

Apress™

Books for Professionals by Professionals

Index

A Programmer's Introduction to PHP 4.0

by William Jason Gilmore

ISBN # 1-893115-85-2

Copyright ©2001 William J. Gilmore. World rights reserved. No part of this publication may be stored in a retrieval system, transmitted, or reproduced in any way, including but not limited to photocopy, photograph, magnetic or other record, without the prior agreement and written permission of the publisher.

info@apress.com

Index

NUMBERS AND SYMBOLS

- [0-9] character range for POSIX regular expressions, 177
- 1.inc, 2.inc, and 3.inc files for window.php example, 400–401
- [[:alnum:]] character class for POSIX regular expressions, 178
- [[:alpha:]] character class for POSIX regular expressions, 178
- & (ampersand)
 - converting to HTML, 201
 - using with assignment of variables by reference, 49
 - in XML entity references, 364
- * (asterisk)
 - escaping, 160
 - flag after character sequences in POSIX regular expressions, 177
 - operator in DTDs, 370
- */ (asterisk slash) used in comments, 28–29
- @ (at sign) operator, 56
- [a-z], [A-Z], and [a-Z] character ranges for POSIX regular expressions, 177
- \ (backslash), 34, 184
 - using to link to PHP-enabled pages, 216
 - using with POSIX regular expressions, 178
- `` (backticks), enclosing commands with, 158, 418
- \[0-7]{1,3} character sequence, 34
- \\ character sequence, 34
- \\$ character sequence, 34
- : (colon) in authentication text file, 429
- , (comma)
 - operator in DTDs, 370
 - using with initialization variables, 70–71
- { } (curly brackets)
 - referencing multidimensional arrays with, 97–98
 - in sample template file, 302
- [[:digit:]] character class for POSIX regular expressions, 178
- \$ (dollar sign) string delimiter, 41
- " (double quotation mark) string delimiter, 33–34
- "" (double quotation marks),
 - converting to HTML, 201
 - using to link to PHP-enabled pages, 216
- // (double slash) used in comments, 28
- = (equals) sign used with ASP-style escape tag, 24–25
- > (greater than) operator, 56, 62
- >= (greater than or equal to) operator, 56, 62
- < (left angle bracket) character,
 - converting to HTML, 201
- < (less than) operator, 56, 62
- <= (less than or equal to) operator, 56, 62
- * (multiplication) operator, 56–57
- ! ~ operator, 56
- %=&= |= ^= operator, 56
- && || operator, 56
- &^| operator, 56
- + - . operator, 56
- ++ — operator, 56
- / * % operator, 56
- <<< operator in Here doc syntax, 34–35
- = += *= / = .= operator, 56
- == != === <> operator, 56

Index

- ?: operator, 56
 - () (parentheses)
 - operator, 56
 - used in DTDs, 367
 - | (pipe) operator
 - in DTDs, 370
 - using with POSIX regular expressions, 176–177
 - + (plus)
 - flag after character sequences in POSIX regular expressions, 177
 - operator in DTDs, 370
 - # (pound symbol)
 - placing in href for JavaScript, 397–398
 - used in comments, 28
 - ? (question mark)
 - flag after character sequences in POSIX regular expressions, 177
 - operator in DTDs, 370
 - > (right angle bracket) character,
 - converting to HTML, 201
 - ; (semicolon) in XML references, 364
 - ' (single quotation) marks, using with strings, 33
 - / (slash), using with Perl style regexp metacharacters, 184
 - /* (slash asterisk) used in comments, 29
 - [[:space:]] character class for POSIX regular expressions, 178
 - [] (square brackets)
 - using with indexed arrays, 93
 - using with POSIX regular expressions, 177
 - \x[0-9A-Fa-f]{1,2} character sequence, 34
 - | (vertical bar), using with list() language construct, 95
- A**
- a file mode, 146
 - a+ file mode, 146
 - About This Site static page, 228
 - absolute value of integers, 103
 - Access and PHP, 288–291
 - access counter project, 168
 - Adabas D database servers, support for, 283
 - add_bookmark.php file, 295–296
 - add_guest.php file, 257–259
 - addition operator, 56
 - address-book creation project, 314–320
 - address_sql() SQL parsing method, 317–319
 - Advertising Information static page, 228
 - ampersand (&)
 - converting to HTML, 201
 - using with assignment of variables by reference, 49
 - in XML entity references, 364
 - AND operator, 60
 - AND XOR OR operator, 56
 - Apache module
 - installing PHP as, 13–14
 - restriction from using safe_mode_boolean security configuration option with, 412
 - Apache server, downloading, 10–11
 - Apache Web site, 11, 19, 411
 - Apache Week Web site, 411
 - Application-level security safeguards, 7
 - (array) cast operator, conversion for, 47
 - array data type, 35–38
 - array elements
 - creating HTML table from, 105–106
 - definition of, 93
 - locating, 98–100
 - array indexes, impact of asort() function of, 113
 - array keys, assigning file contents to, 303–304
 - array() language construct, creating arrays with, 94–95
 - array_count_values() function, 110–111
 - array_flip() function, 109, 203
 - array_keys() function, locating array elements with, 99–100
 - array_merge() function, using with arrays, 117
 - array_pad() function, 102–103
 - array_pop() function, 101–102
 - array_push() function, 101
 - array_reverse() function, 108

- array_shift() function, 102
- array_slice() function, 117–118
- array_sorting.inc function library, 90–91
- array_splice() function, 118–119
- array_unshift() function, 102
- array_values() function, locating array elements with, 100
- array_walk() function, using with arrays, 107–108
- arrays
 - adding elements to, 100–103
 - advancing pointers to next elements in, 104–108
 - counting frequency of values appearing in, 110–111
 - creating, 93–96
 - expanding to precise sizes, 102–103
 - flipping values in, 109
 - managing sizes of, 109–111
 - merging, 117
 - padding, 102–103
 - popping values from end of, 101–102
 - pushing values to end of, 101
 - reading files into, 153
 - removing elements from, 100–103
 - removing elements from beginnings of, 102
 - returning key-value pairs for current pointer positions, 104–108
 - returning number of elements in, 109–110
 - returning slices of, 117–118
 - rewinding pointers of, 104
 - shortening lengths of, 101–102
 - sorting, 111–117
 - sorting based on predefined criteria, 116–117
 - sorting elements in reverse order in, 112–115
 - sorting randomly, 119
 - traversing, 103–109
 - using array_flip() function with, 109, 203
 - using array_merge() function with, 117
 - using array_reverse() function with, 108
 - using array_splice() function with, 118–119
 - using array_walk() function with, 107–108
 - using arsort() function with, 113–114
 - using asort() function with, 113
 - using count() function with, 110
 - using each() function with, 104–106
 - using krsort() function with, 115
 - using ksort() function with, 114–115
 - using next() function with, 107
 - using prev() function with, 107
 - using reset() function with, 104
 - using rsort() function with, 112–113
 - using shuffle() function with, 119
 - using sort() function with, 111–112
 - using uasort() function with, 117
 - using uksort() function with, 117
 - using usort() function with, 116–117
- arsort() function, using with arrays, 113–114
- asort() function, using with arrays, 113
- ASP-style tags for escaping to PHP, 24–25
- asp_tags [on | off] configuration directive, 21
- assignment of variables by value and reference, 48–49
- assignment operators, 58–59
- associative and indexed array indexes, mixing, 38
- associative arrays, 36–38
 - creating, 94
 - using foreach control structure with, 72–73
 - using mysql_fetch_array() function with, 276–277
- associativity of operators, 57
- asterisk (*)
 - escaping, 160
 - flag after character sequences in POSIX regular expressions, 177
 - operator in DTDs, 370
- asterisk slash (*/) used in comments, 28–29
- at sign (@) operator, 56

