">__</span><p>nbsp;The Second Figure
consists of relationships between different elements, represented a set of
three nesting triangles: the first showing difference, concordance and
contrariety; the second showing the beginning, middle and end of all things;
and the third showing the relationships of majority, equality and minority.
As in the <i>Ars demonstrativa</i>, this Figure (&quot;T&quot;) serves the
First Figure as its constraint or &quot;tool.&quot; As Lull writes: &quot;by
joining this figure to the first, the intellect acquires knowledge. And
because this figure is general, therefore the intellect becomes general&quot;
(<a href="bibliography.html" target="_blank">Llull<span>Llull, Ramon.
<u>Selected Works of Ramon Llull.</u> Vol. 1. Ed. and trans. by Anthony
Bonner. Princeton, NJ: Princeton University Press, 1985.</span></a> <i>Ars
brevis</i> 585). </p>';
      break;
case '311':
      echo '<p><span class="volvelles">__</span><p>nbsp;The Third Figure is a
table of the 36 possible non-repeating combinations of the two sets of B
through K represented in the First and Second Figure, and &quot;is meant to
show that any principle can be attributed to any of the others&quot; such
&quot;that the intellect may know each principle in terms of all the others,
and be enabled to deduce many arguments from a single proposition&quot; (<a
href="bibliography.html" target="_blank">Llull<span>Llull, Ramon. <u>Selected
Works of Ramon Llull.</u> Vol. 1. Ed. and trans. by Anthony Bonner.
Princeton, NJ: Princeton University Press, 1985.</span></a> <i>Ars brevis</i>
586). Thus &#39;goodness is great&#39;, &#39;goodness is enduring&#39;,
&#39;goodness is powerful&#39;, &#39;goodness is different&#39;,
&#39;goodness is concordant&#39;, and so on. On this Figure, Llull sets the
condition that one dyad can never be contrary to another, so that all accord
in their conclusion.</p>';
      break;
case '312':
      echo '<p><span class="volvelles">__</span><p>nbsp;The fourth and final
figure of Ramon Llull&#39;s ars is his combinatory mechanism: a set of three
nesting discs, each inscribed with the letters B through K, such that
rotating them against each other generates a three-letter code &mdash; say, B
C D &mdash; and therefore six dyads or (as Llull terms them)
&quot;conditions&quot; that relate, for instance, B to C and B to D, then C
to B and C to D, then D to B and D to C.</p><p>After acquiring these six
conditions, &quot;the intellect acquires another six, by turning the smallest
circle, putting its E where its D was, opposite the C on the middle
circle,&quot; thereby forming the code B C E and a new set of E-related
conditions (<a href="bibliography.html" target="_blank">Llull<span>Llull,
Ramon. <u>Selected Works of Ramon Llull.</u> Vol. 1. Ed. and trans. by
Anthony Bonner. Princeton, NJ: Princeton University Press, 1985.</span></a>
<i>Ars brevis</i> 601). In this way, spinning the discs spits out all
possible combinations of the divine attributes in relation to the order of
the world.</p>';
      break;
case '313':
      echo '<blockquote><p class="blockquote"><span
class="volvelles">__</span><p>nbsp;From what we have said above, the artist
can formulate questions concerning the Hundred Forms and solve them in
```

```
accordance with how the questions are variously treated and derived ... And
thus the intellect learns in what way it can be most general in formulating
many questions and solving them by the method indicated in the Evacuation of
the Third Figure and in the Multiplication of the Fourth Figure. As a result,
who could possibly enumerate all the questions and solutions that could be
thus formulated? (<a href="bibliography.html"
target="_blank">Llull<span>Llull, Ramon. <u>Selected Works of Ramon
Llull.</u> Vol. 1. Ed. and trans. by Anthony Bonner. Princeton, NJ: Princeton
University Press, 1985.</span></a> <i>Ars brevis</i> 643)</p></blockquote>';
      break;
case '314':
      echo '<p><span class="volvelles">__</span><p>nbsp;As Bonner points out,
&#39;ars&#39; is the &quot;usual scholastic translation of the Greek
&tau;&epsilon;&chi;&nu;&eta;, meaning a &quot;technique&quot; &mdash;
&quot;not a body of doctrine, but a system&quot; or &quot;structure&quot; (<a
href="bibliography.html" target="_blank">Bonner<span>Bonner, Anthony.
&quot;Introduction.&quot; <u>Selected Works of Ramon Llull.</u> Vol. 1. Ed.
