

Index

A

- @ character, array names beginning with, 64
- Access, and files ending in .wml, 134
- Access control
 - with .htaccess, 50–55
 - mod_perl and enhancement of, 272
 - and mod_perl programming, 243
- AccessFileName directive, 43
- access_log, 254, 272
- Accounts, Linux, 18–19
- a.cgi, mod_perl used for executing, 232–233
- a.cgi program, 6
- action= form attribute, 387
- \$action, 366
- address field, in manufacturers table, 304
- Advanced Package Tool (APT), 12
- age.cgi, 223
- Age information
 - before adding new entry, 228
 - after adding new entry, 229
- age_information table, 223
 - in MySQL, 112
 - records in, 124
- AIDE program, 30
- Aliased commands, 27
- AllowOverride directive, 190
- Ampersand (&), in format of query string, 208
- and operator, in PHP, 391
- a option, for building WML files in directory, 136

Note: Italicized page locators refer to figures.

- Apache, xxix, xxviii, xxx, 10, 34, 37, 130, 187, 264, 310, 374
 - configuration files, 271
 - configuration of, 39–40, 190–191, 355–356
 - Embperl and configuration of, 289–291
 - EmbperlObject and configuration file in, 311
 - functionality added to normal logging phase of, 268
 - and HelloWorldPerl module, 245
 - Mason and configuration of, 334–335
 - mod_perl and configuration of, 233–234, 235
 - and note about mod_perl, 246
 - and PerlLogHandler lines, 272
 - and PHP, 379
 - and root account, 19
- Server Side Includes processed by, 277
- and SSI configuration, 278
- starting, stopping, and restarting, 36–39
- versatility with, 55
- welcome page, 36
 - and WML chapter, 134
- Apache::Constants module, 244
- Apache::DBI, 328
- Apache environment variables, Mason access to, 338
- Apache log files and access control with .htaccess, 50–55
- Apache::PerlRun, 242–243
- Apache::Registry, 231, 235–237, 241
- Apache::Request, 249–251, 287

- Apache security, 41–45
 - access to local documentation, 42
 - disable CGI programs, 45
 - disallow directory indexes, 44
 - disallow public_html web sites, 42
 - disallow symbolic links, 44
 - .htaccess, 43
 - proxy server warning, 44–45
 - reload configuration file, 45
 - remove online manuals, 41–42
 - server-status and server-info removal, 43
 - set user and group, 41
 - Apache Server Side Includes, 272
 - Apache::Session, 332
 - apache user
 - and DBI module, 123
 - mysql database, 120, 121
 - permission granted to, 256
 - privileges with PHP pages, 421
 - Apache Web Server, 5, 33–55, 231
 - Apache web site creation, 46–49
 - Applets, 4, 9
 - Application Program Interface (API), 109
 - apt-get mechanism, 12
 - \$arg, 374
 - <%args> tag, 350–353, 358
 - %ARGS, posted data handled with, 343–344
 - <%args ... %args> tag, posted data grabbed with, 344–348
 - Arguments
 - concatenating with CGI.pm HTML shortcut, 200
 - passing to PHP functions, 401–404
 - Arithmetic operators, Perl, 71–72
 - \$array, 431
 - Array elements
 - accessing, 65
 - and Embperl commands, 292
 - array() function, 385, 386
 - Array functions, 66–67
 - in PHP, 406–410
 - array_keys() function, in PHP, 409
 - array_pop() function, in PHP, 406
 - array_push() function, in PHP, 406
 - array_reverse() function, in PHP, 409–410
 - Arrays
 - associative, 67, 384, 386, 396, 397, 416
 - enumerated, 384, 385, 396, 397, 407, 412
 - array_unshift() function, in PHP, 406
 - array_values() function, in PHP, 409
 - Array variables, 64–66
 - files read into, 97
 - reading input into, 94
 - Arrow notation, methods invoked with, 105
 - arsort() function, in PHP, 407
 - asort() function, in PHP, 407
 - Assignable lists, 106
 - Assignment operator (=), 73
 - Associative arrays, 67
 - fetching row as, 416
 - in PHP, 384, 386, 396, 397
 - Asterisk (*)
and navigation bar definition, 153
and SELECT command, 115
as wildcards, 26, 159
 - Attributes
 - for multiple cells, 159
 - in navigation bar definition, 153
 - tag, 172
 - %attributes variable, 172
 - Authentication
 - and Embperl, 287
 - and mod_perl, 243, 272
 - \$author variable, in WML template file, 141
 - Autodecrement operator (-), 75
 - auto_increment integer, 420
 - Autoincrement operator (++), 75
 - awk, 28, 57, 87
- B**
- Background color
 - in Embperl project, 323
 - and readability, 264, 323
 - setting in CGI.pm program, 199
 - in page tag, 148
 - in WML template files, 140–141, 142
 - Backquotes, 104
 - Backticks, 64
 - commands bracketed by, 195
 - Backups, in Linux, 14, 15
 - Bandwidth, xxvii
 - Banner ads, getting rid of, 30
 - Banners, 187
 - Bare page, with Mason project
 - components, 360
 - Base files, 311, 312

- base.html, 311, 312, 313, 315, 316, 325
bash, 22, 27
Benchmark module, 239, 240
Berners-Lee, Tim, 33*n*
better1.html, HTML file in, 166–167
BIGINT, in MySQL, 113
Binary Large Object (BLOB), 113, 114, 225
/bin directory, 28
bk2site program, 30
Blank lines
 after mail header, 271
 in CGI programs, 188, 189, 190, 192, 195
Body, with CGI program, 188, 189, 190
<body> tag, 142, 147, 148
book_information table, 256
BookListingMysql.pm, 265
book.pl program, flat file of book
 information in, 257
Books database
 book information processed in, 263–264
 and MySQL functions, 412–415
 and PHP mysql example 3, 417–418, 419
/boot directory, 29
Borders, for tables, 159
.bottom component, in Mason project, 359, 361, 363
bottom() function, 308, 309
bottom.html, 223, 259, 260, 266, 316
Brackets ([]), Embperl commands starting
 and ending with, 292
bread_crumb() function, 320
Bread crumbs, 179
Bread crumb strings, 320, 325
Bread crumb text, 319
\$bread_crumb_text, 320
Bread crumb trails, 176
 generating for www.opensourcewebbook.com, 180–181
 in Mason project, 358
break, in PHP switch statement, 393
Browsers, 3, 35, 189
BSD, xxxii, xxxiii
BSD-type Open Source license, 33
Buffer overflows, 221
Bugs, xxix
 security, xxxi
 and template files, 165
Bugtraq, xxxi
- Built-in functions in PHP, 405–411
 array functions, 406–410
 important functions, 405–406
 string functions, 410–411
button(), for form widget creation, 218
Byte code, 60
bytes_sent(), 270
- C**
- C, 61, 377, 382, 391
C++, 377, 382, 391
Caching, with Mason, 331
Camel Book, 57, 81, 106
Caret character (^), 83
Case sensitivity
 with PHP, 381
 and SQL commands/subcommands, 111
category field, in products table, 304
cd ~/bin command, 25
cd ./bin, 25
cd command, standard shortcuts, 25–26
Cells, for tables, 158
CERN, 33*n*
CGI. *See* Common Gateway Interface
cgi-bin, removing CGI scripts in, 218
cgi-bin directory, 190
CGI data, handling with mod_perl using
 Apache::Request, 251
CGI.pm, 105, 187, 198
 introduction to, 197–200
CGI.pm HTML shortcuts, 200–202
 and named parameters *versus* ordered
 arguments, 202
 nesting of, 201
 using, 200–201
CGI programming, 105
CGI programs
 disabling with Apache, 45
 downloading advice for, 188
 information received by, 202–216
 path information received by, 205–206
 posted form data received by, 206–216
 session information received by, 202–204
CGI script, and exec directive, 283
\$CGIURL variable, 238
Character and string substitution phase
 (Pass 6), in WML, 137
Character classes, in Perl, 82–83