Index

- attribute
 - declarations in DTDs, 370–372
 - types for DTDs, 372–374
- attributes
 - manipulating in classes with OOP, 124
 - of XML documents, 364
- authenticate.txt file, 428–429
- authentication variables, checking, 427–428
- auto_start session-handling directive, 335
- autoincrement and autodecrement operators, 56, 60
- Automation Technologies database drivers for ODBC, 282
- B**
- backslash (\), 34, 184
 - linking to PHP-enabled pages with, 216
 - using with POSIX regular expressions, 178
- backticks (`), enclosing commands with, 158, 418
- Bakken, Stig user affirmation, 8
- basename() function
 - using with server files, 160–161
 - versus dirname() function, 165–166
- bill against credit limit example, 86–87
- binary representations of decimal integers, 62
- binary tar files, downloading PHP as, 10–11
- bitwise AND operator, 56
- bitwise operators, 62–63
- bitwise OR operator, 56
- bitwise XOR operator, 56
- block allocation used with stat()
 - function, indexed value for, 161
- body, header, and footer example, 224–226
- body section example, 223
- book.html template, 314–315
- bookmark repository project, 291–299
- bookmarks.xml example, 387–389
- boolean AND operator, 56
- Boolean logic, using with DTDs, 370
- boolean NOT and bitwise NOT, operator for, 56
- boolean OR operator, 56
- boolean values, 39–40
-
 tags, converting newline (\n) characters to, 200
- brackets ([]), using with POSIX regular expressions, 177
- break statement control structure, 74–76
- browscap.ini file, 396
- browser, displaying information about, 52
- browser attributes, displaying, 394–395
- browser capabilities, retrieving with get_browser() function, 394
- browser detection example, 207–211
- browser.ini file, 394
- byte size used with stat() function, indexed value for, 161
- C**
- c 5 (-n 5 for Windows) parameter, using with ping, 158
- cache_expire session-handling directive, 336
- cache_limiter session-handling directive, 336
- calendar project, 77–80
- carriage return string delimiter, 34
- case sensitivity of identifiers, 41
- cast operators for variables, 47
- CBC: Cipher Block Chaining mode, for mcrypt() data-encryption package, 422
- CCVS (Credit Card Verification System), 426
- CDATA attribute type for DTDs, 372
- CDATA in XML documents, 363
- CFB: Cipher Feedback mode, for mcrypt() data-encryption package, 423
- character classes for POSIX regular expressions, 178
- character data, indicating in DTDs, 368–369

- character handling, 34–35
- character ranges and sequences for
 - POSIX regular expressions, 177
- characters, restricting ranges in DTDs, 374
- chdir() function, changing file
 - directories with, 167
- checkbox mouse-oriented form entity, 234–235
- chgrp() function, changing groups of
 - files with, 163
- chmod() function, changing mode of
 - filename to permissions with, 163
- chop() function, returning strings minus
 - whitespace and newlines with, 189
- chown() function, changing ownership
 - of filenames with, 163
- Clark Cooper, 376
- Clark, James, 376
- class abstraction, role in OOP (object-oriented programming), 132–133
- classes
 - definition of, 122–123
 - relationship to objects, 38–39
 - role in OOP (object-oriented programming), 122–123
- client operating system and browser, determining, 209–211
- closedir() function, closing directory
 - streams with, 166
- cnet Web site as example of table sorter, 280
- code volume, limiting with variable
 - functions, 88–89
- coding issues associated with security, 417–419
- colon (:) in authentication text file, 429
- color and screen resolution, detecting
 - with JavaScript, 396–397
- COM (Component Object Model), 402–408
- COM (Component Object Model)
 - functionality, using to interact with Microsoft Word, 406–408
- COM (Component Object Model)
 - objects
 - implementing methods for, 403–404
 - instantiating, 403
- com_get() function, retrieving COM
 - object properties with, 404–405
- com_set() function, setting object
 - attributes with, 405–406
- comma (,)
 - operator in DTDs, 370
 - using with initialization variables, 70–71
- command output, returning in
 - unformatted form, 159
- commenting code, 28–29
- comments in XML documents, 365
- company sample database, 263
- comparison operators, 61–62
- concatenating strings, 59
- concatenation operator, 56
- conditional expressions, omitting
 - components of, 71
- conditional *if* statement, using with
 - one-script form processing, 242
- conditional/iterative statements,
 - building functions into, 86
- conditional statements, using in_array()
 - function with, 98–99
- conditionals, components of, 69
- configuration files, hiding, 415–417
- configuration issues associated with
 - security, 412–417
- constants, defining, 53
- constructors, role in OOP (object-oriented programming), 126
- Contact Us static page, 228
- continue statement control structure, 76–77
- control statements, using in_array
 - function with, 98–99
- control structures, definition of, 63
- cookbook.dtd file, 366–367
- cookie_domain session-handling
 - directive, 335
- cookie_lifetime session-handling
 - directive, 335, 338

Index

- cookie_path session-handling directive, 335
- cookie_serialize_handler session-handling directive, 335
- cookies
 - checking for, 351–352
 - components of, 324–335
 - definition of, 323–324
 - determining domains for, 335
 - determining lifetimes of, 335
 - determining parent path directory for, 335
 - and PHP, 326–330
 - resources for, 333
 - retrieving user information from databases with, 329
 - setting, 326
 - specifying names for, 328
 - storage format details of, 325
 - storing page-formatting preferences with, 327–328
 - using session.name directive with, 335
 - using variables for, 326
- Cooper, Clark, 376
- copy() function, copying files with, 164
- Cosby, Randy user affirmation, 8–9
- count() function, using with arrays, 110
- Credit Card Verification System (CCVS), 426
- crypt() predefined function, securing data strings with, 420–421
- curly brackets ({})
 - referencing multidimensional arrays with, 97–98
 - in sample template file, 302
- Cybercash ecommerce functions, 425–426
- cyberlib.php API, using with Cybercash, 425
- cyScape, Inc. browser.ini file, 394
- D**
- data
 - retrieving with mysql_fetch_array() function, 276–277
 - retrieving with mysql_fetch_row() function, 275–276
 - storing in MySQL databases, 340–341
- data encryption, 419–424
- data files, hiding, 415–417
- data sets, producing with mysql_query() function, 273–275
- data type casting, 47–48
- data type juggling, 46–47
- database drivers, for ODBC, 282
- database normalization, 430
- database querying, providing functional interface for, 271
- database servers, PHP support of, 265
- database table, for creating address book, 314
- database tools, for SQL, 262
- databases
 - selecting from servers, 269–270
 - storing multiple user authentication in, 430–431
- date() function, 26
- date.php sample page, 215–216
- decimal integers and binary representations, 62
- default document type, setting with com_set() function, 405–406
- default enumerated attributes, for DTDs, 373–374
- default tags, for escaping to PHP, 23
- defined classes, retrieving with get_declared_classes() function, 141
- DELETE FROM command in SQL, 264
- destroy() function, using with destructors, 127
- destructors, role in OOP, 127
- device used with stat() function, indexed value for, 161
- Dézéus, Jean-Pierre, 348
- die() command
 - using with class abstraction, 132–133
 - using with fopen() function, 146
- directories
 - specifying in user's home directory for PHP scripts, 414–415

