



INDEX

- ; (semicolon), 111
- :: (scope resolution) operator, 289–293
- * (asterisk), 399, 541
- ` (backticks), 415
- // (comment symbol), 44–46
- @ (error suppression operator), 314
- | (or symbol), 541
- % (percent sign), 37, 42
- + (plus sign), 541
- ? (question mark), 541
- “ (quotation marks), 284, 444, 445–455
- ‘ (quote marks), 444, 445–455
- +/- (sign specifier), 37
- / (slash), 189

A

- a mode, 374
- a+ mode, 374
- abstract classes, 300–307
- abstract keyword, 300
- abstract methods, 300–307
- abstraction, 234
- access control, 234, 281–288
- ActiveXObject() function, 492–494
- addAttachment() method, 478
- addAttribute() method, 556
- addBcc() method, 470
- addCc() method, 470
- addChild() method, 556
- addHTMLImage() method, 476–477, 479
- addItem() method, 333–334, 344
- addresses table, 182
- addRule() method, 461–467
- addUser() function, 149
- addUser() method, 145
- Advanced Encryption Standard (AES), 158
- advcheckbox element type, 456
- AES (Advanced Encryption Standard), 158
- Ajax, 481–528
 - considerations, 528
 - database creation, 485–486
 - described, 481–482
 - example, advanced, 506–522
 - example, simple, 484–505
 - GET requests, 495, 526
 - HTTP status codes, 496
 - introduction to, 482–483
 - populating database, 486
 - POST requests, 523, 526
 - server requests, 482, 496
 - Web browser support, 482–528
 - XML and, 481, 483
- Ajax applications
 - creating HTML for, 500–503, 520–522
 - creating PHP script for, 487–490, 507–512
 - debugging, 523–528
 - testing, 504–505
 - testing PHP script, 491, 524–527
 - Web browser support, 482–528
 - writing JavaScript for, 492–499, 513–519
- Ajax model, 482
- Ajax transactions, 482, 483, 494
- alignment specifier, 37
- alphabetical sorts, 2–7

- alphanumeric rule, 462
 - Alternative PHP Cache (APC), 80
 - Amazon, 530
 - `&` entity, 536
 - Andrews, Tjobbe, 190
 - ANY type, 541
 - Apache server, 424
 - APC (Alternative PHP Cache), 80
 - API (application programming interface), 232
 - `'` entity, 536
 - application programming interface (API), 232
 - applications
 - Ajax. *See* Ajax applications
 - command-line, 435, 437
 - PuTTY, 422
 - running through URLs, 524
 - shopping cart, 212–223
 - SSH, 422
 - templates for, 190–196
 - Terminal, 400
 - testing, 504–505
 - Web. *See* Web applications
 - `applyFilter()` method, 461–463, 468
 - `$argc` array, 432
 - arguments
 - command-line, 432–436
 - constructors, 251
 - filter, 137
 - methods, 235, 240, 300, 301
 - sessions, 84
 - stored functions, 110
 - `$argv` array, 432
 - arrays. *See also specific arrays*
 - adding tasks to, 23
 - database-driven, 8–17
 - default form values, 461
 - filters, 142
 - form validation, 131, 461–465
 - multidimensional, 2–24
 - nested structure, 8–17
 - serialized, 89
 - sorting, 2–7
 - two-dimensional, 2
 - `array_sum()` function, 303
 - ASCII characters, 38
 - ASCII integer, 37
 - asterisk (*), 399, 541
 - `asXML()` method, 561
 - asynchronous behavior, 483
 - Asynchronous JavaScript and XML. *See* Ajax
 - Atom format, 568
 - `attach()` method, 395
 - attachments, email, 469, 478, 479
 - attributes. *See also specific attributes*
 - class, 235, 236, 244
 - description, 542
 - empty elements, 536–539
 - format of, 541–542
 - static, 294–299
 - values, 541–542
 - visibility of, 281
 - vs. variables, 236
 - XML, 536–539
 - Auth class, 144–156
 - `auth` element, 470
 - Auth package, 143–156
 - Auth type, 144
 - authentication. *See also* passwords; security
 - custom, 151–156
 - database-driven, 144
 - HTTP, 157, 372
 - logout feature, 154
 - Mail_Mime package, 469
 - MySQL, 145–149, 152
 - optional, 150
 - PEAR Auth package, 143–156
 - simple, 144–150
 - Web site, 143–156
 - authentication system, 143–156
 - Auth_HTTP authorization, 157
 - `autocomplete` element type, 456
 - `autoload()` function, 260–262, 306
 - autoloading classes, 260–262
 - autoloading functions, 306
- ## B
- `b` mode, 374
 - backticks (`), 415
 - backups, database, 375–384
 - base class, 265