and trans. by Anthony Bonner. Princeton, NJ: Princeton University Press,
1985.</span></a> 62). Of course, &tau;&epsilon;&chi;&nu;&eta; is also the
root of the English word &#39;technology&#39;. More than a system or
structure, even, Ramon Llull develops a new technology: a tool for managing
and, in turn, automatically manipulating the web of interlocking concepts
evident in the external world.</p>';
      break;
case '315':
      echo '<p><span class="volvelles">__</span><p>nbsp;Today, our
relationship to Llull&#39;s <i>Ars magna</i> is always already refracted
through Renaissance personalities such as Bernard de Lavinheta, Heinrich
Cornelius Agrippa von Nettesheim and Giordano Bruno, who transformed the
Lullian Art into a form of alchemical Hermeticism, and Llull himself into a
Kabbalist mystic. As Rossi writes, &quot;the interest in the cabbala, in
universal and artificial writing, in the discovery of the primary constituent
principles of all possible knowledge, in the art of memory, and the continual
appeal to logic understood as the &#39;key&#39; capable of opening the
secrets of reality: all these themes appeared inextricably connected with the
revival of Lullism in the Renaissance&quot; (<a href="bibliography.html"
target="_blank">Rossi<span>Rossi, Paolo. <U>Logic and the Art of Memory: The
Quest for a Universal Language.</U> Trans. by Stephen Clucas. Continuum
International Publishing Group, 2006.</span></a> 41).</p>';
      break;
case '316':
      echo '<p><span class="volvelles">__</span><p>nbsp;It is not our goal
here to trace Llull&#39;s journey the through fifteenth, sixteenth and
seventeenth centuries; those paths have already been trod (see, for example,
<a href="bibliography.html" target="_blank">Rossi<span>Rossi, Paolo. <U>Logic
and the Art of Memory: The Quest for a Universal Language.</U> Trans. by
Stephen Clucas. Continuum International Publishing Group, 2006.</span></a>,
<a href="bibliography.html" target="_blank">Yates<span>Yates, Frances. <U>The
Art of Memory.</U> Chicago, IL: University of Chicago Press,
1966.</span></a>). Suffice to say the very same figures who imagined
universal languages and encyclopedias to both store and generate knowledge
&mdash; including Heinrich Cornelius Agrippa von Nettesheim and Johann
Heinrich Alsted &mdash; also expressed an interest in the Lullian Art, as did
those who experimented with permutational poetry, such as Bernard de
Lavinheta. In doing so, Renaissance Lullists shrouded  a thirteenth-century
machine for converting Muslims &mdash; that is, for bringing them to the
light &mdash; with dark shades of mysticism; and a technology for cultural
```

```
translation became a mechanism for generation, spitting out not reflections
of the world but text, the language of babble. </p>';
      break;
case '317':
      echo '<blockquote><p class="blockquote"><span
class="volvelles">__</span><p>nbsp;Lullist combinatorics changed its
character in the 17th century from a theological device to a generative
classificatory system of knowledge. Before Diderot and d&#39;Alembert and
their revolutionary reinvention of the encyclopedia in the late 18th century,
knowledge in encyclopedias was not structured arbitrarily by the alphabet,
but in a systematic order of things according to their place in a cosmology.
Lullist combinatorics allowed the generation of complex hierarchical systems
for knowledge classification through the exhaustive combination of
categories. (<a href="bibliography.html" target="_target">Cramer<span>Cramer,
Florian. <U>Words Made Flesh: Code, Culture, Imagination.</U> Piet Zwart
Institute, 2005.</span></a> 39)</p></blockquote>';
      break;
case '318':
      echo '<blockquote><p class="blockquote"><span
class="volvelles">__</span><p>nbsp;These apocryphal writings, and other
commentaries on, or epitomes of, Llull&#39;s Art, tend to abandon Llull&#39;s
formal, exemplarist approach to natural language in favor of a mechanistic
and generative model of discourse more compatible with conventional
rhetorical or dialectical precepts. Rather than establish a system of formal
logic based on explicit ontological categories, they seek to perfect the
discursive machinery of the Lullian program of inquiry. (<a
href="bibliography.html" target="_blank">Johnston<span>Johnston, Mark D.
&quot;The Reception of the Lullian Art, 1450-1530.&quot; <U>Sixteenth Century
Journal</U> 12.1 (1981): 31-48.</span></a> 47-8)</p></blockquote>';
      break;
case '319':
      echo '<p><span class="volvelles">__</span><p>nbsp;It is worth pausing
over at least one seventeenth-century book on the Lullian Art: Athanasius
Kircher&#39;s <i>Ars magna sciendi sive combinatoria</i>, published in 1669,
three years after Leibniz&#39;s dissertation on <i>ars combinatoria</i>.
Kircher &mdash; a student of Chinese characters, cryptography and
hieroglyphics, as well as the author of a polygraphy machine &mdash; set
about reforming Llull&#39;s work by linking each letter of Llull&#39;s
alphabet to an icon, much as Comenius&#39; <i>Orbis Sensualium Pictus</i>
(1658) connects letters and language to icons of the world. In this way, the
Lullian Art not only decodes reflections but itself reflects, generating,
combining and recombining a seemingly infinite series of representations of
the world. </p>';
      break;
case '320':
      echo '<p><span class="volvelles">__</span><p>nbsp;The connection
between the Lullian Art and Kabbalah is strong &mdash; in fact, as Yates
argues, &quot;there was a Cabalist element in Lullism from the start&quot;
(<a href="bibliography.html" target="_blank">Yates<span>Yates, Frances.