- working with, 165–167
- directory structure, example of, 165
- dirname() function, determining paths with, 165–166
- disable_functions string security configuration option, 414
- display_errors [on | off] configuration directive, 22
- division operator, 56
- doc_root string security configuration option, 414
- document type, setting default with com_set() function, 405–406
- DocumentRoot file in Apache's httpd.conf file, 415
- documents
 - obtaining outside document roots as security measure, 416
 - parsing with xml_parse() function, 383
- dollar sign (\$) string delimiter, 41
- domain component of cookies, 324–325
- (double) cast operator, conversion for, 47
- double quotation (") marks
 - converting to HTML, 201
 - using to link to PHP-enabled pages, 216
- double quotation mark (") string delimiter, 33–34
- double slash (//) used in comments, 28
- do..while control structure, 68–69
- DTD (document type definition)
 - attribute declarations in, 370–372
 - attribute types for, 372–374
 - closing, 369
 - components of, 369–376
 - creating for XML documents, 366–376
 - element operators for, 370
 - entity declarations in, 374–375
 - indicating character data in, 368–369
 - role in SGML, 359
 - in XML prologs, 362
- Dynamic Apache Module, installing PHP as, 15–17
- dynamic date insertion example, 25–26

- dynamic forms construction, 252–254
- dynamic HTML tag example, 26
- dynamic pop-up windows, building, 397–402
- dynamic pull-down menu, generating, 253–254

E

- each() function, using with arrays, 104–108
- Easysoft database drivers, for ODBC, 282
- ECB: Electronic Code Book mode, for mcrypt() data-encryption package, 423
- ecommerce functions, 424–426
- EGPCS (Environment, Get, Post, Cookie, Server) values, 334
- element attributes in DTDs, 370–372
- element declarations in DTDs, 369–370
- element operators in DTDs, 370
- elements
 - removing from arrays, 101–103
 - returning number of in arrays, 190–110
 - of XML documents, 363–364
- elseif statement control structure, 65
- enable-trans-sid configuration flag, role in session handling, 334
- encapsulation
 - advisory about, 125–126
 - definition of, 121–122
- enclosure bracketing, 67
- encryption, 419–424
- encryption modes, resource for, 422
- encryption types, for crypt() predefined function, 420
- end() function, using with arrays, 107
- entities and entity attributes, for DTDs, 374
- entity declarations, in DTDs, 374–375
- entity references, in XML documents, 364–365
- entropy_file session-handling directive, 336
- entropy_length session-handling directive, 336

Index

- enumerated attributes, for DTDs, 373
 - equality operators, 61
 - equals (=) sign, used with ASP-style
 - escape tag, 24–25
 - ereg() POSIX-extended regexp function,
 - searching strings with, 179–180
 - ereg_replace() POSIX-extended regexp
 - function, searching strings with, 180–181
 - eregi() POSIX-extended regexp function,
 - searching strings with, 181
 - eregi_replace() POSIX-extended regexp
 - function, searching strings with, 182
 - error concealment, operator for, 56
 - error values, retrieving with
 - xml_get_error_code() function, 384
 - error_log [filename] configuration
 - directive, 22
 - error_reporting [1-8] configuration
 - directive, 21
 - errors, configuring logging of, 22
 - escape tags, configuring, 20
 - escapeshellcmd() function security
 - feature, 160, 417–418
 - events calendar project, 77–80
 - events.txt file
 - displaying contents to browser, 79
 - using to send user information via email, 246–247
 - exec() function, executing external
 - programs with, 157–158
 - Expat 2.0 parser, 376
 - expiration date component of cookies, 324
 - explode() function, dividing strings with, 195
 - expressions, 55, 66
 - external entity declarations in DTDs, 375
 - using with
 - xml_set_external_entity_ref_handler() function, 379–380
 - using
 - xml_set_unparsed_entity_decl_handler() function for, 382
- ## F
- factorial example, 67–69
 - fclose() function, closing files with, 146–147
 - fgets() function, reading from files with, 150–151
 - fgetss() function, reading from files with, 151–152
 - file characteristics, displaying and
 - modifying, 162–164
 - file() function, reading files into indexed
 - arrays with, 153
 - file I/O (input/output), opening and
 - closing, 145–147
 - file modes, 145–146
 - file parsing, role in developing advanced
 - templates systems, 303, 305–307
 - file pointers, opening with fopen()
 - function, 154–155
 - file printing, role in developing
 - advanced template systems, 303, 307–311
 - file registration, role in developing
 - advanced template system, 303–304
 - file system I/O block size used with stat()
 - function, indexed value for, 161
 - file system, working with, 160–162
 - file_exists() function, verifying existence
 - of files with, 143–144
 - filegroup() function, managing file
 - characteristics with, 163
 - filenames, configuring error logging to, 22
 - fileowner() function, returning user ID
 - with, 164
 - fileperms() function, returning
 - permissions of files with, 163
 - files
 - building components of, 221
 - converting to HTML, 200–205
 - copying and renaming, 164
 - deleting, 165
 - including in templates, 217–221
 - outputting line by line, 151

- reading from, 148–152
- reading into arrays, 153
- redirecting to output, 153–154
- registering, 303–304
- verifying existence and sizes of, 143–144
- writing to, 147–148
- `filesize()` function, determining file size with, 144, 149
- `#FIXED` attribute flag, for DTDs, 372
- flags after character sequences, in POSIX regular expressions, 177–178
- (float) cast operator, conversion for, 47
- floating-point data type, 32
- footer file example, 222–223
- footer, header, and body example, 224–226
- `fopen()` function, opening files with, 145
- for loop control structure, 69–71
- foreach control structure, 71–73, 76
- form entities
 - checkbox mouse-oriented type of, 234–235
 - enclosing in HTML tags, 231–232
 - keyboard-oriented type of, 232–234
 - mouse-oriented type of, 234–238
 - password text box type of, 233
 - pull-down menu mouse-oriented type of, 236
 - radio button mouse-oriented type of, 235
 - Reset button mouse-oriented type of, 238
 - Submit button mouse-oriented type of, 237
 - text area box keyboard-oriented type of, 233–234
 - text box keyboard-oriented type of, 232–233
- form fields, displaying missing and erred types of, 250–252
- form information
 - adding to text files, 248–249
 - error checking, 250–252
 - passing from one script to another, 55, 240–243
 - sending to email addresses, 243–245
 - sending via email, 246–247
- form interface for `add_browser.php` file, 296
- form processing with one script, 242–243
- forms
 - constructing dynamically, 252–254
 - example of, 238–240
 - resource for, 232
- `fpassthru()` function, executing external programs with, 159
- `fputs()` function, comparing to `fwrite()` function, 148
- `fread()` function, reading from files with, 149–150
- freeware drivers, for ODBC, 282
- `fsockopen()` function, opening socket connections with, 155–156
- `ftp://` files, opening, 145
- `func_get_art()` function, using with method overloading, 134
- `func_num_args()` function, using with method overloading, 134
- function declarations, nesting, 83–84
- function definitions, 82
- function libraries, building, 90–91
- function names, 82, 414
- function parameters, declaring, 43
- functional iteration, 88
- functions
 - applying to elements in arrays, 107–108
 - building into conditional/iterative statements, 86
 - definition of, 81
 - nested type of, 83–84
 - recursive type of, 88
 - restricting with `safe_mode_boolean` security configuration option, 413
 - returning values from, 85–88
 - role in OOP (object-oriented programming), 122–123
 - storing templates as, 226–227
 - using lists with, 87
 - variable type of, 88–89