Characters, in Perl, 82
 Character strings, in MySQL, 112
 CHAR data type, in MySQL, 113
 Check box, in posted form data, 211
 checkbox(), for widget creation, 216
 checkbox_group, for form widget creation, 218
 check_man_id(), in Mason project, 364, 365
 check_prod_id() function, in Mason, 367
 Checks, for CGI/MySQL/DBI program, 226
 chkconfig, as root, 110
 chkconfig command, 27
 chmod, 23
 chomp() function, 95, 98, 126, 258
 chomping, of returned newlines, 195
 chown command, 23
 chsh command, 22
 city field, in manufacturers table, 304
 Cleanup tags, in Mason, 372–373
 CLI. *See* Command line interface
 Client, and Apache log, 40
 Client-server relationship, 10
 Client-side browser, 133
 Closed-source software, xxxiii
 Closed systems, xxix
 close() function, 97
 Closing tags, in HTML, 170, 171
 Cluster software, 18
 cmp, 74
 \$cnt variable, in Embperl, 328
 Code, spacing dialog inside, 223
 \$col, and dynamic table generation, 301, 303
 Colons, value separated by, 153
 Color, background, 140–141, 142, 148, 199, 264, 323
 Columns, 159
 combined format, 40
 combined output, 50
 [\$ Command Arguments \$] Embperl
 command, 295–298
 Command line interface, 15
 Linux used through, 20, 21
 Commands
 Embperl, 292–299
 and SSI, 277
 Comment blocks, in Embperl, 295
 Comments
 in hello, world program, 59
 in PHP, 382
 with WML, 144–146
 comment style, 144
 COM1, 3
 Common Gateway Interface, 4, 122, 133, 188–190, 189, 273
 Apache configuration, 190–191
 CGI/MySQL/DBI project, 223–230
 and CGI.pm, 197–200
 CGI.pm HTML shortcuts, 200–202
 for dynamic website creation, 306
 errors with programs in, 196–197
 execution, 232
 first program with, 191–196
 form widget methods, 216–218
 information, 204
 information received by CGI
 program, 202–216
 mod_perl *versus*, 231
 note about die(), 222
 path information, 206
 security considerations, 218–222
 Compare operator (=), 73, 74
 Comparison operators, in PHP, 390
 Complex data types, 107
 Component class API, in Mason, 373
 Components, in Mason, 335
 Comprehensive Perl Archive Network, 45, 59, 105, 107, 328, 378
 COM2, 3
 Conditional constructs, 77–81, 337
 Conditional expressions, and SSI, 277
 Conditional statements, Mason, 340
 config directive, 281, 284–285
 Configuration file, reloading with
 Apache, 45
 connect() method, 123, 124
 content_type(), 244
 continue statement, in PHP, 395
 \$_COOKIE, 390
 Cookies
 controlling, 30
 distrust of, 221
 Cool.html file, EmbperlObject example, 313, 314
 count() function, in PHP, 408
 CPAN. *See* Comprehensive Perl
 Archive Network
 CPAN module, 106
 cp command, 26
 Crackers, 219, 220, 307, 355, 356

- Cracking, 9
 - hacking *versus*, xxx
- CREATE DATABASE command,
 - in MySQL, 111, 121
- CREATE TABLE command,
 - in MySQL, 112, 122
- csh, 22
- CUPS, 30
- Curly braces ({}), 69, 77
 - for bodies of PHP constructs, 391
 - in PHP multiline bodies, 382
- Customization, with mod_perl, 231
- CustomLog options, with Apache, 40
- Custom tags, 170, 184
- CVS, xxxii, xxxiii
- D**
- Daemons, 23, 33
- Darwin, xxxiii
- Dash (-), for specifying a range, 82
- Data
 - checking for security
 - reasons, 219–220
 - and table joins, 127
- Database handle, 123
- DataBase Independent interface, 109, 122–127, 130
 - hooking into CGI program, 223
 - ISBN retrieval with, 262–263
 - and Mason project, 368
- Databases
 - connecting to in Perl, 122
 - loading and dumping, 129–130
 - in MySQL, 111
 - querying selected, 416
 - and SELECT command, 115
 - selecting, 416
- Data directory, Mason, 335
- \$DATA_LOCAL variable, 285
- Data types
 - examples in PHP, 384
 - in MySQL, 112
 - in PHP, 383
- DATE, in MySQL, 113
- Date
 - and Apache log, 40
 - display of in CGI program, 194
 - and SSI, 277
- \$date, 201
- date field, in journal MySQL database, 420
- DATE_GMT variable, and echo directive, 279
- DATE_LOCAL variable, and echo directive, 279
- DATETIME, in MySQL, 113
- \$dbh value, 224, 259
- DBI. *See* DataBase Independent interface
- DBI method, 123
- DBI module, for book listing, 265
- D character, and standard input, 95
- dd(), in CGI.pm, 200
- Debian GNU/Linux, 12
- DECIMAL, in MySQL, 113
- Decrement operators (--), in Perl, 75–76
- Default arguments, providing to functions in PHP, 402
- defaults(), for form widget creation, 218
- Default values, for variables assigned with posted data, 212
 - <define-tag> macro, 170
- DELETE command, in MySQL, 118–119, 122
- DELETE MySQL query, 416
- DELETE query, and DBI, 122
- DESCRIBE command, in MySQL, 114, 122
- description field, in products table, 304
- Design, of navigation bars, 152
- /dev directory, 29
- df command, 26
- dhandlers (or default handlers), and Mason, 374, 375
- die, scripts and, 123, 124
- die() function, 96
 - and book database, 258
 - note about, 222
 - in PHP, 405, 406
 - and %args ... %args tag, 344, 347
- “Digital divide,” xxvii
- Digital Millennium Copyright Act (DMCA), 16
- digit character (\d), 83
- Directory directive, 290
- Directives, with Server Side Includes, 277
- Directories
 - for mod_perl programs, 233
 - searching with WML, 138
- Directory indexes, disallowing in Apache, 44
- Directory listing sample, 23

Directory permissions, and server errors, 197
`disconnect()` method, 123
 Disk Druid, 17
 DIVERSION, 163
 Diversion filter phase (Pass 5), in WML, 137
 Diversions, 162–164, 182
 divert Name, 163
`dl()`, in CGI.pm, 200
 DNS, 55
 do ... while loop, in PHP, 395
 doctype, 48
`DOCUMENT_NAME` variable, and echo directive, 279
 Document Object Model (DOM), 6
 Document root, 34, 47
`DOCUMENT_URI` variable, and echo directive, 279
 Dollar sign (\$), 61
 and non-root user, 19
 variable names in PHP beginning with, 382
 Dotfile generator, 22
 Dotfiles, 22
 DOUBLE, in MySQL, 113
 Double parentheses, with wml::fmt::xtable, 158
 Double quotes, 64, 196
 in Perl, 178
 and PHP, 425, 383
 string literals created with, 63
 and WML, 138, 140
`[do]...[$ until condition $]` metacommand, in Embperl, 299
 Downloaded CGI programs, avoiding, 218–219
`-DROOT~`, 176
 DROP TABLE command, in MySQL, 112
 Dual booting, 15
`du` command, 26
 Dumping databases, 130
`dvipdf`, xxxii
 Dynamic clickable menus, 6
 Dynamic content, 4
 Dynamic data, serving up, 6
 Dynamic HTML, 6, 229
 Dynamic HTML web pages, mod_perl and creation of, 272

Dynamic web pages, 187
 and embedded HTML, 7
 mod_perl and creation of, 231
 PHP and creation of, 377
 Dynamic web sites, 133
 and structural portion of LAMP, 130

E

`echo` directive, 279–280
`echo` PHP statement, 380
`echo` statement, 406
`else` part, 217
`emacs`, xxxii, xxxiii, 379
`e-mail`, xxx, 271
 PHP and sending of, 377
 and web page traffic, 272
 Embedded content, 4
 Embedded HTML, serving up content with, 7–8
 Embedded programming, approaches to, 7
 Embedded programs, for building HTML, 307
 Embedded web sites, and structural portion of LAMP, 130
 Embperl, 7, 8, 122, 133, 174, 184, 286, 287–329, 378, 379
 and Apache configuration, 289–291
 commands in, 292–299
 “hello, world!” with, 291, 292
 installing, 288–289
 introduction to, 287
 Mason compared with, 332–333, 375
 and persistent database connections, 328
 PHP as alternative to, 431
 posted data and %fdat, 299–301
 project with, 304–327
 \$row and \$col, 301–303
 sample list of variables in, 328–329
 session handling with %udat and %mdat, 328
 XML and XSLT support with, 329
 Embperl commands, 292–299
`[$ command arguments $]`, 295–298
`[! Perl Code !]`, 294–295
`[+ Perl Code +]`, 292–294
`[– Perl Code –]`, 294
`[# some text #]`, 295
`[$ while $]`, 299
`EMBPERL_DEBUG`, 290