base64_decode() function, 157, 166
 base64_encode() function, 157, 162
 base64_encode() version, 163
 BASE_URI constant, 54
 BASE_URL constant, 54
 BEGIN keyword, 111
 Benchmark package, 444–456
 benchmark timers, 445–450
 benchmarking, 444–456
 benchmarking functions, 451–456
 Benchmark_Iterate class, 451, 455
 binary data, 157, 380
 binary files, 382
 binary integer, 37
 binary mode, 374
 binary numbers, 37–38
 Boolean value, 84, 466
 bots, 135
 browsers. *See* Web browsers
 browsing, by category, 199–204
 button element type, 456
 buttons

- associating keys with, 398
- calculator, 386–396, 398
- radio, 456
- reset, 456
- submit, 456, 457, 460, 516, 520

C

cache-control directives, 75
 Cache-Control header, 75, 79
 caching

- APC, 80
- browsers, 74–80, 526
- considerations, 80
- in proxy servers, 74
- server-side, 80
- session, 80
- in Web pages, 74–80

 calculate() function, 387–388, 392
 calculator example, 386–398
 call() method, 280
 call_this_function() function, 495
 captcha test, 135
 carriage returns (\r), 124

cart class, 340–346
 catch block, 311, 316, 319
 catch statement, 310
 catching exceptions, 310–316
 categories

- browsing by, 199–204
 - no products in, 211
 - printing as links, 194
 - validating, 202

 categories table, 172, 176–178, 193, 194
 category ID, 194, 200–204
 category page, 200–204
 CDATA type, 542
 CGI (Common Gateway Interface), 418, 427
 character data, 554
 Character Type functions, 127
 characters

- ASCII, 38, 499
- control, 127
- escaped, 189, 499
- formatting, 37–38
- length, 510
- letters only, 127
- lowercase, 127
- newline, 124, 423, 427, 438
- out-of-range, 499
- printable, 127
- special, 536
- uppercase, 127
- UTF-8, 499
- white space, 127
- XML, 549

 check boxes, 32–33, 35, 456
 checkAuth() method, 145, 154
 checkbox element type, 456
 checkdate() function, 126, 132
 checkout systems, 224–232
 check_url() function, 359, 362
 check_urls() function, 361
 check_username() function, 498, 502, 505
 child classes, 264–270
 chmod utility, 431
 ciphers, 158
 class constants, 295
 class definition files, 261–262