Index

`fwrite()` function, writing to files with,
147–148

G

`gc_maxlifetime` session-handling
directive, 336

`gc_probability` session-handling
directive, 336

`get` method, using with forms, 232

`get_browser()` function, retrieving
browser capabilities with, 394

`get_class()` function, role in OOP, 139

`get_class_methods()` function, role in
OOP, 134

`get_class_vars()` function, role in OOP,
135

`get_declared_classes()` function, role in
OOP, 141

`get_html_translation_table()` function,
using to translate text to HTML
equivalent of, 202–203

`get_meta_tags()` function, searching
HTML files for META tags with,
204

`get_object_vars()` function, role in OOP,
136–137

`get_parent_class()` function, role in OOP,
139–140

`getlastmod()` function, using with server
files, 161

global authentication variables,
checking, 427–428

global variables, accessing, 44

`gpc` (`get/post/cookie`) in configuration
directive, 22

graphics, using external entities in DTDs
for pointing to, 375

greater than (`>`) operator, 62

greater than or equal to (`>=`) operator, 62

guestbook-creation project, 254–260

H

handler functions, for PHP and XML
documents, 378–382

handlers, specifying to serialize data for
cookies, 335

hashing algorithms for security,
421–423

header file example, 221–222

header, footer, and body example,
224–226

`header()` function, using with dynamic
pull-down menus, 253–254

Here doc syntax, 34–35

hexadecimal integers, 32

hexadecimal notation regular expression
pattern string delimiter, 34

Heyes, Richard, 312

hidden form values, 236–237

hit counter example, 339

horizontal tab string delimiter, 34

`htdocs` directory, testing Dynamic
Apache Module with, 16

HTML (HyperText Markup Language)

advent of, 360

converting from XML to, 387–389

converting strings and files to,
200–205

converting to plain text, 203–205

embedding in PHP, 25–27

embedding PHP in, 5

integrating PHP with, 9–10

HTML files, stripping tags from, 151–152

HTML tables, creating from array
elements, 105–106

HTML tags, enclosing forms entities in,
231–232

`htmlentities()` function, converting
characters to HTML entities
with, 201

`htmlspecialchars()` function, converting
special characters to HTML
with, 201–202

`http://` files, opening, 145

`httpd.conf` file

configuring to deny file extension
access, 417

modifying in Apache module, 15, 18

modifying in Dynamic Apache
module, 15–16

I

- i* modifier, using with Perl style regular expressions, 185
- I/O (input/output), opening and closing, 145–147
- IBM DB2 database servers, support for, 283
- ID, IDREF, and IDEREFS attributes, for DTDs, 372–373
- identical to operator, 56
- identifiers, 40–41
- if statement control structure, 64
- if statements, nesting, 65–66
- #IMPLIED attribute flag, for DTDs, 372
- implode() function, using with arrays to form strings, 195
- in_array() function, locating array elements with, 98–99
- include() function, using with templates, 217–219
- include_once() function, using with templates, 219
- included files, preventing viewing of, 222
- increment component of conditionals, 69
- index values of arrays, retrieving, 99–100
- indexed and associative array indexes, mixing, 38
- indexed arrays, 35–38
 - assigning cookie names with, 328
 - creating, 93
 - returning with stat() function, 161–162
 - using file() function with, 153
 - using mysql_fetch_row() function with, 275–276
- index.php file, 224–226
 - for bookmark repository project, 297–298
 - for template parser, 320
 - for visitor log project, 351–352
- information hiding, definition of, 121–122
- inheritance, role in OOP (object-oriented programming), 127–132
- initialization component, of conditionals, 69
- initialization file, creating for bookmark repository project, 292–295
- init.inc file
 - creating for bookmark repository project, 292–295
 - creating for guestbook-creation project, 254–257
 - creating for visitor log project, 348–351
- init.tpl filename, 219–220
- Inode device type used with stat() function, indexed value for, 161
- input files, forms of, 145
- INSERT command, in SQL, 264
- instantiation class, returning name of, 139
- (int) or (integer) cast operator, conversion for, 47
- integer data type, 31–32
- integer set, summing with recursive function, 88
- integers, absolute value of, 103
- internal entity declarations, in DTDs, 375
- {int.range} flag after character sequences in POSIX regular expressions, 177
- ODBC database servers, support for, 283
- is equal to operator, 56, 61
- is identical to operator, 61
- is not equal to operator, 56, 61
- is_dirname() function, verifying files as directories with, 166
- is_file() function, verifying existence of files with, 144
- is_readable() function, reading from files with, 149
- is_subclass_of() function, role in OOP, 140–141
- is_writeable() function, writing to files with, 147
- ISO-8859-1 encoding
 - changing to UTF-8 encoding, 387
 - converting data to and from, 384

Index

J

James Clark's Expat (XML Parser Toolkit)
package, 376

JavaScript

detecting color and screen
resolution with, 396–397
introduction to, 393–402
specifying `newWindow()` function
in, 397–398
verifying whether enabled for
particular browser, 396

JavaScript key, verifying existence of, 396

JavaScript templating strategies, 312

`join()` function, 196

K

key-value pair, returning with `each()`
function, 104–108

key values, sorting arrays by, 114–115

keyboard-oriented form entities,
232–234. *See also* mouse-
oriented form entities

keys for input array, returning with
`array_keys()` function, 99–100

keywords, using with entity references in
XML documents, 364–365

`krsort()` function, using with arrays, 115

`ksort()` function, using with arrays,
114–115

L

last access time used with `stat()`
function, indexed value for, 161

last change time used with `stat()`
function, indexed value for, 161

last modification time used with `stat()`
function, indexed value for, 161

`lаторre.txt` example, 154

left angle bracket (<) character,
converting to HTML, 201

Lerdorf, Rasmus, 3

less than (<) operator, 62

less than or equal to (<=) operator, 62

less than and greater than, operator for, 56

linking, to PHP-enabled pages, 215–216

links, displaying for browsers, 216

`list()` language construct

creating arrays with, 95–96

using with `each()` function, 105

lists, using with functions, 87

local variables, declaring, 42–43

`log_errors` configuration directive, 22

logical operators, 60–61

login, location of, 427

looping alternatives, 70

looping through statements, 67–68

`ltrim()` function, removing whitespace
and special characters from
strings with, 190

M

m modifier, using with Perl style regular
expressions, 185

magic number used in example, 76

`magic_quotes_gpc` configuration
directive, 22, 216

`mail()` function, using to send form
information to email addresses,
243–245

majordomo mailer Web site, 244

mapping arrays associatively, 94–95

markup, introduction to, 357–361

`max_execution_time` configuration
directive, 21, 414

`mcrypt()` data-encryption package,
422–423

`md5()` third-party hash algorithm
comparing to `crypt()` predefined
function, 421