- EmbperlObject, 306
- Embperl project, 304–327
 - CGI used in, 306
 - consistent look and feel created in, 305–306
 - embedded programs for building HTML in, 307–310
 - hard-code HTML files, 306
 - HTML::EmbperlObject used in, 310–318
 - mod_perl used in, 306–307
 - problem defined for, 304–305
 - product filter in, 319–327
 - WML used to create HTML files, 306
- embperl_template.pl, contents of, 307
- Empty strings, 178
- Empty string values, 322, 323
- _END__ file, markers added to, 144
- end_form(), for widget creation, 216
- end_html() method, in CGI.pm program, 199
- _END__ marker, 145
- End-of-diversion marker, 163
- End Of HTML (EOHTML), 196
- End-of-line, 144
- Engelschall, Ralf, 133
- entries table, in journal MySQL database, 420
- Enumerated arrays
 - fetching row as, 412
 - in PHP, 384, 385, 396, 397
 - sorting in ASCII ascending/descending order, 407
- Environment variables, 24, 25
 - displaying in HTML table, 302
 - displaying with Embperl's while command, 298
 - UNIX, 24, 25
- \$_ENV variable, 390
- eperl, 136–137, 174–179
- .epl extension, 289
- Equal sign (=), in format of query string, 208
- Equal-tilde (= \sim), 81
- Equivalence operator (==), 73
- \$error, 371
- Error handler, rolling, 222
- Errors, 37
 - fixing, 38
 - server, 190, 192
- \$error variable, 369, 370
- Escape characters, in PHP, 383
- \$escmode, 320, 329
- \$escmode = 0; statement, 309
- /etc directory, 29
- /etc/init.d directory, 29
- /etc/X11 directory, 29
- etherape program, 30
- Ethernet, 3
- Examples
 - Embperl, 291
 - EmbperlObject, 312, 313, 314
 - Mason, 335–336
 - PHP, 379–380
- Exclamation point character (!), 94, 208
- .exe, xxviii
- +ExecCGI option, and Apache::Registry, 236
- exec directive, 283–284
- exec.html, 283
- @execute_args, 322
- execute() method, 125, 126
- expects_array() function, 404
- Expression modifiers, in Perl, 100
- Extensible Markup Language, 8–9
 - Embperl integrated with, 329
 - PHP combined with, 378
- F**
- False values, in Perl, 73
- FAT filesystem, 16
- %fdat, 323, 325
 - posted data and, 299–301
 - using, 301
- fetchrow() method, 125
- @ffld variable, in Embperl, 328
- FH variable, 96
- Fields, in MySQL, 111
- _FILE__, 145
- File contents, and SSI, 277
- filefield(), for form widget creation, 218
- Filehandles, 96, 98
- File information, with WML, 143–144
- File I/O, with Perl, 93–98
- Filenames
 - distrust of, 221
 - and SSI, 277
- _FILE__, 136, 143

- File permissions
 - changing, 23
 - and server errors, 197
- Files *.html directive, 290
- Files, and Perl, 95–98
- Files directive, 43, 134, 356
- file.shtml, 282
- File slurp, 97
- Filesystem Hierarchy Standard, 13
- Filters, with Mason, 331
- finish() method, 125
- fips, 16
- Firewalled router, 20
- Firewalls, 20, 55
- Firewire, 3
- flastmod directive, 282, 283
- Flat-file solutions, for small databases, 254–255
- FLOAT, in MySQL, 113
- Floating points, in MySQL, 113
- Floats, 63, 383
- Flow-control constructs,
 - in Perl, 77–81
 - in PHP, 377, 391–398
- _footer component, 375
- Footer file, in Mason, 348, 349
- Footers, 165, 166
 - for base.html, 316
 - in “hello, world!”, 138, 139
 - and navigation bar definition, 153
 - in WML template file, 375
- footer2 component, in Mason, 352
- foreach loop, 80–81, 248
 - in Mason, 341
 - in PHP, 395–398
- Foreach metacommand, in Embperl, 295
- for loop, 79
 - in Mason, 340
 - in PHP, 395
- Form data
 - processing with PHP, 378
 - subroutines for processing, 224–225
- Forms interfaces, 187
- Forms, PHP data processing submitted via, 386–387
- <form> tag, and PHP project
 - add page, 423
- Form widgets, 210, 216–218
- freetable, warning about tables generated by, 157, 160, 161
- FrontPage, 378, 379
- fszie directive, 281, 283, 284
- Function arguments, in Perl, 90–93
- Function calls, and Embperl commands, 292
- Function names, in PHP, 381
- Functions
 - array, 66–67
 - hash, 70–71
 - string, 107
 - user-defined, 87
- G**
- Galeon, 3
- Game-boxes, 11
- GET, 247
- get() function, 238
- GET method, 208, 388, 389
- get_remote_host(), 270
- \$_GET variable, 390
- Ghostview, xxxii
- Gimp, xxix
- gkrellm program, 30
- Global variables, 89
 - caution in use of, 244
 - in ill-considered mod_perl
 - CGI programs, 242
 - in Mason, 336, 337, 373
 - and mod_perl programs using Apache::Registry, 237
 - in PH version 4.1.0, 390
- Gnome, 18, 20
- Google, 25, 38
- GPL license listing, xxx
- graceful option, 38
- GRANT command, 122, 256
- Graphical user interfaces (GUI), 18
- Graphic designers, 7
 - and Mason, 332, 333
- grep command, 26, 87
- Groups
 - setting in Apache, 41
 - and UNIX, 22
- GUI configuration programs, with Red Hat, 30–31

H

Hacking, cracking *versus*, xxx
HAKMEM, xxx
HandleCGIData.pm, 246
HandleCGIData2.pm, 250
handle_error(), 225, 226
handle_error subroutine, 224, 259
handler() function, 261, 266
handler() method, 244, 252, 269
handler() subroutine, 245, 247
Hard-coding HTML files, 306
Hash character (#), with Unix root accounts, 19
Hash elements, and Embperl commands, 292
Hashes, creating, 68
Hash functions, 70–71
Hash variables, 67–69, 175
.header component, in Mason, 375
Header file, in Mason, 348
header() method, 105, 199
Headers, 35, 50, 165, 146, 166
 with CGI program, 188, 189, 190, 192
 in “hello, world!”, 138, 144
 in navigation bar definition, 153
 referer information sent through, 221
 in tables, 156
<head> information, 223
<head> tag, in wml::std::page, 147
head.wml, 140
“Hello, world!”, 49, 59–60
 CGI program, 191–196, 192, 193, 194
 with Embperl, 291, 292
 with mod_perl, 237
 with PHP, 380
 with pure mod_perl, 243–244, 245
version with WML, 138, 139, 140
version 2 with WML, 142
version 3 with WML, 144–146, 147
hello.cgi, creating, 191
hello.php file, 379–380
hello2.cgi program, 192
HelloWorld package, 244
HelloWorld.pm module, 245
here document, 195, 196, 215, 216, 223,
 229, 245, 260
hidden(), for form widget creation, 218
Hidden fields, distrust of, 220

High-level macro construct expansion phase (Pass 2), in WML, 136
hitit.pl program, 237, 241
hitit2.pl program, 239, 240, 241
<hN> tags, with wml::std::toc, 149, 150
/home directory, 28
h1() method, 229
 in CGI.pm program, 199, 200
<h1> tag, 375
Horizontal rule, 309
\$host, 201
Hostname
 and CGI.pm program, 201
 display in CGI program, 194
hr() method, in CGI.pm program, 198, 199
htaccess directive, uncommented, 43, 51
.htaccess, 43, 50–55, 190
h3() method, in CGI.pm, 300
HTML
 for components2.html, 352
 embedded programs used for building, 307
 and eperl, 174, 175, 176, 178
 for navigation bar rendering, 154
 PHP embedded into, 378–379
 SSI and embedding executable code within, 286
 and text diversion filter, 162
 with wml::std::page, 148
 in wml::std::toc, 150
HTML code, 137, 138
HTML editors, 378, 379
HTML::Embperl, 287, 289, 299, 300, 377
HTML::EmbperlObject, 310–311
.html extension, 289, 379
HTML files
 for Apache, 34
 building for WML, 135–136
 and dynamic data, 6
 hard-coding, 306
 Server Side Includes embedded in, 277
 and static data, 5
 %args tag in, 346
HTML lists, methods for generation of, 204
HTML::Mason, 331, 333, 377
HTML pages, Mason and Embperl and, 332
HTML tables, cells formatted in, 158
<html> tag, in wml::std::page, 147
HTML tags, closing tags required by, 171

httpd file, location of, 288
 httpd.h file, location of, 288
 httpd PIDs, 37
`$HTTP_GET_VARS`, 388, 389
`%http_headers_out`, in Embperl, 329
`$HTTP_POST_VARS`, 388, 389
 HTTP requests, 3, 35
 HTTPS, 53
`h2()` method, in CGI.pm, 200
 HyperText Transfer Protocol (HTTP), 33