- class keyword, 237
- classes. *See also specific classes*
 - abstract, 300–307
 - attributes, 235, 236, 244
 - autoloading, 260–262
 - base, 265
 - child, 264–270
 - COM, 414, 415
 - CSS, 362, 550–554
 - defining, 235–239
 - derived, 265
 - described, 234, 235
 - inheriting, 266–270
 - members of, 265
 - methods within, 235
 - names, 235, 243, 262
 - network-related, 354
 - OOP. *See* OOP classes
 - parent, 264–270
 - shopping cart, 328–346
 - static class attributes, 294–299
 - subclass, 265, 276, 288, 298
 - superclass, 265
 - visibility, 265
- clauses
 - IN, 220
 - else, 167, 334, 442, 490
 - else if, 494
 - “out,” 18
 - RETURNS, 111
 - WHERE, 106
- clear() function, 396
- CLI (Command Line Interface), 417–442
 - arguments, 432–436
 - errors, 431, 435, 436, 441–442
 - running scripts from, 424–431
 - testing code snippets, 422–423
 - testing installation of, 418–422
 - uses for, 417, 423
 - vs. CGI, 418
- CLI applications, 435, 437
- CLI php.ini file, 424
- CLI scripts
 - creating, 424–427
 - running, 428–431
- client URL (cURL) utility, 80, 368–372, 399, 401
- client/server model, 482
- client/server performance, 80
- client-side validation, 462, 464, 465–466
- “close” function, 84
- close() method, 326
- close_session() function, 94
- closing brackets, 46
- code. *See also* scripts
 - benchmarking, 444, 445–450
 - comments, 44–46
 - consistency and, 47
 - indenting, 48
 - structure, 47–48
 - style, 47–48
 - testing snippets of, 422–423
 - updates to, 46
 - white space in, 44, 48
- code blocks, 423
- code snippets, 422–423
- colors table, 172, 176, 179
- columns, 83, 98–100, 102, 413
- COM (Component Object Module), 404–415
- COM classes, 414, 415
- com() function, 404
- COM objects, 410
- com_exception class, 414
- Command Line Interface. *See* CLI
- commands. *See also specific commands*
 - running on server, 125, 415
 - SQL, 98, 173, 486
 - variables and, 125
- comment symbol (//), 44–46
- comments
 - errors and, 287
 - filtering, 140–142
 - regular expression patterns, 126, 132
 - writing, 44–46
 - XML documents, 533
- Common Gateway Interface (CGI), 418, 427
- compare rule, 462, 465
- Component Object Module (COM), 404–415
- compression, 80, 374–384
- CONCAT_WS() function, 107

- conditionals
 - commenting, 46
 - nested, 310
 - processing form data, 466, 468
 - session handlers, 92
 - switch, 64–65, 238, 324
 - validation, 202–204
 - configuration files
 - creating, 183–189
 - e-commerce application, 183–189
 - index pages, 197–198
 - Web applications, 50–57
 - connections. *See* database connections
 - connect_simple() method, 386, 390, 398
 - Console_Getargs package, 436
 - Console_Getopt package, 436
 - const keyword, 295
 - constants
 - accessing, 295
 - BASE_URI, 54
 - BASE_URL, 54
 - class, 295
 - DB, 54, 187
 - “magic,” 465
 - server-specific, 53–54, 187
 - STDIN, 437–439, 442
 - values, 295
 - construct() method, 251–252, 255, 271
 - constructors
 - arguments, 251
 - calling, 255
 - creating, 251–255
 - declaring, 255
 - default, 255
 - described, 251
 - Exception, 318, 327
 - inheritance, 271–275
 - subclass, 272–275
 - using, 252–256
 - vs. destructors, 256
 - containers, 385, 394
 - content modules, 66–69
 - contingencies, 45
 - control characters, 127
 - cookies, 157, 159–163, 372
 - counter, 116, 120, 121, 295–298, 427
 - count_to_100() function, 25
 - CREATE TABLE command, 10, 151–152, 174–178
 - credit card processing, 171, 224–232
 - cron utility, 399–401
 - crontab file, 399–401
 - crontab format, 399
 - cryptType function, 151
 - CSS classes, 362, 550–554
 - CSS files, 58, 190
 - ctype_ functions, 127
 - ctype_alnum() function, 131
 - ctype_digit() function, 132
 - cURL (client URL) utility, 80, 368–372, 399, 401
 - curl_errno() function, 372
 - curl_error() function, 372
 - curl_exec() command, 369
 - curl_getinfo() function, 372
 - curl_init() function, 368, 370
 - CURLOPT_POSTFIELDS option, 370
 - curl_setopt() function, 368
 - curly braces, 48
 - custom authentication, 151–156
 - customers table, 171, 174, 182
- ## D
- data
 - binary, 157, 380
 - encrypting/decrypting. *See* encryption
 - escaping, 189, 468
 - importing, 98
 - processing in forms, 466–468
 - reading from Web sites, 348–354
 - session, 82–95, 125
 - trimming, 461
 - validating. *See* validation
 - data source name (DSN), 144
 - database connections
 - closing, 195, 370–372, 490
 - configuration file, 184–186
 - errors, 14, 189, 379, 489
 - establishing, 184, 189, 495, 498
 - file compression and, 378–379