<U>The Art of Memory.</U> Chicago, IL: University of Chicago Press,
1966.</span></a> 188). Kabbalah was influential force in thirteenth-century
Spain, and Llull&#39;s system of nine Divine Dignities reflects both
Aristotelian categories and the ten names of God in the Sephiroth. A century
after Llull&#39;s death, Pico della Mirandola explicitly relates the Lullian
Art to Kabbalah (see <a href="bibliography.html"
target="_blank">Yates<span>Yates, Frances. <U>The Art of Memory.</U> Chicago,
IL: University of Chicago Press, 1966.</span></a> 188).</p>';
      break;
```

```
case '321':
      echo '<p><span class="volvelles">__</span><p>nbsp;During the
Renaissance, however, Llull&#39;s reputation as a Kabbalist rested on the
<i>De auditu cabbalistico</i> (1518), a harmony of the Lullian Art and
Kabbalah then thought to be authored by Llull himself. Scholars have since
attributed it to Pietro Mainardi (1456-1529); but, during the sixteenth and
seventeenth centuries, it was included as an authentic Lullian text in one of
the more widely printed anthologies of Llull&#39;s work, Lazarus
Zetzner&#39;s <i>Opera ea quae ad adinventam ab ipso Artem
universalem</i>.</p>';
      break;
case '322':
      echo '<blockquote><p class="blockquote"><span
class="volvelles">__</span><p>nbsp;The figures of his Art, on which its
concepts are set out in the letter notation, are not static but revolving.
One of the figures consists of concentric circles, marked with the letter
notations standing for the concepts, and when these wheels revolve,
combinations of the concepts are obtained. In another revolving figure,
triangles within a circle pick up related concepts. These are simple devices,
but revolutionary in their attempt to represent movement in the psyche. (<a
href="bibliography.html" target="_blank">Yates<span>Yates, Frances. <U>The
Art of Memory.</U> Chicago, IL: University of Chicago Press, 1966.</span></a>
176)</p></blockquote>';
      break;
case '323':
      echo '<p><span class="volvelles">__</span><p>nbsp;As Frances Yates
argues, the Lullian Art was also an art of memory, using a system of places
and constants to store and thereby help its users memorize the order of the
universe. In one manuscript of the <i>Ars demonstrativa</i>, the Figure S is
represented as a hand &mdash; a common figure in the classical art of memory
&mdash; with Llull&#39;s alphabet inscribed along the fingers. </p>';
      break;
case '400':
      echo '<p><span class="volvelles">__</span><p>nbsp;Manuscripts of
Lucretius&#39;s <i>De rerum natura</i> also contain circular diagrams that
perhaps later inspired Peter Abelard (1079-1142) to write one of the earliest
surviving examples of a wheel poem (see <a href="bibliography.html"
target="_blank">Higgins<span>Higgins, Dick. <U>Pattern Poetry: Guide to an
Unknown Literature.</U> Albany: State University of New York Press,
1987.</span></a> 28-9). After Abelard (who, importantly, was predated by Odo
of Paris, said to have written a circular poem in the ninth century),
circular labyrinths saw a revival in Europe during the sixteenth and
seventeenth centuries, sparked in part by the work of the Italian Fortunius
Licetus and Albert Molnar, a Hungarian. Both Fortunius and Molnar compiled
volumes of visual poems, including Latin <i>carmina figurata</i> and ancient
Greek <i>technopaegnia</i>, or poems showing &quot;technical
virtuosity.&quot; Now, of course, we apply the term <i>technopaegnia</i> to
any figure or &quot;pattern&quot; poem that exploits its shape.</p>';
      break;
case '401':
      echo '<p><span class="volvelles">__</span><p>nbsp;Every poem has, in
the words of Dick Higgins, a &quot;formal matrix&quot; in which text and its
physical space interact to make meaning (<a href="bibliography.html"
target="_blank">Higgins<span>Higgins, Dick. <U>Pattern Poetry: Guide to an
Unknown Literature.</U> Albany: State University of New York Press,
1987.</span></a> 206). For traditional verse form, we tend to think of the
poem as lines laid flat in list, each edge neatly trimmed into a rectangle, a
page, a book in miniature. Even nontraditional works, such as
```

```
Mallarm&eacute;&#39;s <i><a href="#">Un coup de d&eacute;s Jamais
N&#39;Abolira Le Hasard<span>A Throw of the Dice will Never Abolish
Chance</span></a></i> (1897), are experimental in so much as they play with
the white spaces of paper and the folded, bound form of the book.</p><p>What,
though, of the poem as a circle? Lines of text bent into infinite loops
&mdash; textual labyrinths, with no beginning and no end?</p>';
      break;
case '402':
      echo '<blockquote><p class="blockquote"><span
class="volvelles">__</span><p>nbsp;Most short poems for instance involve a
significant degree of iconicity: we see the poem as a visual whole before we
read it. Perceived optically as a complete unit the page is qualified to such
an extent that it ceases to function as an arbitrary receptacle, or surface,
for the maximum number of words it can contain (functioning thereby as a
random-sized unit in a larger construct), becoming instead the frame,
landscape, atmosphere within which the poem&#39;s own unity is enacted and
reacted upon. Page and type function as the two ingredients in a verbal
sculpture. (<a href="bibliography.html" target="_blank">Caffrey and
Nichol<span>Caffrey, Steve and bpNichol. <U>Rational Geomancy: The Kids of
the Book-machine: the collected research reports of the Toronto Research
Group, 1973-1982.</u> Vancouver: Talonbooks,  1992.</span></a>
61)</p></blockquote>';
      break;
case '403':
      echo'<p><span class="volvelles">__</span><p>nbsp;In 1663, the Spanish
Cistercian monk Juan Caramuel (1606-1682) published his <i>Primus calamus ob
oculos ponens metametricam ...</i>, known simply as the <i>Metametrica</i>.