I

ICANN, 55
`$id`, 431
 IDENTIFIED BY clause,
 in SQL command, 121
 id field, in journal MySQL database, 420
 if condition, PHP syntaxes for, 391–392
 ifconfig, netstat command, 27
 if directive, 285
 if metacommand, in Embperl, 295, 296
 if statement, 57, 77, 100
 in Mason, 339
 in Mason project, 363
 in PHP, 394
 widgets.cgi modified by, 213
 image_button(), for form widget creation, 218
 Image rollovers, 6, 9, 151
 Images, 4
 images subdirectory, of HTML root
 directory, 169
`$(IMG)`, 169
 imgbase attribute, in navigation
 bar definition, 153
 imgstar, in navigation bar definition, 153
 Immunix, 13
`#include`, 143, 144
 include directive, 280–281, 282
 Include directories, 170
 include (-I), in WML, 138
 Increment operators, in Perl, 75–76
 index.html file, 47, 313, 314
 and Mason top-level components, 374
 simple, 51
 Infinite loops, preventing, 328
 Informix, 109
`<info>` tag, 143, 144, 146, 179
 info.wml file, 143

`<%init>` block, 365
 in Mason, 368
 in Mason project, 358
`<%init ... %init>` tag, in Mason, 353–354
`<%init>` section, and top-level
 components, 374
 Initialization tags, in Mason, 372
 Inner HTML, and templates, 163
 Input, reading with, 106
 INSERT command, in MySQL,
 114–115, 122
 INSERT INTO, 115
 INSERT MySQL query, 416
 INSERT query
 and DBI, 122
 SQL prepared for, 126
 INTEGER, in MySQL, 113
 Integers, 63
 in MySQL, 112
 in PHP, 383
 International characters, 137
 Internet, 9
 and Open Source programs, xxx
 Internet address, 3
 Internet protocol (IP), xxvii
 Internet service provider (ISP), xxvii
`int()` function, 72
`-I` option, 136, 138
 IP addresses, and Apache log files, 55
 IP port numbers, 3
 IP verification, and .htaccess file, 53
`@isbn`, ISBNs pushed onto, 262, 263
 ISBNs, 266, 267
 and Apache::Request object, 261
 retrieving in books database, 262
`$isbn` variable, 266

J

Java programming language, 9, 61, 382, 391
 Java applets, 4
 JavaScript, 4, 6, 9, 154, 155
 JavaScript preprocessing, distrust of,
 221–222
 JavaServer Pages (JSP), 9
`join()` function, in PHP, 410
 journal database, 425, 431
 Journal MySQL database, 420–421
 jpg extension, 291

K

kbdconfig program, 31
KDE, 18, 20
Key, book_information created with, 256
Keys() function, 70, 409
killall command, 24, 38
killall foo, 24
kill command, 24, 38
kill -9 command, 38
Konquerer, 3
krsort() function, in PHP, 407, 408
ksort() function, in PHP, 407, 408

L

LAMP, xxviii, xxxiii
PHP within, 377
programs, xxxii
structural portion of, 130
Language notes, Embperl, 299
Language syntax, for PHP, 381–404
LAST_MODIFIED variable, and echo directive, 279
LaTeX, xxxii, xxxiii
Left rail, 223, 260, 262
for www.opensourcewebbook.com, 181
.leftrail component, in Mason project, 358, 361, 363
.leftrail component, in Mason, 371
Legacy CGI scripts, mod_perl and speeding up of, 273
Less-than symbol (<), 96
LGPL license listing, xxx
libapreq package, 249
/lib directory, 28
Licenses, Open Source, xxix
li(), 200, 204
__LINE__, 136, 143, 145
% Lines, in Mason, 337
Link identifier, in MySQL, 411
Links, 3
in Embperl, 293
to eperl web sites, 176
in PHP project launch page, 422
rolling over, 151
<link> tag, 172, 179

Linux, xxviii, xxix, xxxi, xxxii, 10, 11–31, 109, 130, 278

accounts, 18–20
and Apache configuration, 190
decisions about installation of, 15–16
distributions, 11–13
download and install, 14–15
installation CD, 18
partition sizes, 16–18
and PHP, 379
recommended partition sizes, 17
root user, 110
security and, 20
useful programs, 30–31

linuxconf program, 30
linuxconfig System, 31

Linux Documentation Project, 38
Linux Filesystem Standard, 13
Linux Security HOWTO, 9
Lists, in CGI.pm, 200
LiveScript, 4n
ln command, 26
Loading databases, 129
Local documentation, access to in Apache, 42
localtime() function, 201
Local variables, in Perl, 89–90
locate command, 26
Locations, and diversion filter, 137

Locked accounts, 41
LOG, in Embperl, 329
Log file, in MySQL, 110
Logging
and Embperl, 287
mod_perl and enhancement of, 272
and mod_perl programming, 243

Logical and operator, 76
Logical and operator (&&), 76
short-circuited nature of, 101, 102
Logical not operator (!), 76
Logical operators, in Perl, 76–77
Logical or operator (||), 76, 209
short-circuited nature of, 101, 102
Logic operators, short-circuited, 101–103
Log monitoring program, 53
logrotate, 50
Log-watching program, 20
LONGBLOB, in MySQL, 114
LONGTEXT, in MySQL, 114

- Looping constructs, 77–79
 embedding in Mason, 337
 in Embperl, 296
- Looping statements, in Mason, 342
- Low-level macro construct expansion phase (Pass 4), in WML, 137
- ls command, 26
- LWP::Simple module, 238
- Lynx, 3
- M**
- Macros, creating in WML, 170–174
- Mail header, 271
- make capacity, of WML, 184
- make-like command, 133
- man command, 20, 25
- Mandrake, 12
- _man_form, in Mason project, 363–364, 365, 368
- man function, 24
- man_id field, 304
- man-k best guess at a useful function, 24
- man -k command name, 20
- man -k foo command, 25
- man man command, 20, 25
- man mysqldump, 130
- man pages, 57
- man strftime, 285
- Manufacturer IDs
 in Mason, 370
 in Mason project, 361, 364, 366, 367
- Manufacturers, modifying in Mason project, 371
- Manufacturers table in Embperl project, fields in, 304
- Markup code fixup phase (Pass 7), in WML, 137
- markup code splitting and output generation phase (Pass 9), in WML, 138
- Markup code stripping phase (Pass 8), in WML, 137
- Mason, 7, 8, 122, 133, 184, 286, 331–375, 377, 378, 379
 Apache configuration with, 334–335
 component class API, 373
 components, 350
 components in, 335, 348–356
 conditional statements, 340
- Embperl compared with, 332–333, 375
 examples in, 335–336, 345, 347
 and inline Perl sections, 336–343
 installation of, 333–334
 introduction to, 331
 % lines in, 338
 looping statements in, 342
 PHP as alternative to, 431
 posted data handled with %ARGS and %args in, 343–348
 project, 357–371, 372
 request API with, 373
 security with, 354–356
 serving up components in, 355
 session handling with, 373
 tags for initialization and cleanup in, 372–373
 top-level components in, 374–375
- Mason Request object, API for, 373
- matches_class() function, 91
- \$maxcol, in Embperl, 328
- \$maxrow, in Embperl, 328
- maxsize, 225
- %mdat, session handling with, 328
- MEDIUMBLOB, in MySQL, 114
- MEDIUMINT, in MySQL, 113
- MEDIUMTEXT, in MySQL, 114
- Memory, and Perl, 85–87
- Metacommands, in Embperl, 295
- Metainformation, 223
- Metalanguage, 133
- method=“POST”, 208
- Methods, 104
- Microsoft Windows, and dual booting with Linux, 15
- MIME. *See* Multipurpose Internet Mail Extensions
- mkdir command, 26
- MLDBM. *See* Multi-Level DataBase Manager
- /mnt directory, 29
- mod_env.so file, location of, 288
- Modify button, in Mason, 371
- mod_perl, 6, 122, 133, 187, 188, 229, 231–273, 265, 269, 329
- CGI data handled with, 248
- CGI data handled with using Apache::Request, 251
- CGIs turned into programs in, 235–243