encrypting data with, 419–420

memory

conserving when making query
calls, 271

specifying for scripts, 414

`memory_limit` integer security
configuration option, 414

META tags, searching HTML files for,
204–205

metacharacters, using with Perl style
regular expressions, 184

- method overloading, 133–135
 - method_exists() function, role in OOP, 138
 - methods
 - retrieving with get_class_methods() function, 134
 - role in OOP (object-oriented programming), 122–123, 124
 - mhash() module, 421–423
 - Microsoft Access, and PHP, 288–291
 - Microsoft Word documents
 - opening after instantiating new COM objects, 404
 - writing information to, 406–408
 - mkdir() function, creating directories with, 166
 - modifiers, using with Perl style regular expressions, 185
 - modulus operator, 56
 - mouse-oriented form entities, 234–238. *See also* keyboard-oriented form entities
 - MSDN Web site, 403
 - mysql_selct_db() function, selecting databases with, 269–270
 - multidimensional arrays
 - associative, 38
 - creating, 96–97
 - indexed, 37
 - referencing, 97–98
 - multilevel inheritance, 130–132
 - multiline comments, 28–29
 - multiple expression evaluation, 66
 - multiple inheritance, 130
 - multiple user authentication, 428–431
 - multiplication (*) operator, 56–57
 - MySQL
 - configuring, 267
 - implementing session-handling functions with, 345–347
 - installation of, 266–267
 - MySQL database server Web site, 12, 261, 266
 - MySQL databases
 - decoding session data stored in, 342
 - retrieving and formatting data in, 273–275
 - using session_encode() function to store data in, 340–341
 - MySQL predefined functions, 267–268
 - MySQL server
 - closing connection to, 270
 - connecting to, 267–269
 - MySQL table
 - for bookmark repository project, 291
 - for visitor log project, 347–353
 - mysql_affected_rows() function, 271–272
 - mysql_close() function, closing connections with, 270, 273–275
 - mysql_connect() predefined function
 - connecting to MySQL server with, 268–269
 - ignoring information passed to, 414
 - mysql_fetch_array() function, using with associative arrays, 276–277
 - mysql_fetch_row() function, assigning rows to indexed arrays with, 275–276
 - mysql_free_result() function, conserving memory with, 271
 - mysql_num_rows() function, 272–273
 - mysql_pconnect() function, 269
 - ignoring information passed to, 414
 - mysql_query() function, 271
 - using with mysql_result() function, 273–275
 - mysql_result() function, producing data sets with, 273–275
 - mySQL's MyODBC database drivers, 282
- ## N
- \n (newline) character, 34, 200
 - name component of cookies, 324
 - name session-handling directive, 335
 - nested functions, 83–84
 - nested if statement control structure, 65–66
 - Netcraft Web site, 4
 - new operator, 56
 - newline character string delimiter, 34
 - newline characters, removing from strings, 189

Index

- newWindow() function, specifying in
 - JavaScript, 397–398
- next() function, using with arrays, 107
- nl2br() function, converting newline (\n)
 - characters to
 tags with, 200
- NMTOKEN and NMTOKENS attributes,
 - for DTDs, 374
- [none] element operator, in DTDs, 370
- normalization of databases, 430
- NOT operator, 60
- notation declarations, using
 - xml_set_notation_declaration_handler() function with, 380
- O**
- (object) cast operator, conversion for, 47
- object data type, 38–39
- object instantiation, operator for, 56
- object/method mapping, verifying with
 - method_exists() function, 138
- object-oriented programming (OOP). *See*
 - OOP (object-oriented programming)
- object variables, obtaining with
 - get_object_vars() function, 136–137
- object.class parameter, of COM objects, 403
- objects
 - casting data types as, 48
 - creating and working with, 125
 - determining mode of creation for, 140–141
 - role in OOP (object-oriented programming), 123
- octal integers, 32
- octal notation regular expression pattern
 - string delimiter, 34
- ODBC (Open Data Base Connectivity)
 - introduction to, 261, 282–283
 - support by PHP, 283–288
- ODBC functions, using to interface with
 - Microsoft Access, 290–291
- odbc_close() function, closing
 - connections with, 284–285
- odbc_connect() function, establishing
 - connections with, 283–284
- odbc_exec() function, 286
- odbc_execute() function, executing
 - queries with, 285–286
- odbc_free_result() function, restoring
 - resources with, 287–288
- odbc_pconnect() function, 284
- odbc_prepare() function, using before
 - executing queries, 285
- odbc_result_all() function, formatting
 - and displaying rows with, 287
- OFB: Output Feedback mode for
 - mcrypt() data-encryption package, 423
- OOP (object-oriented programming),
 - 38–39
 - advantages of, 121
 - case convention for classes in, 124
 - and PHP (Hypertext Preprocessor), 122
 - role of class abstraction in, 132–133
 - role of constructors in, 126
 - role of destructors in, 127
 - role of get_class function in, 139
 - role of get_class_methods() function in, 134
 - role of get_class_vars() function in, 135
 - role of get_declared_classes() function in, 141
 - role of get_object_vars() function in, 136–137
 - role of get_parent_class() function in, 139–140
 - role of inheritance in, 127–132
 - role of is_subclass_of() function in, 140–141
 - role of method_exists() function in, 138
 - role of methods in, 124
 - role of objects in, 123
- opendir() function, opening directory
 - streams with, 166
- OpenLinkSoftware database drivers for
 - ODBC, 282
- operands in expressions, 55

- operating system, displaying
 - information about, 52
- operator associativity, 57
- operator precedence, 57
- operators in expressions, 55–63
- OR operator, 60
- owner group ID used with `stat()`
 - function, indexed value for, 161

P

- page-formatting preferences, storing
 - with cookies, 327–328
- page-generation project, 227–229
- parent path directory, determining for
 - valid cookies, 335
- parentheses `()`
 - operator, 56
 - used in DTDs, 367
- parenthesized substrings, back-
 - referencing, 181
- `parse_str()` function, parsing strings into
 - variables with, 194
- parser resources, freeing with
 - `xml_parser_free()` function, 383
- parsers, creating with
 - `xml_parser_create()` function, 383
- parsing
 - files, 303, 305–307
 - functions, 382–383
- `passthru()` function, executing external
 - programs with, 159
- password text box form entity, 233
- password verification system, using
 - `md5()` hash algorithm as, 419–420
- passwords
 - hardcoding in scripts, 428
 - location of, 427
 - storing and comparing with `crypt()`
 - predefined function, 420–421
- path component of cookies, 325
- pathnames, specifying for directories, 166
- pattern matching, definition of, 175
- Payflow Pro service, 424
- `pclose()` function, closing files with, 155

- Perl style regular expression syntax, 183–188
- `pfsckopen()` function, using with socket
 - connections, 157
- `php://` files, opening, 145
- PHP Classes Repository templating
 - strategies, 312
- PHP (Hypertext Preprocessor)
 - characteristics of, 5–9
 - checking global authentication
 - variables for, 427–428
 - class declaration structure of, 123
 - COM functionality for, 402–406
 - commenting code, 28–29
 - configuring, 20–23
 - configuring frequency of script
 - execution in, 21
 - and cookies, 326–330
 - database support by, 265–266
 - decompressing distributions of, 12–13
 - designating directory of session files
 - for, 335
 - development of, 3–5
 - downloading, 10–11
 - as embedded language, 7
 - embedding HTML in, 25–27
 - escaping to, 23
 - handler functions for XML
 - documents, 378–382
 - installing and configuring on UNIX, 12–13
 - installing as Apache module, 13–14
 - installing as Dynamic Apache
 - Module, 15–17
 - installing on Windows 95/98/NT, 17–19
 - introductory example, 9–10
 - manual for, 283
 - and Microsoft Access, 288–291
 - and OOP (object-oriented
 - programming), 122
 - parsing functions for, 382–383
 - Perl compatible regexp functions, 185–188
 - POSIX-extended regexp functions, 179–183