- RSS feeds and, 565
 - secure, 422
 - database creation
 - Ajax example, 485–486
 - e-commerce example, 171–182
 - multidimensional arrays, 10–11
 - database files, 57, 187
 - database-driven arrays, 8–17
 - databases. *See also* records; tables
 - adding tasks to, 12–18
 - advanced concepts, 81–122
 - backing up, 375–384
 - compression, 374–384
 - connections. *See* database connections
 - creating. *See* database creation
 - escaping data for, 468
 - names, 375
 - normalization, 172, 173, 176
 - populating, 486
 - selecting, 174
 - storing sessions in, 82–95
 - zip code, 96–109
 - date element type, 456
 - date() function, 423, 567
 - date_added attribute, 9
 - date_completed attribute, 9
 - day menu, 456
 - DB constant, 54, 187
 - DB container, 151
 - debug mode, 187–188
 - \$debug variable, 54
 - debuggers, 527
 - debugging. *See also* errors
 - Ajax applications, 523–528
 - error messages associated with, 188
 - JavaScript, 523–528
 - selecting mode, 187
 - debugging level, 54
 - decimal point, 390, 393, 396
 - DECLARE statement, 110
 - decrypting data, 164–167
 - DELETE query, 89
 - delete_item() method, 335–336
 - deleting
 - objects, 240, 242, 249, 259
 - products from shopping cart, 219, 223, 335–336
 - sessions, 89
 - delimiters, 111
 - derived class, 265
 - design patterns, 235
 - destroy events, 397
 - destruct() function, 256, 258
 - destructors
 - creating, 256–259
 - described, 256
 - inheritance, 271, 274
 - Java and, 259
 - OOP, 256–259
 - vs. constructors, 256
 - directories
 - backup, 376–383
 - permissions, 378, 380–381
 - session, 82, 125
 - site structure, 48
 - temporary, 82
 - writable, 378
 - display() method, 445, 466–468
 - display windows, 391, 395, 396
 - display_cart() method, 346
 - distance calculations, 103–109
 - dl() function, 391
 - DMCinsights.com, 170
 - do() function, 289
 - do() method, 244
 - document modeling, 540–541
 - Document Object Model. *See* DOM
 - Document Type Definitions (DTDs), 540–547
 - Dojo software, 518
 - DOM (Document Object Model), 496, 548
 - DOM parsers, 548, 556–561
 - drop-down menus, 15, 16
 - DSN (data source name), 144
 - DTDs (Document Type Definitions), 540–547
- ## E
- eBay, 530
 - echo() function, 37, 398, 447–454, 475
 - echo() statement, 423
 - ECMAScript language, 494