- configuration of, 233–235
embperl hooked into, 287
introduction to, 231–233
MySQL, DBI and mod_perl project, 253–272
other capabilities with, 272
path information with, 253
pure programming in, 243–253
speed examples in, 237–242
for web page creation, 306–307
mod_perl programs, turning CGIs into, 235–243
\$MODPERLURL, 238
Modular pages, with Mason, 331
Modules
 installing, 105, 106
 strict used in, 244
mod_user module, 42
Month, 374
more command, 27
Mouse clicks, and navigation bar
 definition attributes, 153
mouseconfig, 31
Mozilla, xxix, 33*n*, 50
[+ \$msg +] Embperl command, 291
\$msg, 224, 259, 291, 294
mSQL, 109
Multi-Level DataBase Manager,
 installing, 333
multiple attribute set, 212
Multiple disks, dual boot installation
 option, 16
Multipurpose Internet Mail
 Extensions, 193
munge_phone() function, 91
mv command, 26
Mv -i option, 26
my() function, 89
my() operator, 69
MySQL, xxviii, 109–130, 265, 269, 377
 administrative details with, 119–121
 CREATE DATABASE command
 in, 111
 CREATE TABLE COMMAND
 in, 112
 and Database Independent interface, 122–127
 DELETE command in, 118–119
DESCRIBE command in, 114
implementing book program in, 255
INSERT command in, 114–115
insert commands in, 305
loading and dumping a database in, 129–130
log file, 109
and PHP, 411–420
root user, 110
SELECT command in, 115–117
SHOW DATABASES command in, 111
SHOW TABLE command in, 112
table in, 227
table joins in, 127–129
tutorial, 109–122
UPDATE command in, 117–118
USE command in, 111
mysqladmin command, 120
mysql_affected_rows() function, 416
mysql_change_user() function, 419
mysql_close() function, 412, 413
MySQL commands, 110
mysql_connect() function, 411, 413
mysql_create_db() function, 419
mysql database, 119, 120
MySQL database, persistent connections
 to, 328
mysql_data_seek() function, 419
mysql_db_query() function, 412, 413
mysql_drop_db() function, 419
mysql_error() function, 412, 413
mysql_fetch_array() function, 416
mysql_fetch_field() function, 419
mysql_fetch_row() function, 412, 413
mysql_field_name() function, 419
mysql_field_table() function, 419
mysql_field_type() function, 419
mysql_free_result() function, 417
MySQL functions, Part 2, 416–418
mysql_list_dbs() function, 419
mysql_list_fields() function, 420
mysql_list_tables() function, 420
mysql_num_rows() function, 412, 413
mysql1.php, 413
mysql_query() function, 416
mysql_select_db() function, 416
MySQL server, 110, 258
 closing connection to, 412

- MySQL server (*Continued*)

connecting to, 411

disconnecting from, 126
- MySQL user, Linux user differentiated

from, 120
- my variables, arguments copied into, 91
- N**
- \$name, 326

name, in navigation bar definition, 153

nameage.cgi, contents of, 208–209

Named anchors, 149, 150

Named parameters, *versus* ordered

arguments, 202

name field, 304

Namespace, 244

Name/value pairs, in Mason

components, 351

National Center for Supercomputing

Applications (NCSA), 33

navbar definition, 152–153

<navbar:epilog> tag, 153

navbar:jsfuncs, 154

<navbar:prolog> tag, 153

<navbar:render> tag, 154

navbar.wml, 154

Navigation bars, 151

designing, 152

rendering in WML, 154

neat program, 30

nessus program, 30

Nested if statements, 77

Netscape, 33n

Netscape/Mozilla, 3

Network connection, 3

Newline character (“\n”), 60, 192

removing, 94, 95, 104

and standard input in Perl, 93

new method, in Perl, 105

new() method, of Apache::Request

class, 261

newtitle.html, 318

using title redefined in, 319
 *nix systems, xxxii, xxxiii
 no_default_args function, 402, 404
 Nondestructive functions, 67
 Nondigit character (\D), 83
 Nonspace character (\S), 83
 Nonword character (\W), 83
- <nostrip> tags, 138

now() function, 421

NTFS filesystem, 16

ntop program, 30
 \$NUM_BOOKS variable, 261
 NUMERIC, in MySQL, 113
 Numerical comparison, in Perl, 73–74
 Numeric literals, 63
- O**
- Object-oriented programming (OOP),

introduction to, 104–106

Object-oriented techniques with Mason, 331
 -o example.html option, 136
 offline mode prompt, 199
 -oi, sendmail invoked with, 271
 OmniWeb, 3
 <%once> block, in Mason project, 358
 <%once ... %once> tag, in Mason, 354
 Online manuals, removing

in Apache, 41–42
 open() function, 95, 96, 98, 258
 Open Office, xxix
 Open Source, 11, 377

model, drawbacks with, xxx–xxxii

software, description and philosophy of,

xxviii–xxx

web development, 9
 Open Source Standard Query Language, 109
 Open systems, xxix
 Opera, 3, 50
 Operators

Perl, 71–77

PHP, 390–391
 /opt directory, 28
 Options directive, disallowing Indexes

in, 44
 Oracle, 109
 Ordered arguments, *versus* named

parameters, 202
 or die() code, in DBI, 125
 or operator, 96, 391
 oswb.wml, 170, 179
 OS X, xxxii
 OUT, in Embperl, 329
 Outer HTML, and templates, 163
 output.txt, 98
 Owner, 22
 Ownership, of file, 23

P

- p(), in CGI.pm, 200
- Packages, in HelloWorld, 244
- PAGE_BODY, 182
- <page> tag
 - for WML file, 148
 - with wml::std::toc, 150
- \$(page) variable, 176, 178, 180
- Paragraph breaks, in CGI.pm, 200
- Parallel printer port, 3
- Parameters
 - in Mason, 331
 - passing to WML files, 140
- param() function, 213, 299
- param() method, 209
- %params hash, posted data in, 247, 248
- Params::Validate, installing, 333
- Parentheses
 - with wml::fmt::xtable, 158
 - in WML template files, 140
- Partition Magic, 16
- Partition sizes, Linux, 16, 17, 18
- password_field(), for form widget
 - creation, 218
- Password files, creating, 51–52
- Passwords, 9
 - choosing, 120
 - combining with IP verification, 55
 - MySQL root, 119
 - sniffing, 53
- Patches, xxxi
- PATH, 24
- path_info() method, 203, 251
- PathInfo.pm, 251
- Path information
 - and mod_perl programming, 243
 - obtaining in mod_perl, 251–253
 - received by CGI program, 202, 205–206
- Pathnames, with system() and
 - backquotes, 104
- PATH variable, 104
- Patterns, rules for in Perl, 82
- PDFs, PHP and creation of, 377
- people database
 - DBI module connected to, 123
 - in MySQL, 110
 - mysqldump command with, 130
 - and privileges, 120, 121
 - and table joins, 128–129
- Percent character, in format of query string, 208
- <%perl ... %perl> tag, in Mason, 342–343
- Perl, xxix, xxviii, 10, 57–108, 130, 188, 377, 382, 391
 - arithmetic operators in, 71–72
 - benefits with, 187
 - and CGI.pm, 197
 - and CGI use, 191
 - documentation, 58
 - Embperl based on, 329
 - expression modifiers in, 100
 - file I/O with, 93–98
 - flow-control constructs in, 77–81
 - increment and decrement operators in, 75–76
 - local variables in, 89–90
 - logical operators in, 76–77
 - and memory, 85–87
 - mod_Perl based on, 231
 - numerical comparison in, 73–74
 - operating system calls made in, 103–104
 - quantifiers in, 83–85
 - reading from file in, 95–98
 - reformatting/transferring data with, 256
 - regular expressions in, 81–87
 - return values in, 88–89
 - scripts in, 191
 - short-circuited logic operators in, 101–103
 - string comparison in, 74–75
 - string operators in, 72–73
 - usefulness of, 107
 - WML and power of, 184
 - writing to files in, 98
- perlcc, 60
- Perl code, embedding into Mason pages, 336–338
 - [! Perl Code !] Embperl command, 294–295
 - [+ Perl Code +] Embperl command, 292–294
 - [- Perl Code -] Embperl command, 294
- Perl.com, 59
- Perl constructs, embedding in Mason
 - HTML files, 338–341
- Perl DBI, 109, 122–127
- perldoc, 59, 328
- perldoc CPAN, 106
- PerlLogHandler, 272