It is an unusual and fantastic work of linguistic aesthetics, containing
pattern poems and sound poems, anagrams and permutational poetry,
mathematical language games and a plan for a universal language based on the
work of Caramuel&#39;s contemporary and fellow Jesuit, Athanasius Kircher.
Most interesting for our purposes are his circular labyrinths.</p>';
      break;
case '404':
      echo '<p><span class="volvelles">__</span><p>nbsp;Three circular
labyrinths in Caramuel&#39;s <i>Metametrica</i> &mdash; the &quot;Maria
Stella,&quot; the &quot;Iesus Sol,&quot; and the &quot;Coelum&quot; &mdash;
use the same printed template: a disc embedded in a set of six nested wheels,
the innermost containing twelve phrases, with the number of words around each
subsequent circle increasing by twelve. Moving from the innermost to
outermost circle, the reader builds a line of verse, such as (from the
&quot;Maria Stella&quot;) <i>Parens / Te matrem / Nomina / fontes / Lumina /
quot stellas / Crux quot amara dedit</i>.</p><p>Thus the page presents a
database of isolated words and phrases organized spatially around a set of
circles; only the combinatory actions of the reader construct a poem out of
its 9,644,117,432,715,608 possible lines.</p>';
      break;
case '405':
      echo '<p><span class="volvelles">__</span><p>nbsp;Caramuel&#39;s
<i>Metametrica</i> also contains two other kinds of circular labyrinths. The
first is a set of four wheels of two discs each, as well as phrase in the
center, laid against each other in a diamond. Between all four discs is a
center circle divided into four quadrants, each containing a phrase. By
rotating the wheels against each other, the reader can generate lines of
text.</p><p>The second is an illustration for a text-generation machine: a
cylinder in four parts, each inscribed with a list of words. Turning the
cylinders against each other creates lines of verse, such as <i>Lucifer
terris referat colores</i> and <i>Lucidus coclo reparat triumphos.</i> In
```

```
nine labyrinthine diagrams, Caramuel abstractly depicts all possible line
combinations.</p>';
      break;
case '406':
      echo '<p><span class="volvelles">__</span><p>nbsp;Although Caramuel
describes his circular labyrinths in terms of the <i>ars magna</i> of
&amp;quot;Raymundus,&amp;quot; Ramon Llull, in fact none of his poems physically
rotate. Yet they are an important bridge between the permutational poems of
Quirinus Kuhlmann (or indeed of Caramuel himself) and the spinning volvelles
of G. P. Harsd&amp;ouml;rffer; for unlike proteus verse, Caramuel&amp;#39;s distichs
do not exist without the intervention of the reader. By exploiting the
spatial dimensions of the circle, infinite loops, Caramuel imagines a
mechanism for storing, retrieiving and combining textual information.</p>';
      break;
case '408':
      echo '<p><span class="volvelles">__</span><p>nbsp;Caramuel was not the
only seventeenth-century poet to produce circular poems. For instance, in
1666 Duke Christian Albrecht (1641-94) produced a wheel poem that radiates
lines of text out from a center point like the spokes of a wheel (see <a
href="bibliography.html" target="_blank">Higgins<span>Higgins, Dick.
<U>Pattern Poetry: Guide to an Unknown Literature.</U> Albany: State
University of New York Press, 1987.</span></a> 44). Star- and sun-shaped
pattern poems, popular during the seventeenth-century, engage with the space
of the page in a similar manner (see, for example, Hermannus de Santa
Barbara&amp;#39;s acrostic sun-shaped poem (1687), pictured on <a
href="bibliography.html" target="_blank">Higgins<span>Higgins, Dick.
<U>Pattern Poetry: Guide to an Unknown Literature.</U> Albany: State
University of New York Press, 1987.</span></a> 47). Most interesting for our
purposes are the combinatory, Kabbalistic diagrams that recur in the work of
pattern poets, conceptualizing cosmological relationships through bits of
text inscribed around the edge of nested circular diagrams. </p>';
      break;
case '409':
      echo '<p><span class="volvelles">__</span><p>nbsp;Athanasius
Kircher&amp;#39;s &amp;quot;Kabbalistic tree&amp;quot; is now the most well-known of the
mystical, combinatory diagrams. Appearing in his <i>Oedipus aegypticus</i> of
1652, the diagram takes the shape of a sunflower of nesting rings radiating
outward, aesthetically influenced by the Lullist and Lullist-derived works of
the generations before him. The center illustrates the Tetragrammaton, the
ineffable four-letter name of God; the smallest rings illustrate His twelve-
letter name; then His forty-two-letter name; and finally His seventy-two
names, iterated through the seventy-two nations of humanity. As Stolzenberg
points out, in this diagram &amp;quot;Kircher places the claim that all the
nations of the world possess a divinely-inspired four-letter name of God on
par with the classic argument about the wonder-working, five-letter name of
Christ and its identity with the Tetragrammaton&amp;quot; (<a
href="bibliography.html" target="_blank">Stolzenberg<span>Stolzenberg,
Daniel. &amp;quot;Four Trees, Some Amulets, and the Seventy-two Names of God:
Kircher Reveals the Kabbalah.&amp;quot; <U>Athanasius Kircher: The Last Man Who
Knew Everything.</U> Ed. by Paula Findlen. New York: Routledge, 2004.