Index

- PHP (*continued*)
 - specifying probability of activating garbage collection routine for, 336
 - Web site, 10
- PHP (Hypertext Preprocessor) module, and `safe_mode` security configuration option, 415
- PHP (Hypertext Preprocessor) scripts, embedding in documents, 27
- phpinfo.net Web site, 348
- php.ini file
 - location of, 20
 - security configuration options in, 412–417
 - session-handling directives in, 335–336
- PHPLib Base Library templating strategies, 312
- PI (processing instructions), in XML documents, 365
- pipe (|) operator
 - in DTDs, 370
 - using with POSIX regular expressions, 176–177
- plain text
 - converting HTML to, 203–205
 - converting to HTML, 200–203
- plus (+)
 - flag after character sequences in POSIX regular expressions, 177
 - operator in DTDs, 370
- pointers of arrays
 - advancing to next elements, 104–108
 - moving back on position, 107
 - moving to last position in, 107
 - rewinding, 104
- pointers, opening with `popen()` function, 154–155
- pop-up windows, building dynamic type of, 397–402
- `popen()` function, opening process file pointers with, 154–155
- POSIX regular expression syntax, 176–183
- post method, using with forms, 232
- pound symbol (#)
 - placing in href for JavaScript, 397–398
 - used in comments, 28
- precedence of operators, 57
- precedence ordering, operator for, 56
- precision [integer] configuration directive, 21
- predefined character ranges, for POSIX regular expressions, 178
- predefined variables, 50–52
- `preg_grep()` Perl style regexp, searching strings with, 188
- `preg_match()` Perl compatible regexp function, searching strings with, 186
- `preg_match_all()` Perl style regexp function, matching patterns in strings with, 186–187
- `preg_replace` Perl style regexp, replacing elements of arrays with, 187
- `preg_split` Perl style regexp, dividing strings with, 187–188
- `prev()` function, using with arrays, 107
- privilege tables, in MySQL, 267
- procedure, definition of, 85
- process file pointers, opening with `popen()` function, 154–155
- processing instructions in XML documents, 365
- programs, executing externally, 157–160
- projects
 - access counter, 168
 - building page generators, 227–229
 - creating a visitor log, 347–353
 - creating address books, 314–320
 - creating bookmark repository, 291–299
 - creating guestbooks, 254–260
 - setting up browser detection, 207–211
 - site map generator, 169–172
- pull-down menu mouse-oriented form entity, 236, 252–254

Q

- quantifiers, for POSIX regular expressions, 177
- queries, executing with `mysql_query()` function, 271
- queries, executing with `odbc_execute()` function, 285–286
- query calls, conserving memory for, 271
- question mark (?)
 - flag after character sequences in POSIX regular expressions, 177
 - operator in DTDs, 370
- queue, definition of, 101

R

- `\r` character sequence, 34
- `r` file mode, 146, 150
- `r+` file mode, 146
- radio button mouse-oriented form
 - entity, 235
- `rand()` function, using with cookies and UINs, 330–331
- `range()` language construct, creating arrays with, 96
- read only file mode, 146
- `readdir()` function, returning directory elements with, 167
- `readfile()` function, redirecting files
 - directly to output with, 153–154
- reading and writing file mode, 146
- (real) cast operator, conversion for, 47
- recursive functions, 46, 88
- `referer_check` session-handling directive, 336
- regex functions (POSIX extended), 179–183
- `register_globals` flag, role in session handling, 334–342
- registering files, 303–304
- regular expressions (regexps), 176–178
 - alternatives for, 193–199
 - resources for, 185
- relational databases, definition of, 262–263

- `rename()` function, renaming files with, 164
- `require()` function
 - using templates, 217
 - using with templates, 219
- `require_once()` function, using with templates, 220–221
- `#REQUIRED` attribute flag for DTDs, 372
- Reset button mouse-oriented form
 - entity, 238
- `reset()` function, using with arrays, 104
- `rewinddir()` function, resetting directory pointers with, 167
- Richard Heyes's Template Class
 - templating strategy, 312
- right angle bracket (>) character,
 - converting to HTML, 201
- Robin Cover/OASIS XML Cover Pages
 - Web site, 359
- root directory for PHP files, specifying, 414
- `root_element_name` in DTDs, 366–367
- rows affected by SQL queries, returning number of, 271–272
- rows
 - assigning to indexed arrays with `mysql_fetch_row()` function, 275–276
 - formatting and displaying with `odbc_result_all()` function, 287
- rows returned by SELECT query
 - statement, determining, 272–273
- `rows.addresses` template, 315–316
- RPM (RedHat Package Manager),
 - downloading PHP as, 10–11
- `rsort()` function, using with arrays, 112–113

S

- `s` modifier, using with Perl style regular expressions, 185
- Sablotron XSLT (XSL transformation) processor, 390
- safe mode, running PCP in, 6–7, 21

Index

- safe_mode_boolean security
 - configuration option, 412–413
- safe_mode_exec_dir string security
 - configuration option, 413
- safe_mode [on | off] configuration
 - directive, 21
- safe_mode security configuration option
 - and PHP module, 415–416
- sales tax function example, 85–86
- salt lengths for crypt() predefined
 - function, 420
- save_handler session-handling directive, 335–336, 343
- save_path session-handling directive, 335
- save_use_cookies session-handling
 - directive, 335
- scalar attributes, 48
- science.html file, 151–152
- scientific notation for floating-point
 - numbers, 32
- scope of variables, 42–46
- script embedding, 27
- script tags for escaping to PHP, 24
- scripts
 - hardcoding username and password
 - in, 428
 - specifying directories for, 414–415
 - specifying execution time of, 414
 - specifying memory for, 414
- search engines, building, 277–279
- searchengine.php file, 278–279
- security, 6–7
 - coding issues associated with, 417–419
 - configuration issues associated
 - with, 412–417
 - role of escapeshellcmd() function
 - in, 160
- security component of cookies, 325
- seenform hidden variable, using with
 - one-script form processing, 242
- SELECT command in SQL, 264
- SELECT query statement, determining
 - number of rows returned by, 272–273
- semicolon (;) in XML references, 364
- server directory, displaying structure of, 169–172
- server files, viewing and manipulating, 160–162
- server processes
 - opening file pointers to, 154–155
 - retrieving information about, 156–157
- servers
 - executing programs on, 157–160
 - pinging with exec() function, 157–158
- session data
 - decoding in MySQL databases, 342
 - deleting for
 - session_set_savehandler()
 - function, 344
 - specifying time for destruction of, 336
 - writing for
 - session_set_savehandler()
 - function, 344
- session handling, 333–336
- session-handling directives, 335, 338
- session-handling functions,
 - implementing with MySQL, 345–347
- session information, storing on servers, 335
- session pages, determining cache
 - control method for, 336
- session tracking, 333
- session variable values, retrieving for
 - session_set_savehandler()
 - function, 344
- session_decode() function, decoding
 - session data with, 341–342
- session_destroy() function, destroying
 - sessions with, 337–338
- session_encode() function, formatting
 - session variables with, 340–341
- session_id() function, returning user's
 - SID with, 338–339
- session_is_registered() function,
 - determining registration of
 - variables with, 339