- Perl pragma, 105
- perl process, 232
- Perl scripts, 93, 106
- PerlSendHeader, 236
- Perl Server Side Includes, 272
- Perl syntax rules, 59–104
 - backquotes, 104
 - control flow constructs, 77–81
 - declaring variables with use strict, 61–63
 - expression modifiers, 100–101
 - file I/O, 93–98
 - functions, 87–93
 - hello, world program, 59–60
 - and making operating system calls, 103–104
 - for Mason, 331
 - operators, 71–77
 - regular expressions, 81–87
 - short-circuited logic operators, 101–103
 - unless statement, 99
 - until loop, 99–100
 - variables, 63–71
- Permissions, 48, 218
 - for peoples database, 121
 - and server errors, 197
 - UNIX, 23
- Persistent database connections, with Embperl, 328
- Personal computers, economics and, xxvii
- Personal digital assistants, 11
- Personal digital video recorders, 11
- PHP, xxviii, 7, 8, 57, 122, 286, 377–432
 - arrays in, 384–386
 - built-in functions, 405–411
 - configuration of, 379
 - data types in, 383–384
 - embedding into HTML, 378–379
 - examples, 379–380
 - flow-control constructs in, 391–398
 - form 1, 387, 389
 - functions in, 399–404, 405
 - if example, 393
 - information with phpinfo(), 381
 - introduction to, 377–378
 - language syntax, 381–404
 - loops example, 398
 - and MySQL, 411–420
 - mysql examples, 414, 415, 417–418, 419
 - and MySQL functions, 411–418
 - operators in, 390
 - project, 420–431
 - variables in, 382
 - Web variables in, 386–387
 - PHP: Hypertext Preprocessor, 377
 - .php extension, PHP files named with, 379
 - PHP functions
 - arguments passed to, 401–404
 - example, 403–404
 - writing, 399
 - phpinfo() function, 380, 381
 - PHP project, 420–431
 - add page, 423
 - journal MySQL database, 420–421
 - launch page, 422
 - pages in, 422–431
 - view entry page, 430
 - view page, 426
 - PHP Version 4.1.0, 388–390
 - PID. *See* Process ID
 - Pipe, 271
 - Pipe character (|), 27
 - Placeholders, 126
 - Plain text, 4, 192
 - Plus sign (+), in format of query string, 208
 - .pm files, readability of, 245
 - popd command, 26
 - pop() function, 66
 - popup_menu, for widget creation, 216
 - Pop-up windows, 6
 - Portability
 - design for, 123
 - with Perl, 187
 - Ports, 3
 - POST, 247
 - \$posted_city, 321
 - Posted data
 - and %fdat, 299–301
 - handling with %ARGS and %args, 343–348
 - handling with mod_perl, 241
 - and PHP, 378, 388
 - Posted form data
 - handling with mod_perl, 246–249
 - and mod_perl programming, 243
 - name and age results, 210
 - processing, 206–216
 - received by CGI program, 202, 206–216

- PostgreSQL, 109
Postincrement, 75, 76
\$_POST variable, 390
Pound sign (#), comments begun with, 144
Pragma, Perl, 105
Precedence
 differences with logic operators, 103
 of PHP operators, 390
Preincrement, 75
prepare() method, in DBI, 125
preserve href, 172
preserve text, 172
Preshipped CGIs, avoiding, 218–219
Previewing, with Mason, 331
price field, in products table, 304
Primitive data types, in PHP, 384
printenv command, 24
print() function, 60, 93, 98, 188, 191, 192, 199
 content type sent with, 195
 in PHP, 405, 406, 410
print() method, HTML content generated
 with, 244–245
printtool program, 31
Prioxy program, 30
Privileges, 120, 121
/proc directory, 29
Processes, 23–24
process_form_data(), 226
process_form_data subroutine, 225
Process ID, 23
Processing instruction, in XML, 378
_prod_form_, in Mason, 368
prod_id field, in products table, 304
\$prodid, 366
Product detail, in Embperl project,
 324–325, 327
Product filter, in Embperl project, 319–327
Product filtering system project, in Embperl,
 304–327
Product ID, in Embperl project, 325–326, 327
Product page, in Mason project, 368
Products in Embperl project, fields in, 304
Program execution, and SSI, 277
Programmers
 hacking done by, xxx
 and Mason, 332, 333
Programming construct expansion (Pass 3),
 in WML, 136–137
- Programs
 CGI, 191–196
 with DBI, 122
 with two functions, 91–93
 WML, 134–135
Projects
 CGI/MySQL/DBI, 223–229
 Embperl, 304–327
 MySQL, DBI and mod_perl, 253–272
 template for www.opensourcewebbook.com,
 179–183
<protect> tag, 138, 306
Proxy servers, warning about with Apache,
 44–45
ps command, 24, 37
ps2pdf, xxxii
-p switch, and MySQL root password,
 119, 120
Public_html web sites, disallowing
 in Apache, 42
Pure mod_perl programming, 243–253
 and Apache::Request, 249–251
 getting path information with, 251–253
 note about mod_perl and, 246
 posted form data handled with mod_perl,
 246–249
Pure mod_perl programs, 231
pushd command, 26
push() function, 66
pwd command, 26
Python, 57
- Q**
Quantifiers, in Perl, 83–85
Queries, executing, 412
Query string, 206, 208
query_string(), 203, 208, 270
QUERY_STRING_UNESCAPED variable,
 and echo directive, 279
Question mark (?), 205
 as placeholder, 126
 and posted data, 207
- R**
\$r, 244, 259
Radio button group, in posted form
 data, 211
Radio buttons, 360

radio_group(), for widget creation, 216
rc file, with WML, 167
Readability
 and background color, 264, 323
 whitespace characters for, 126
Read mode, 96
REAL, in MySQL, 113
Reboots, xxxii
Records
 in `age_information` table, 124
 and `auto_increment` integer, 420
 and `DELETE` command, 118–119
 entering of into tables, 125
 in MySQL, 111
 and `SELECT` command, 115
Red Hat, xxxiii, 12, 13, 14, 17, 288, 333,
 377, 388
 installer, 16
 Linux programs with, 30
 and security, 20
Red Hat Package Manager, 12
Red Hat up2date mechanism, 12
red tag, defining for www.opensourcewebbook.com, 179
References, 107
Referer
 and Apache log, 40
 and session information received by CGI program, 203
 Referer information, distrust of, 221
Regular expressions, in Perl, 81–87, 86
Relational databases, 127
remote_host(), and session information received by CGI program, 202
REPLACE command, 371
\$report_contents, in Mason, 374
\$req, request object assigned to, 320
\$_REQUEST, 390
Request API, in Mason, 373
Requested file, 311
Reset button, in posted form
 data, 212
restore href, 172
restore text, 172
\$result, 367
Result identifier, 412
return operator, 89
 return statement, in PHP, 399
 Return values, in Perl, 88–89
reverse() function, 67
REXX, 57
 Richter, Gerald, 287
Right rail, WML with, 181–183
\$(rightrail), 181
RIGHTRAIL section, in `www.opensourcewebbookpage.com`, 183
rm command, 26
\$(ROOT), 169, 176, 181
root access, 23
root account, 18–19
/root directory, 28
Rooting your box, 41
root ownership, 23
Root user, 110
\$(ROOT) variable, link `hrefs` based on, 179
\$row, and dynamic table generation, 301, 303
Rows, 111, 159
rpmfind.net, 30
rset(), for widget creation, 216
rsort() function, in PHP, 407
RTFM Web site, xxxin

S

SAINT, 30
Samba, 18
Sanity checks, 321, 369, 425
SARA, 30
/sbin directory, 28
Scalar variables, 61, 63–64, 93
Scams, and hidden fields, 220
Scientific float, 63
ScriptAlias directive, 190
script_name(), and CGI program, 203
Scripts, 122
 CGI, 187
 Perl, 93, 106, 122, 191
 security warning about running of, 107
 unexpected fields passed to, 219
<SCRIPT> tag, 378
scrolling_list(), for widget creation, 216
Scrolling list box, 248–249