149.</span></a> 156-7; see Stolzenberg for a more thorough description and
contextualization).</p>';
      break;
case '410':
      echo '<p><span class="volvelles">__</span><p>nbsp;Although deeply
embedded in a combinatory, Kabbalistic understanding language as a reflection
of the world&amp;#39;s order, Athanasius Kircher&amp;#39;s circular depiction of the
seventy-two names of God is not itself explicitly mechanical, since the
```

wheels function as diagrams, not volvelles. For his most famous text-generating mechanism, we must turn to a later work, his Polygraphia nova et universalis, published in 1663, the same year as Caramuel's Metametrica. ';
	break;
case '411':
	echo ' __ nbsp;Although published three years before Leibniz's Dissertatio de Arte Combinatoria, Kircher's Polygraphia is structured using what might anachronistically be called Leibniz's model for a mathesis universalis: that is, it begins with Section I, "The Reduction of All Languages to One," which encodes words in a two-bit symbol; then moves to Section II, "The Extension of One Language to All," which decodes text to produce entire encyclopedias from a single poem; and ends with "a Techno-logica; or, a universal Steganographic Secret operating by combinations of things" (see SaussySaussy, Haun. "Magnetic Language: Athanasius Kircher and Communication." <U>Athanasius Kircher: The Last Man Who Knew Everything.</U> Ed. by Paula Findlen. New York: Routledge, 2004. 263. 263). In other words, operating within the programmatic epistemology of the seventeenth-century, Kircher begins first by reducing knowledge to a set of universal primitives, then mechanically permutes those constants to generate new knowledge. ';
	break;
case '412':
	echo '<blockquote> __ nbsp;Language for Kircher partakes of two domains: a sympathetic network linking the particles of the cosmos together, and a technology of language based on the shuffling of letters. (SaussySaussy, Haun. "Magnetic Language: Athanasius Kircher and Communication." <U>Athanasius Kircher: The Last Man Who Knew Everything.</U> Ed. by Paula Findlen. New York: Routledge, 2004. 263. 265) </blockquote>';
	break;
case '413':
	echo ' __ nbsp;Like Leibniz, Harsdörffer and Kuhlmann, Athanasius Kircher, too, was a German working within an Empire of culturally, linguistically and religiously disparate states recently torn apart by the Thirty Years War. From this angle, his desire to find a machine for generating a universal language for communication — a project commissioned by the Hapsburg Emperor Ferdinand III (see SaussySaussy, Haun. "Magnetic Language: Athanasius Kircher and Communication." <U>Athanasius Kircher: The Last Man Who Knew Everything.</U> Ed. by Paula Findlen. New York: Routledge, 2004. 263. 268) — was not a symptom of baroque quirkiness but of the very real challenges facing intellectuals within a heterogeneous environment. For if reading and writing become mechanical practices, then communication becomes the material manipulation of language, a form of nonalphabetic, programmatic literacy. ';
	break;
case '414':
	echo ' __ nbsp;Athanasius Kircher's Polygraphia nova is one of dozens, possibly hundreds of universal language projects from the seventeenth century. Yet, unlike many of its predecessors and successors, Kircher's project takes a technical turn: he invents a chest, his "Glottotactic Ark," filled with slats of wood inscribed with letters. Much the way a cryptographic volvelle could

```
be used to encode and decode ciphers, the wooden slats or tablets of
Kircher&#39;s ark could be used to mechanically permute and thereby translate
languages.</p><p>As Saussy points out, only Kircher&#39;s patrons were given
these combinatory text-generating machines; it survives now in the form of
diagrams and descriptions within his <i>Polygraphia nova</i> (<a
href="bibliography.html" target="_blank">Saussy<span>Saussy, Haun.
&quot;Magnetic Language: Athanasius Kircher and Communication.&quot;
<U>Athanasius Kircher: The Last Man Who Knew Everything.</U> Ed. by Paula
Findlen. New York: Routledge, 2004. 263.</span></a> 270-1).</p>';
      break;
case '415':
      echo '<p><span class="materiality">__</span><p>nbsp;Kircher explicitly
divorces his mechanical text-generating device from the printed word. As he
writes, while attempting to find a common root to all languages,
suddenly</p><blockquote><p class="blockquote">the same thing happened to me
that might happen to a typesetter who has several pages of type laid out and
ready for putting under the press: by some inexplicable chance the bonds
dissolve and the letters rain down onto the floor, retaining no trace of
their true former meaning and no longer capable of being brought back to
their lost prototype. (Kircher <i>Turris Babel</i> 218, quoted in <a
href="bibliography.html" target="_blank">Saussy<span>Saussy, Haun.
&quot;Magnetic Language: Athanasius Kircher and Communication.&quot;
<U>Athanasius Kircher: The Last Man Who Knew Everything.</U> Ed. by Paula
Findlen. New York: Routledge, 2004. 263.</span></a>
270)</p></blockquote><p>Thus type, laid out linearly through a process of
accretion, could not easily accommodate Kircher&#39;s programmatic
epistemology of reduction and mechnical permutation.</p>';
      break;
case '416':
      echo '<blockquote><p class=&quot;blockqoute&quot;><span
class="materiality">__</span><p>nbsp;The lesson of the machine is that no
matter how marvelous, it is still not miraculous; no matter how many
combinations a finite set of elements cn produce, its number still falls
infinitely short of infinity; and thus the triumphant display of large
numbers stands for the exhaustion of language as much as for its fecundity.