- session_register() function, registering variable names with, 339
- session_set_save_handler() function, defining user-level session storage and retrieval functions with, 343–347
- session_start() function, initiating sessions with, 337
- session_unregister() function, destroying session variables with, 339–340
- session.name, using as cookie name, 335
- sessions
 - deleting for
 - session_set_savehandler() function, 344
 - destroying, 337
 - initiating, 337
- sessions, initiating with initial client requests, 335
- set_socket_blocking() function, using with socket connections, 155–156
- setcookie() function, 326
- SGML (Standard Generalized Markup Language), 358–359
- shift left or shift right, operator for, 56
- short tags for escaping to PHP, 23–24
- short_open_tag [on | off] configuration directive, 20
- shortcut assignment operators, 58
- shuffle() function, using with arrays, 119
- single-dimension indexed and associative arrays, 35–37
- single-line comments, 28
- single quotation (') marks, using with strings, 33
- site map generator project, 169–172
- site visitors, creating log of, 347–353
- site_init.tpl file, 226–227
- site.txt file, using to send user information via email, 246–247
- sizeof() function, using with arrays, 109–110
- slash (/), using with Perl style regexp metacharacters, 184
- slash asterisk (/*) used in comments, 28
- sniffer.php file, 209–211
- socket connections, opening, 155–157
- Solid database servers, support for, 283
- sort() function, using with arrays, 111–117, 119
- split() and spliti() POSIX-extended regexp functions, dividing strings with, 181–182
- split() POSIX-extended regexp function, dividing strings with, 195
- SQL parsing method, address_sql(), 317–319
- SQL (Structured Query Language), 262–265
- SQL queries, determining number of rows affected by, 271–272
- sql_regcase() POSIX-extended regexp functions, converting characters into bracketed expressions with, 183
- sql.safe_mode integer security configuration option, 414
- square brackets ([])
 - using with indexed arrays, 93
 - using with POSIX regular expressions, 177
- srand() function, using with cookies and UINs, 331
- standard notation for floating-point numbers, 32
- stat() function, using with server files, 161–162
- static variables, declaring, 45–46
- static.php file, 229
- str_pad() function, padding strings with, 189–190
- str_replace() function
 - replacing occurrences in strings with, 196–197
 - using to add user information to text files, 249
- str_replace() function, replacing occurrences in strings with, 199
- strcasecmp() function, performing case-insensitive string comparisons with, 192

Index

- strcmp() function
 - performing case-sensitive string comparisons with, 191–192
 - using with one-script form processing, 242–243
 - strcspn() function, comparing strings with, 193
 - (string) cast operator, conversion for, 47
 - string concatenation operator, using to reference multidimensional arrays, 97
 - string conversion for cast operators, 47
 - string data type, 33–35
 - string operators, 59
 - strings
 - comparing, 191–193
 - converting into uppercase and lowercase letters, 205–207
 - converting to HTML, 200–205
 - finding out lengths of, 190–191
 - padding and compacting, 189–190
 - strip_tags() function
 - using to convert special characters to HTML, 201
 - using to remove HTML and PHP tags from strings, 203–204
 - using to remove HTML tags from strings, 418–419
 - strlen() function, determining string lengths with, 190–191
 - strpos() function, finding position of first occurrence in strings with, 196
 - strrpos() function, locating first occurrence of characters in strings with, 196
 - strspn() function, comparing strings with, 192–193
 - strstr() function, returning remainder of strings with, 197
 - strtok() function, tokenizing strings with, 193–194
 - strtolower() function, converting strings to lowercase letters with, 206
 - strtoupper() function, converting strings to uppercase letters with, 206
 - strtr() function, converting characters to HTML with, 203
 - Submit button mouse-oriented form entity, 237
 - substr() function, returning parts of strings with, 197–198
 - substr_count() function, using with strings, 198
 - substr_replace() function, replacing portions of strings with, 197
 - subtraction operator, 56
 - SuExec Web site, 415
 - switch statement control structure, 73–75
 - Sybase SQL Anywhere servers, support for, 283
 - system commands, executing, 158
 - system() function, executing external programs with, 159
 - system-level security safeguards, 6–7
 - system setup, viewing predefined variables offered on, 50–52
- ## T
- \t character sequence, 34
 - table sorters, building, 280–282
 - tables in databases, purpose of, 262
 - tablesorter.php file, 280–281
 - tagName XML parser handler, using with xml_set_element_handler() function, 379
 - tags
 - role in DTDs, 368–369
 - using in tag elements, 370
 - team.txt file, using to send user information via email, 246–247
 - template class, expanding, 311–312
 - template parser, 320
 - template.class file, 309–311
 - templates, 217
 - book.html example of, 314–315
 - including files in, 217–221
 - optimizing for sites, 226–227
 - role of file parsing in development of, 303, 305–307
 - role of file registration in development of, 303–304
 - role of variable registration in development of, 303–305

- rows.addresses, 315–316
- sample file for, 302
- using include() function with, 218–219
- using include_once() function with, 219
- using require_once() function with, 220–221
- templating strategies, 312–314
- ternary operator, 56
- text area box keyboard-oriented form entity, 233–234
- text box keyboard-oriented form entity, 232–233
- text files, storing multiple user authentication information in, 428–430
- .tpl extension, preventing viewing of files with, 222
- track_vars configuration directive, 22
- track_vars flag, role in session handling, 334
- trim() function, removing whitespace from characters with, 190
- trinary operator, 62
- true/false evaluation, using control structures for, 63
- true/false values, 39–40
- TTL (time to live), determining for cached session pages, 336
- type casting, 47–48
- type juggling, 46–47

U

- uasort() function, using with arrays, 117
- ucfirst() function, capitalizing first letters of strings with, 206–207
- ucwords() function, capitalizing first letters of each word in strings with, 207
- UDP (User Datagram Protocol), using fsocket() function with, 155–156
- UIN (user identification number) stored in cookies, 324, 330–333
- uksort() function, using with arrays, 117
- Unified ODBC Functions, 283–288

- uniqid() function, creating UINs with, 330
- unique IDs, creating for window.php file, 400–401
- UNIX file system, resource for, 162
- UNIX, installing PHP on, 12–13
- unlink() function, deleting files with, 165
- unset() function, using with destructors, 127
- use_include_path parameter, using with fopen() function, 145–146
- user affirmations, 8–9
- user authentication, 426–431
- user callbacks, specifying as storage modules, 343–347
- user feedback form example, 238–240
- user information
 - adding to text files, 248–250
 - redirecting with mail() function, 244–245
 - retrieving from databases with cookies, 329
 - sending via email, 246–247
- user input
 - accepting, 417–419
 - escaping, 160
- user registration process, creating with UINs, 331–333
- user-requested information, sending with forms, 246–247
- user_dir string security configuration option, 414–415
- username, hardcoding in scripts, 428
- usort() function, using with arrays, 116–117
- UTF-8 encoding, changing from ISO-8859-1 to, 387
- utf8_decode() function, converting data to ISO-8859-1 encoding with, 384

V

- value component of cookies, 324
- values
 - denoting position of, 35–36
 - pushing to end of arrays, 101–102
 - returning from functions, 85–88

Index

- values for input arrays, returning with
 array_values() function, 100
- variable arrays, configuring, 22
- variable assignments, 48–49
- using list () language construct for, 95
- variable functions, 88–89
- variable registration, role in developing
 advanced template systems,
 303–305
- variable scope, 42–46
- variables, 41–46, 50
 - cast operators for, 47
 - predefined type of, 50–52
- vehicle types example of inheritance,
 128–130
- Vehicles superclass, multilevel
 inheritance model of, 130–132
- Verisign ecommerce functions, 424–425
- vertical bar (|), using with list() language
 construct, 95
- view_bookmark.php file, 296–297
- view_guest.php file, 257–258
- viewStats() function, using with visitor
 log project, 353
- visitor log project, 347–353
- visitors to sites, tracking with
 identification numbers, 353

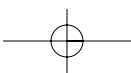
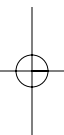
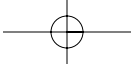
W

- w file mode, 146
- w+ file mode, 146
- W3 Web site, 207, 360
- weather information, displaying in pop-
 up window, 401
- while control structure, 67–68
- whitespace, removing from strings,
 189–190
- Widenius, Michael “Monty” user
 affirmation, 8
- window.php file, 399–400
- Windows 95/98/NT database drivers for
 ODBC, 282
- Windows 95/98/NT, installing PHP on,
 17–19
- WML (Wireless Markup Language), 361
- write only file mode, 146