- Scrolling list box widget, 212
\$search, 427
Search engines, 187
Secure shell, xxxiii, 3, 20
Security, 9, 225, 307. *See also* Apache security
and books listing database, 270
CGI, 218–222
Embperl project, 325
and Linux, 20
and MySQL root password, 119
with PHP, 387–388
with Server Side Includes, 286
Security bugs, xxxi
sed, 28, 57, 87
Select buttons, 360
SELECT command
in MySQL, 115–117, 122
and table joins, 128
.select_form component, in Mason, 359,
361, 371
SELECT * FROM age_information
SQL query, 124
SELECT query, 122, 227, 371
<select> tag, 211
Semicolon (;), PHP statements terminated
by, 382
send_http_header(), 244
sendmail, xxx, 271
Server errors, 190, 192, 196
server-info, removing in Apache, 43
Server misconfigurations, 51
server_name() method, 201, 203
server_port, and CGI program, 203
Servers, 4
information received by CGI program
from, 202–216
status of in MySQL, 110
Server Side Includes, 7, 277–286, 278
config directive, 284–285
configuration, 278–279
echo directive, 279–280
exec directive, 283–284
flastmod directive, 282–283
fsize directive, 281
if directive, 285
include directive, 280–281
introduction to, 277–278
security considerations with, 286
set directive, 285
tutorial, 279
Server-side processing, 133
Server-side programming, 6
server_software(), and CGI program, 203
server-status, removing in Apache, 43
\$_SERVER variable, 390
Session handling
in Mason, 373
with %udat and %mdat, 328
Session information, received by
CGI program, 202–204

- SORT_NUMERIC, 407
- SORT_REGULAR, 407
- SORT_STRING, 407
- Source reading and include file expansion
 - WML phase (Pass 1), 136
 - \space character, 83
- Special characters, 137
- Special variables, 106
- Speed, with mod_perl, 231, 232–233, 241
- sprintf() function, in PHP, 410
- \$sql_cond string, 322
- SQL databases, and DBI, 122
- SQL query, for inserting data into
 - manufacturer's table, 370
- Square brackets ([]), in character class, 82
- SSH. *See* Secure shell
- SSI. *See* Server Side Includes
- Staging/production modes, with Mason, 331
- Standard float, 63
- Standard input, and Perl, 93
- Standard mode, CGI.pm module used in, 212
- Standard output, and Perl, 93
- Standard Query Language (SQL), 109, 126
- Standard style, of programming, 197
- Star character (*), and navigation
 - bar definition, 153
- start_form(), for widget creation, 216
- start_html() method, 199, 229
- start_form() method, for form widget creation, 218
- startup.pl file, 234
- state field, in manufacturers table, 304
- Statement handle object, 125
- Static content, 4, 5
- Static data, serving up, 5
- Static web pages, 184, 187
- Static web sites
 - and structural portion of LAMP, 130
 - and Website META language, 133
- STDIN slurp, 94
- \$sth value, 224, 259
- strict, use of in modules, 244
- String comparison, in Perl, 74–75
- String functions, 107
 - in PHP, 410–411
- String literals, 63–64
- Strings
 - concatenation of, 72
 - in Perl, 72–73
 - in PHP, 383
 - replication of, 72
- Subdirectories, in EmbperlObject, 313, 314
- submit(), for widget creation, 216
- Submit button, in posted form data, 212
- Subroutines
 - defining with Embperl, 295
 - in Perl, 87
 - for processing form data, 224–225
 - substr() function, 410
 - summary attribute, 158
- Superuser, 18
- SuSE, 12
- swatch, 30, 50
- switch statement, in PHP, 392–394
- Sybase, 109
- Symbolic links, disallowing in Apache, 44
- Syntax
 - for calling and declaring PHP function, 399
 - for diversion, 162, 163
 - for echo directive, 279
 - in eperl, 175
 - for for loop, 340
 - for foreach loop, 341
 - for foreach metacommand in Embperl, 297
 - for fsize directive, 281
 - for header and footer HTML, 165
 - for here document, 195
 - for if command in Embperl, 295
 - for if directive, 285
 - for if statement, 339
 - for indexing PHP array, 384
 - in PHP, 377, 378
 - for PHP flow-control constructs, 391
 - for PHP foreach, 395–396
 - for PHP for loop, 395
 - for PHP switch statement, 392–394
 - for PHP while loop, 394–395
 - for set directive, 285
 - for unless statement, 339
 - for while loop, 340
 - for while metacommand in Embperl, 296
- Syntax errors, in Mason, 336
- system() function, 103–104
- System information, and mod_perl programming, 243

T

-t, sendmail invoked with, 271
Table cells, defining, 158
Table joins, 127–129
Table of contents, indentation of with
 wml::std::toc, 149
Tables
 and DELETE command, 118–119
 and DESCRIBE COMMAND, 114
 dynamic generation of with Embperl,
 301–303
 in Embperl project, 304–305
 Embperl variables for, 328
 and INSERT command, 114–115
 in MySQL, 111
 and SELECT command, 116
 with three rows and two columns, 159
 <table> tag, 156, 158
\$stabmode variable, in Embperl, 328
Tag attributes, in WML, 172
Tags
 <% ... %>, in Perl, 337
 <? ... ?>, in PHP, 378, 380
 <- ... ->, in Embperl, 321
 <! ... !>, in Embperl, 319
 <& ... &>, in Mason, 349
 defining, 174, 179
 in HTML, 170
TARBALS, downloading, 105
TCP/IP, xxx, 36
tcsh, 22
Telnet, 3
Template creation in WML, 138–146
 comments, 144–146
 file information, 143–144
 standard includes, 143
 use statement, 143
 varying template files, 140–142
Templates, 318
 and bread crumb trails, 176
 with Mason, 331
 for subdirectories, 314
 in WML, 184
test.cgi program, 189–190
test1() and test2() functions, 88
test.txt, 96
TEXT, 114, 225
textarea(), for widget creation, 216

Text areas, in posted form data, 211

Text diversion filter, with WML, 162–164

text field, in journal MySQL database, 420

textfield(), for widget creation, 216

Text-formatting commands, in CGI.pm, 200

Text processing, Perl used for, 187

Ticks, 64

Time

 and CGI.pm program, 201

 display in CGI program, 194

timefmt directive, 285

time() function, 238

TIMESTAMP, in MySQL, 113

timethese() method, 239, 240

TINYBLOB, in MySQL, 114

TINYINT, in MySQL, 113

TINYTEXT, in MySQL, 114

title field, in journal MySQL database, 420

title() function, 318

Titles

 default in embperl project, 318

 dynamic changes for in EmbperlObject, 316

 in Mason project, 358

<title> tag, 317

 defining for www.opensourcewebbook.

 com, 180

 in WML template files, 140, 142

/tmp directory, 28

<toc> tag, with wml::std::toc, 149, 150

-top, in Mason, 357, 371

top command, 24

.top component, in Mason project, 359,
 361, 363

top() function, 308, 309

top_html() function, 223, 260, 262, 266

Top-level components, in Mason, 374–375

Tried-and-true CGI, 6

trim() function, 411

Tripwire, 20, 30

true value, 73, 308

Tutorial, MySQL, 109–122

.txt extension, 291

U

%udat, session handling with, 328

UI()