It is a melancholy Lullism. (<a href="bibliography.html"
target="_blank">Saussy<span>Saussy, Haun. &quot;Magnetic Language: Athanasius
Kircher and Communication.&quot; <U>Athanasius Kircher: The Last Man Who Knew
Everything.</U> Ed. by Paula Findlen. New York: Routledge, 2004.
263.</span></a> 276-7).</p></blockquote>';
      break;
case '417':
      echo'<blockquote><p class="blockquote"><span
class="materiality">__</span><p>nbsp;Since the tablets may be moved about as
many times as there are combinations of the letter sof the alphabet, it is
clear, then, that there can be no end to this undertaking, as is demonstrated
by the number 25852016738884976666400000 which represents the number of
combinations of the 24 letters of the alphabet. ... Surely there is no
conceivable sentence occurring in any language which cannot be represented on
the tablets; thus this narrow box and the letters enclosed therein suprass
all the libraries of the whole world. (Kircher 151, quoted in <a
href="bibliography.html" target="_blank">Saussy<span>Saussy, Haun.
&quot;Magnetic Language: Athanasius Kircher and Communication.&quot;
<U>Athanasius Kircher: The Last Man Who Knew Everything.</U> Ed. by Paula
Findlen. New York: Routledge, 2004. 263.</span></a> 272)</p></blockquote>';
      break;
case '418':
```

```
      echo '<p><span class="materiality">__</span><p>nbsp;Of course, Kircher
and other Lullists of the seventeenth-century were later skewered by Jonathan
Swift in <i>Gulliver&#39;s Travels</i> (1726). At the Academy of Lagado,
Gulliver encounters a frame containing a grid of wooden cubes with language
inscribed on them. By turning the handles of the frame, the pupils of the
Academy form lines dictated to scribes who form the text into books. As Swift
writes, &quot;Six hours a day the young students were employed in this
labour; and the professor showed me several volumes in large folio, already
collected, of broken sentences, which he intended to piece together, and out
of those rich materials, to give the world a complete body of all arts and
sciences&quot; (<a href="bibliography.html" target="_blank">Swift<span>Swift,
Jonathan. <U>Gulliver&#39;s Travels: Complete, Authoritative Text with
Biographical and Historical Contexts. New York: Palgrave Macmillan,
1995.</span></a> III.5).</p>';
      break;
case '419':
      echo'<p><span class="materiality">__</span><p>nbsp;Kircher&#39;s
<i>Polygraphia nova</i> is often (and rightly) situated within the web of
universal language project proliferating across Europe during the same
period. Yet what if, instead of linking Kircher&#39;s project to abstract
theories of language, we link it to a notion of literacy &mdash; in other
words, to a site of human interaction with a material instance of language?
From this angle, Kircher does not  &mdash; or, more accurately, not
<i>only</i> &mdash; aim to form a new grammar or lexicon, but seeks a new
method of reading and writing based on the material manipulation of slotted
bits of wood. In other words, like Leibniz&#39;s binary system, Kircher&#39;s
technical polygraphy is a read/write system, such that knowing the abstract
logic behind its function equates with knowing all of its possible
outcomes.</p>';
      break;
case '1000':
      echo '<a href="#"><img src="images/thumbnails/denckring_thumbnail.JPG"
/><span>Harsd&ouml;rffer&#39;s  <i>F&uuml;nffacher Denckring der Teutschen
Sprache</i>, or  Five-fold  Thought-ring of the German Language, from the
<i>Delici&aelig; Physico-Mathematic&aelig;, Oder, Mathemat. Und
Philosophische Erquickstunden...</i> (1651); image from the Beinecke Library
at Yale.</span></a>';
      break;
case '1001':
      echo '<a href="#"><img
src="images/thumbnails/denckring_bottomleft_thumbnail.JPG" /><span>The bottom
left and top right corners of Harsd&ouml;rffer&#39;s  <i>F&uuml;nffacher
Denckring der Teutschen Sprache</i>, or  Five-fold  Thought-ring of the
German Language, from the <i>Delici&aelig; Physico-Mathematic&aelig;, Oder,
Mathemat. Und Philosophische Erquickstunden...</i> (1651); image from the
Beinecke Library at Yale.</span></a>';
      break;
case '1002':
      echo '<a href="#"><img
src="images/thumbnails/denckring_topleft_thumbnail.JPG" /><span>The top left
and bottom right corners of Harsd&ouml;rffer&#39;s  <i>F&uuml;nffacher
Denckring der Teutschen Sprache</i>, or  Five-fold  Thought-ring of the
German Language, from the <i>Delici&aelig; Physico-Mathematic&aelig;, Oder,
Mathemat. Und Philosophische Erquickstunden...</i> (1651); image from the