X

- XLL (Extensible Linking Language), 361
- XML (Extensible Markup Language),
 360–361
 - components of documents,
 376–377
 - converting to HTML from, 387–389
 - parser options for, 386–387
 - parsing process for, 377–378
 - resources for, 376
 - sample document in, 361–362
- XML documents
 - attributes in, 364
 - comments in, 365
 - creating DTDs for, 366–376
 - elements of, 363–364
 - entity references in, 364–365
 - PHP handler functions for, 378–382
 - processing instructions in, 365
- XML parser, changing target encoding
 options from ISO-8859-1 to
 UTF-8, 387
- XML prologs, 362–363, 367
- xml_error_string() function, returning
 text description of error codes
 with, 385
- xml_get_current_column_number()
 function, pinpointing location
 of errors with, 385–486
- xml_get_current_line_number()
 function, retrieving line being
 currently parsed with, 385
- xml_get_error_code() function,
 retrieving error values with, 384
- xml_parse() function, parsing
 documents with, 383
- xml_parser_create() function, creating
 parser with, 383
- xml_parser_free() function, freeing
 parser resources with, 383
- xml_parser_get_option() function,
 retrieving XML parser options
 with, 386–387
- xml_parser_set_option() function,
 configuring XML parser options
 with, 386–387

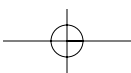
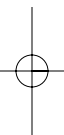
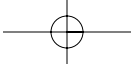
- `xml_set_character_data_handler()`
 - function, using with XML-document character data, 378
- `xml_set_default_handler()` function, unregistered components of XML documents, 378–379
- `xml_set_element_handler()` function, using with parse starting and ending element tags, 379
- `xml_set_external_entity_ref_handler()` function, using with external entity references, 379–380
- `xml_set_notation_declaration_handler()` function, using with notation declarations, 380
- `xml_set_object()` function, using with XML parser within objects, 380–381
- `xml_set_processing_instruction_handler()` function, using with processing instructions, 381–382
- `xml_set_unparsed_entity_decl_handler()` function, using with external entity references, 382
- XOR operator, 60
- XSL (Extensible Style Language), 360
- XSLT (XSL transformation) processor, 390
- Z
- Zend Web site, 4



The Story Behind Apress

APRESS IS AN INNOVATIVE PUBLISHING COMPANY devoted to meeting the needs of existing and potential programming professionals. Simply put, the “A” in Apress stands for the “author’s pressTM.” Our unique author-centric approach to publishing grew from con-versations between Dan Appleman and Gary Cornell, authors of best-selling, highly regarded computer books. They wanted to create a publishing company that emphasized quality above all—a company whose books would be considered the best in their market.

To accomplish this goal, they knew it was necessary to attract the very best authors—established authors whose work is already highly regarded, and new authors who have real-world practical experience that professional software developers want in the books they buy. Dan and Gary’s vision of an author-centric press has already attracted many leading software professionals—just look at the list of Apress titles on the following pages.



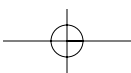
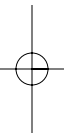
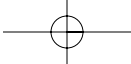
Would You Like to Write for Apress?

APRESS IS RAPIDLY EXPANDING its publishing program. If you can write and refuse to compromise on the quality of your work, if you believe in doing more than rehashing existing documentation, and if you are looking for opportunities and rewards that go far beyond those offered by traditional publishing houses, we want to hear from you!

Consider these innovations that we offer every one of our authors:

- Top royalties with no hidden switch statements. For example, authors typically only receive half of their normal royalty rate on foreign sales. In contrast, Apress' royalty rate remains the same for both foreign and domestic sales.
- A mechanism for authors to obtain equity in Apress. Unlike the software industry, where stock options are essential to motivate and retain software professionals, the publishing industry has stuck to an outdated compensation model based on royalties alone. In the spirit of most software companies, Apress reserves a significant portion of its equity for authors.
- Serious treatment of the technical review process. Each Apress book has a technical reviewing team whose remuneration depends in part on the success of the book since they, too, receive a royalty. Moreover, through a partnership with Springer-Verlag, one of the world's major publishing houses, Apress has significant venture capital behind it. Thus, Apress has the resources both to produce the highest quality books and to market them aggressively.

If you fit the model of the Apress author who can write a book that gives the "professional what he or she needs to know," then please contact any one of our editorial directors, Gary Cornell (gary_cornell@apress.com), Dan Appleman (dan_appleman@apress.com), or Karen Watterson (karen_watterson@apress.com), for more information on how to become an Apress author.



Apress Titles

ISBN	LIST PRICE	AVAILABLE	AUTHOR	TITLE
1-893115-01-1	\$39.95	Now	Appleman	Appleman's Win32 API Puzzle Book and Tutorial for Visual Basic Programmers
1-893115-23-2	\$29.95	Now	Appleman	How Computer Programming Works
1-893115-09-7	\$24.95	Now	Baum	Dave Baum's Definitive Guide to LEGO MINDSTORMS
1-893115-84-4	\$29.95	Now	Baum, Gasperi, Hempel, Villa	Extreme MINDSTORMS
1-893115-82-8	\$59.95	Now	Ben-Gan/Moreau	Advanced Transact-SQL for SQL Server 2000
1-893115-14-3	\$39.95	Now	Cornell/Jezak	Visual Basic Add-Ins and Wizards: Increasing Software Productivity
1-893115-85-2	\$34.95	Now	Gilmore	A Programmer's Introduction to PHP 4.0
1-893115-17-8	\$59.95	Now	Gross	A Programmer's Introduction to Windows DNA
1-893115-86-0	\$34.95	Now	Gunnerson	A Programmer's Introduction to C#
1-893115-10-0	\$34.95	Now	Holub	Taming Java Threads
1-893115-04-6	\$34.95	Now	Hyman/Vaddadi	Mike and Phani's Essential C++ Techniques
1-893115-79-8	\$49.95	Now	Kofler	Definitive Guide to Excel VBA
1-893115-75-5	\$44.95	Now	Kurniawan	Internet Programming with VB
1-893115-19-4	\$49.95	Now	Macdonald	Serious ADO: Universal Data Access with Visual Basic
1-893115-06-2	\$39.95	Now	Marquis/Smith	A Visual Basic 6.0 Programmer's Toolkit
1-893115-22-4	\$27.95	Now	McCarter	David McCarter's VB Tips and Techniques
1-893115-76-3	\$49.95	Now	Morrison	C++ For VB Programmers
1-893115-80-1	\$39.95	Now	Newmarch	A Programmer's Guide to Jini Technology
1-893115-81-X	\$39.95	Now	Pike	SQL Server: Common Problems, Tested Solutions
1-893115-20-8	\$34.95	Now	Rischpater	Wireless Web Development
1-893115-24-0	\$49.95	Now	Sinclair	From Access to SQL Server
1-893115-16-X	\$49.95	Now	Vaughn	ADO Examples and Best Practices

ISBN	LIST PRICE	AVAILABLE	AUTHOR	TITLE
1-893115-83-6	\$44.95	Now	Wells	Code Centric: T-SQL Programming with Stored Procedures and Triggers
1-893115-05-4	\$39.95	Now	Williamson	Writing Cross-Browser Dynamic HTML
1-893115-78-X	\$49.95	Now	Zukowski	Definitive Guide to Swing for Java 2, Second Edition

To order, call (800) 777-4643 or email sales*@apress.com .

Apress Titles Available Spring 2001

ISBN	TITLE	AUTHOR
1-893115-96-8	JSP and Servlet Development Patterns	Jorelid
1-893115-94-1	User Interface Design for Programmers	Spolsky
1-893115-95-X	Cryptography in C and C++	Welschenbach
1-893115-92-5	Java Collections	Zukowski