 in CGI.pm, 200

 for generating HTML lists, 204

- uname command, 27
 - Unclosed tags, ways for using, 171–172
 - Underscore character (_)
 - in files in Mason directory, 355
 - in integers, 63
 - in Mason component files, 349
 - in variable names in PHP, 382
 - University of Illinois Urbana-Champaign, 33
 - Unix, 11
 - basic, 20–29
 - commands, 25–28
 - filesystem essentials, 28–29
 - owner, groups, permissions, ownership, 22–23
 - PATH and Environment, 24
 - processes, 23–24
 - shell, 22
 - Unix variants, xxxii
 - “Unknown unknowns (unk-unk),” xxixn
 - unless statement, 99, 102, 103, 339
 - unshift() function, 66
 - until loop, 99, 101
 - UPDATE command, in MySQL, 117–118, 122
 - UPDATE MySQL query, 416
 - Uppercase letters, filehandles named with, 96
 - UPT, 25
 - uri(), 270
 - URI requests, and Apache log, 40
 - urlbase, in navigation bar definition, 153
 - URL encoded data, 208
 - URLs, 3, 149
 - \$url variable, 293
 - USAH, 25
 - USB, 3
 - usbview, 31
 - Use, 144
 - USE command, in MySQL, 111, 121
 - #use directive, 143
 - use lib statement, 234
 - use pragma, 105, 197, 198, 234, 259
 - User, setting in Apache, 41
 - useradd, 41
 - User agent, and Apache log, 40
 - User-agent(), and session information
 - received by CGI program, 203
 - User-defined functions, 87
 - user_name(), and session information received by CGI program, 202
 - use statements, 143, 178
 - use strict;, 61–63, 69
 - /usr/bin directory, 28
 - /usr/lib directory, 28
 - /usr/local directory, 28
 - USR1 signal, 38
 - /usr/sbin directory, 28
 - /usr/src directory, 28
 - /usr/X11R6 directory, 28
- V**
- Values, 145
 - PHP functions and return of, 399–401
 - values() function, in PHP, 409
 - VALUES qualifier, in MySQL, 115
 - \$value variable, and PHP foreach, 396
 - VARCHAR, in MySQL, 114
 - /var directory, 29
 - \$(VARIABLE), 168
 - Variable names, in PHP, 381, 382
 - Variables, 60, 63–66
 - array, 64–66
 - declaring with use strict, 61–63
 - defining in .wmlrc, 168–169
 - hash, 67–69
 - initializing with Embperl, 295
 - locally scoped in PHP, 401
 - in PHP, 387
 - scalar, 61, 63–64
 - special, 106
 - SSI and display/creation of, 277
 - in WML, 184
 - /var/log directory, 29
 - /var/spool directory, 29
 - /var/tmp directory, 29
 - /var/www directory, 29
 - vi, xxxii, xxxiii, 379
 - virtual_host, and CGI program, 203
- W**
- Web. *See* World Wide Web
 - Web design category includes,
 - in wml::des::all, 161
 - Web designers, 7, 332
 - webmin program, 30
 - Web pages
 - Apache webserver and user requests for, 34–35

- brackets put on, 292
 - dynamic, 7, 187, 231, 377
 - SSI and embedding executable code
 - into, 277
 - static, 184, 187
 - and traffic, note about e-mails and, 272
 - Webserver market, Apache's place in, 33
 - Webservers, 18
 - Website META language, 5, 133–184
 - basics about, 135–138
 - better template with, 165–167
 - and eperl, 174–179
 - helpful include files with, 147–162
 - for HTML file creation, 306
 - installation of, 135
 - introduction to, 133–135
 - macros and custom tag creation
 - with, 170–174
 - phases, 136–138
 - programs, 134–135
 - project with, 179–183
 - template creation with, 138–147
 - text diversion filter with, 162–164
 - .wmlrc configured with, 167–170
 - Web sites
 - consistent look and feel for, 305–306
 - and Mason components, 348
 - and structural portion of LAMP, 130
 - Web variables, in PHP, 386–387
 - WHERE clause, and UPDATE
 - command, 117
 - where command, 27
 - which command, 27
 - [\$ while...\$] command, in Embperl, 299
 - while loop, 57, 77–79, 99, 101
 - in Mason, 340
 - in PHP, 394–395
 - and SELECT query, 125
 - While metacommand, in Embperl, 295, 298
 - Whitespace, 83
 - in diversions, 164
 - and Perl, 60
 - pound sign after, 144
 - Whitespace characters, 126, 137, 171
 - WML and removal of, 140, 142, 172
 - Whitespace-insensitivity, with PHP, 381
 - whitespace=tag, 171
 - who command, 27
- Widget example result, 214
 - widgets.cgi, 211, 213
 - widgets.html, 213, 215
 - widgets2.cgi
 - else part of, 215
 - if portion of, 214, 217
 - WiFi, xxvii
 - Wildcards, 26, 159
 - wmb: Website META Language Bug Reporting Tool, 135
 - wmd: Website META Language Documentation Browser, 134
 - wmk: Website META Language Make program, 134, 135
 - wmk command, 139
 - WML. *See* Website META Language
 - wml:des::all, 161–162
 - wml::des::gfont, 161
 - wml::des::imbg, 161
 - wml::des::imgdot, 162
 - wml::des::lowsrc, 162
 - wml::des::navbar, 151–156, 156, 162
 - wml::des::preload, 162
 - wml::des::rollover, 162
 - wml::des::space, 162
 - wml::des::typography, 162
 - .wml extension, 134, 135
 - .wml files, 133, 146
 - wml::fmt::xtable, 156–161
 - WML macro processor, 170
 - .wmlrc, 176
 - I option set in, 136
 - variables defined in, 168–169
 - WML configured with, 167–170
 - wml::std::info, 179
 - wml::std::page, 147–149
 - wml::std::tags, 162
 - wml::std::toc, 149–150, 151
 - wmu: Website META Language Upgrade Utility, 134
 - Word character (\w), 83
 - World Wide Web, xxx, 3
 - invention of, 33n
 - pictorial overview of, 4
 - Write mode, opening files in, 98
 - w switch, 59
 - w3g, 3
 - www.example.com, 3, 5

www.opensourcewebbook.com project,
46, 47, 48, 49, 179–183
bread crumb trail in, 181
errata, 182, 183
left rail for, 181
new tags defined for, 179–180
PAGE_BODY included, 182
page without right rail, 182
page with right rail, 182–183, 184
RIGHTRAIL section, 183
right rail—yes or no?, 181–182
title varied for, 180

X

Xconfigurator program, 31
xdvi, xxxii
x86 processor (Intel), 11
XML. *See* Extensible Markup Language

xpdf, xxxii
XSLT, Embperl integration with, 329
<xtable> definition, 159
<xtable> tag, 158

Y

Yahoo, 33
YAPL, and PHP, 378
YaST, 12
YDL, xxxii
YEAR, in MySQL, 113
\$year, 374

Z

Zero-based arrays, in PHP, 385
Zero pairs, 68
zip field, in manufacturers table, 304
zsh, 22



informIT

www.informit.com

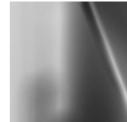
YOUR GUIDE TO IT REFERENCE

Articles



Keep your edge with thousands of free articles, in-depth features, interviews, and IT reference recommendations – all written by experts you know and trust.

Online Books



Answers in an instant from **InformIT Online Book's** 600+ fully searchable on line books. For a limited time, you can get your first 14 days **free**.

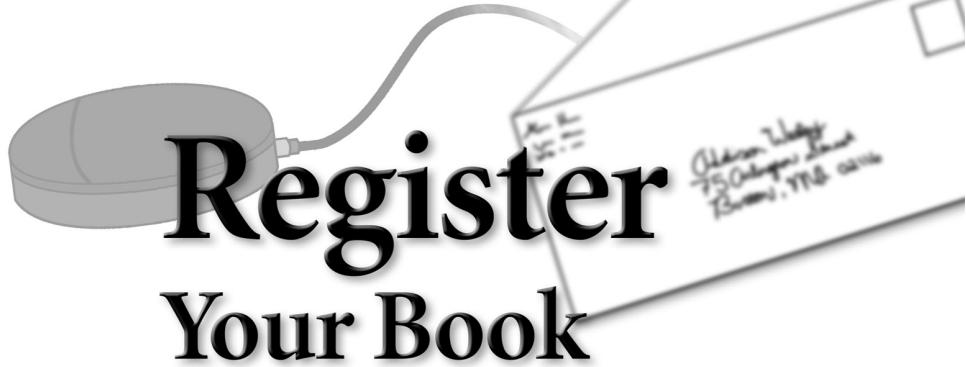


Catalog



Review online sample chapters, author biographies and customer rankings and choose exactly the right book from a selection of over 5,000 titles.





Register Your Book

at www.awprofessional.com/register

You may be eligible to receive:

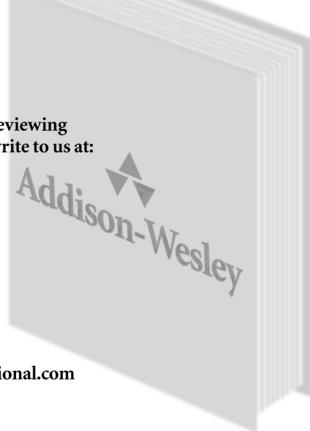
- Advance notice of forthcoming editions of the book
- Related book recommendations
- Chapter excerpts and supplements of forthcoming titles
- Information about special contests and promotions throughout the year
- Notices and reminders about author appearances, tradeshows, and online chats with special guests



Contact us

If you are interested in writing a book or reviewing manuscripts prior to publication, please write to us at:

Editorial Department
Addison-Wesley Professional
75 Arlington Street, Suite 300
Boston, MA 02116 USA
Email: AWPro@aw.com



Visit us on the Web: <http://www.awprofessional.com>