Beinecke Library at Yale.</span></a>';
      break;
case '1003':
```

```
      echo '<a href="http://diglib.hab.de/drucke/ko-
108/start.htm?image=00005" target="_blank"><img
src="images/thumbnails/aristarchus_titlepage_thumbnail.jpg" /><span>The title
page of Martin Opitz&#39;s <i>Aristarchus; sive, de contemptu linguae
Teutonicae</i> (1618); image from the Herzog August Bibliothek.</span></a>';
      break;
case '1004':
      echo '<a href="http://diglib.hab.de/drucke/ko-
108/start.htm?image=00017" target="_blank"><img
src="images/thumbnails/aristarchus_2_thumbnail.jpg" /><span>From Martin
Opitz&#39;s <i>Aristarchus; sive, de contemptu linguae Teutonicae</i> (1618);
image from the Herzog August Bibliothek.</span></a>';
      break;
case '1005':
      echo '<a href="http://diglib.hab.de/drucke/ko-
108/start.htm?image=00027" target="_blank"><img
src="images/thumbnails/aristarchus_3_thumbnail.jpg" /><span>From Martin
Opitz&#39;s <i>Aristarchus; sive, de contemptu linguae Teutonicae</i> (1618);
image from the Herzog August Bibliothek.</span></a>';
      break;
case '1006':
      echo '<a href="http://diglib.hab.de/drucke/ko-
108/start.htm?image=00029" target="_blank"><img
src="images/thumbnails/aristarchus_4_thumbnail.jpg" /><span>From Martin
Opitz&#39;s <i>Aristarchus; sive, de contemptu linguae Teutonicae</i> (1618);
image from the Herzog August Bibliothek.</span></a>';
      break;
case '1007':
      echo '<a href="#"><img
src="images/thumbnails/schottel_portrait_thumbnail.jpg" /><span>A portrait of
Justus Georg Schottelius (1612-1676).</span></a>';
      break;
case '1008':
      echo '<a href="#"><img
src="images/thumbnails/brueghel_tower_thumbnail.jpg" /><span>Pieter Brueghel
the Elder&#39;s <i>Tower of Babel</i> (1563).</span></a>';
      break;
case '1009':
      echo '<a href="#"><img src="images/thumbnails/letters_A_thumbnail.JPG"
/><img src="images/thumbnails/letters_R_thumbnail.JPG" /><span>Images from a
visual alphabet in G. P. Harsd&ouml;rffer&#39;s <i>Delici&aelig; Physico-
Mathematic&aelig;, Oder, Mathemat. Und Philosophische Erquickstunden...</i>
(1651); image from the Herzog August Bibliothek.</span></a>';
      break;
case '1010':
      echo '<a href="#"><img src="images/thumbnails/letters_K_thumbnail.JPG"
/><img src="images/thumbnails/letters_U_thumbnail.JPG" /><span>Images from a
visual alphabet in G. P. Harsd&ouml;rffer&#39;s <i>Delici&aelig; Physico-
Mathematic&aelig;, Oder, Mathemat. Und Philosophische Erquickstunden...</i>
(1651); image from the Herzog August Bibliothek.</span></a>';
      break;
case '1011':
      echo '<a href="http://www.swilliams.ca/bibles/" target="_blank"><img
src="images/thumbnails/harmonies_thumbnail.png" /><span>From a Little Gidding
Harmony.</span></a>';
      break;
case '1012':
```

```
      echo '<a
href="http://www.mirroroftheworld.com.au/inspiration/printed/schatzbehalter.p
hp" target="_blank"><img src="images/thumbnails/schatzbehalter_thumbnail.jpg"
/><span>From a Stephan Fridolin&#39;s <i>Schatzbehalter</i> (1491); image
from the the State Library of Victoria</a>.</span></a>';
      break;
case '1013':
      echo '<a href="http://en.wikipedia.org/wiki/File:Jbulwer.jpg"
target="_blank"><img src="images/thumbnails/chirologia_thumbnail.jpg"
/><span>From John Bulwer&#39;s <i>Chirologia</i>; image from
Wikimedia.</span></a>';
      break;
case '1014':
      echo '<a
href="http://www.robertsabuda.com/everythingpopup/suzannekarr2.asp"
target="_blank"><img src="images/thumbnails/pointingfinger_thumbnail.jpg"
/><span>From Giambattista della Porta&#39;s <i>De occvltis literarvm
notis</i> (1593).</span></a>.)</p>';
      break;
case '1015':
      echo '<a href="#"><img
src="images/thumbnails/comenius_orbis_fire_thumbnail.JPG" /><span>An image
from Comenius&#39; <i>Orbis Sensualium Pictus</i> (1658).</span></a>';
      break;
case '1016':
      echo '<a href="#"><img
src="images/thumbnails/Europe_map_1648_thumbnail.PNG" /><span>A political map
of Europe circa 1648.</span></a>';
      break;
case '1017':
      echo '<a href="#"><img
src="images/thumbnails/leibniz_calculator_thumbnail.jpg" /><br /><p><span>A
model of Leibniz&#39;s calculator.</span></a>';
      break;
case '1019':
      echo '<a href="#"><img src="images/thumbnails/iching_thumbnail.PNG"
/><span>A representation of the Chinese I-Ching.</span></a>';
      break;
case '1020':
      echo '<a href="#"><img src="images/thumbnails/gysin_thumbnail.jpg"
/><span>A portion of Brion Gysin&#39;s permuted poem, &quot;RUB OUT THE
WORD&quot;.</span></a>';
      break;
case '1021':
      echo '<a href="#"><img
src="images/thumbnails/llull_arsdem_thumbnail.gif" /><span>Image from
Llull&#39;s <i>Ars demonstrativa</i>; image from the Biblioteca Nazionale
Marciana.</span></a>';
      break;
case '1022':
      echo '<a href="#"><img
src="images/thumbnails/llull_arsdem_s_thumbnail.gif" /><span>Image from
Llull&#39;s <i>Ars demonstrativa</i>; image from the Biblioteca Nazionale
Marciana.</span></a>';
      break;
case '1023':
      echo '<a href="#"><img
src="images/thumbnails/llull_arsdem_A_thumbnail.gif" /><span>Image from
```

```
Llull&#39;s <i>Ars demonstrativa</i>; image from the Biblioteca Nazionale
Marciana.</span></a>';
      break;
case '1024':
      echo '<a href="http://diglib.hab.de/drucke/6-3-quod-
2f/start.htm?image=00033" target="_blank"><img
src="images/thumbnails/kircher_arsmagna_T_thumbnail.jpg" /><span>The Second
Figure of Llull&#39;s <i>Ars magna</i>, as represented in Kircher&#39;s
<i>Ars magna sciendi sive combinatorica</i> (1669); image from the Herzog
August Bibliothek.</span></a>';
      break;
case '1025':
      echo '<a href="http://diglib.hab.de/drucke/6-3-quod-
2f/start.htm?image=00036" target="_blank"><img
src="images/thumbnails/kircher_llull_3_thumbnail.jpg" /><span>The Third
Figure of Llull&#39;s <i>Ars magna</i>, as represented in Kircher&#39;s
<i>Ars magna sciendi sive combinatorica</i> (1669); image from the Herzog
August Bibliothek.</span></a>';
      break;
case '1026':
      echo '<a href="http://diglib.hab.de/drucke/6-3-quod-
2f/start.htm?image=00039" target="_blank"><img
src="images/thumbnails/kircher_llull_4_thumbnail.jpg" /><span>The build-your-
own version of the Fourth Figure of Llull&#39;s <i>Ars magna</i>, as
represented in Kircher&#39;s <i>Ars magna sciendi sive combinatorica</i>
(1669); image from the Herzog August Bibliothek.</span></a>';
      break;
case '1027':
      echo '<a
href="http://www.newcastle.edu.au/service/archives/digitalscriptorium/arsmem/
index.html" target="_blank"><img src="images/thumbnails/arsmem_thumbnail.jpg"
/><span>Teaching the art of memory using the hand; from a German incunabula,
circa 1490; image from the University of Newcastle Cultural Collections
Unit.</span></a>';
      break;
case '1028':
      echo '<a href="#"><img
src="images/thumbnails/caramuel_mariastella_thumbnail.png"
/><span>Caramuel&#39;s &quot;Maria Stella&quot;, from the <i>Metametrica</i>
(1663).</span></a>';
      break;
case '1029':
      echo '<a href="#"><img
src="images/thumbnails/caramuel_cylinder_thumbnail.png"
/><span>Caramuel&#39;s text-generating cylinder, from the <i>Metametrica</i>
(1663).</span></a>';
      break;
case '1030':
      echo '<a href="#"><img
src="images/thumbnails/kircher_namesofgod_thumbnail.png"
/><span>Kircher&#39;s circular, Kabbalistic diagram showing the seventy-two
names of God.</span></a>';
      break;
case '1031':
      echo '<a href="#"><img
src="images/thumbnails/kircher_music_box_thumbnail.png" /><span>A music box
for automatically generating melodies, from Kircher&#39;s <i>Musurgia
```

```
Universalis</i> (1650). His &quot;Glottotactic Ark&quot; for translating
languages looks very similar to this.</span></a>';
      break;
case '1032':
      echo '<blockquote><p class="blockquote"><i>Possible Epigraphs for the
Soul</i> (from Charles O. Hartman&#39;s &quot;Monologues of Soul &amp;
Body&quot;, partially written using the computer program TRAVESTY)</p><p
class="blockquote">&quot;Little by little&quot; &mdash; this is Maeterlinck
&mdash;<br/>&quot;the years teach every man that truth alone<br/>is
marvelous.&quot; Fabulous old fraud.</p></blockquote>';
      break;
case '1033':
      echo '<a href="#"><img
src="images/thumbnails/kircher_icons_thumbnail.jpg" /><span>From
Kircher&#39;s <i>Ars magna sciendi sive combinatorica</i> (1669). Kircher
recommended the use of icons for the different categories within the Lullian
art.</span></a>';
      break;
 case '1034':
      echo '<blockquote><p class="blockquote"><i><b>Auf</b></i> Nacht / Dunst
/ Schlacht / Frost / Wind / See / Hitz / S&uuml;d / Ost / West / Nord / Sonn
/ Feur / und <i>Plagen</i> /<br/><br/><i><b>Folgt</b></i> Tag / Glantz /
Blutt / Schnee / Still / Land /Blitz / W&auml;rmd / Hitz / Lust / K&auml;lt /
Licht / Brand / und <i>Noth</i>:<br/><br/><i><b>Auf</i></b> Leid / Pein /
Schmach / Angst / Krig / Ach / Kreutz / Streit / Hohn / Schmertz / Qual /
T&uuml;kk / Schimpf / als <i>Spott</i> /</p></blockquote>';
      break;
 };?>
```