- e-commerce
 - checkout systems, 224–232
 - configuration file, 183–189
 - credit card processing, 171, 224–232
 - database creation, 171–182
 - overview, 170
 - payments, 232
 - showing products online, 205–211
 - templates, 190–196
- e-commerce example
 - browse feature, 199–204
 - index page, 197–198
 - product catalog, 205–211
 - shopping cart, 212–223
- e-commerce sites, 170, 174
- element type symbols, 541
- element types, 541
- elements
 - empty, 536–539
 - HTML, 456, 457
 - JavaScript, 500
 - password, 460
 - XML, 530, 536–539
- else clause, 167, 334, 442, 490
- else if clause, 494
- email
 - attachments, 469, 478, 479
 - avoiding abuse of, 124
 - carbon copies, 470
 - HTML, 469–479
 - images, 476–479
 - passwords, 131
 - plain text, 469–470, 474–479
- email address element, 458
- email addresses
 - carbon copies, 470
 - e-commerce sites, 174
 - rules, 462, 464, 465
 - validating, 131, 462
- email rule, 462, 464
- emailing errors, 53, 56, 184
- emailorblank rule, 462
- empty elements, 536–539
- EMPTY type, 541
- encapsulation, 234, 242, 288
- encodeURIComponent() method, 499
- encryption, 157–167
 - AES, 158
 - MCrypt, 157–167
 - passwords, 149, 154
- END keyword, 111
- entities, 536–539
- EOD delimiter, 32
- EOT delimiter, 32
- error handling, 55, 57, 188–189, 310, 327
- error messages, 55–56, 188, 324, 464
- error reporting, 125
- error suppression operator (@), 314
- error_log() function, 56, 188
- errors. *See also* debugging
 - CLL, 431, 435, 436, 441–442
 - COM, 410, 413
 - comments, 287
 - connection, 14, 189, 379, 489
 - cURL, 370, 372
 - emailing, 53, 56, 184
 - exception, 310–316
 - file open, 314
 - handling, 55, 57, 188, 310, 327
 - HTML, 464
 - JavaScript, 464, 527
 - live sites, 125
 - OOP, 234, 310, 327
 - parse, 32, 422
 - PEAR packages, 446, 461, 464
 - printing, 56
 - unable to open URL, 352
 - validation, 133
 - variable names and, 196
- escape() method, 499
- escaped characters, 189
- escapeshellarg() function, 125, 415
- escapeshellcmd() function, 125, 415
- escaping data, 468
- eval() function, 513, 517, 519
- event-based parsers, 548
- events
 - destroy, 397
 - form-related, 456

- JavaScript, 500
 - user, 386, 397
 - Excel class, 415
 - Excel spreadsheets, 405, 406–415
 - Exception class, 311, 317–327
 - Exception class methods, 311
 - Exception constructor, 318, 327
 - exception handling, 310, 312–316, 320
 - exceptions
 - catching, 310–316, 414
 - COM, 414
 - errors, 310–316
 - throwing, 310–316
 - vs. error handling, 310
 - exec() function, 415
 - execution tendencies, 444
 - Expat parser, 548, 549–551, 555
 - Expires header, 75, 79
 - extends statement, 266
 - Extensible Markup Language. *See* XML
 - Extensible Stylesheet Language
 - Transformation (XSLT), 555
 - extensions, 48, 61, 64, 424, 440
- ## F
- f flag, 428
 - f mode, 374
 - factory() method, 470–471, 474
 - fclose() function, 316, 352, 360, 383
 - fgetc() function, 442
 - fgetcsv() function, 352, 353, 442
 - fgets() function, 437
 - fields
 - comments, 141
 - name, 140, 141, 464
 - password, 465
 - required, 465, 484
 - textarea, 456
 - timestamp, 78
 - file element type, 456
 - file extensions, 48, 61, 64, 424, 440
 - file() function, 426
 - file open modes, 374
 - file pointer, 355
 - file upload prompt, 456
 - file_exists() function, 325
 - FileIO class, 326
 - filename rule, 462
 - filenames, 125, 462, 545
 - files
 - backup, 375–384
 - binary, 382
 - class definition, 261–262
 - closing, 316, 374
 - compressing, 374–384
 - configuration. *See* configuration files
 - CSS, 58, 190
 - database, 57, 187
 - FTP, 368, 372
 - help, 418, 420, 421, 475
 - included, 125
 - modification date, 423
 - moving, 125
 - names, 125, 462, 545
 - opening, 316, 348, 374, 380, 426
 - permissions, 314, 315, 348, 381
 - PHP, 199, 400–401
 - php.ini, 406, 424
 - read/write modes, 374–375
 - time modified, 378
 - Word, 404
 - XML. *See* XML documents
 - \$filter argument, 137
 - filter_has_var() function, 142
 - filtering
 - comments, 140–142
 - form data, 461–465
 - variables, 136–142
 - filter_input() function, 137
 - filter_list() function, 137
 - filters
 - array of, 142
 - listed, 137
 - PECL Filter, 136–142
 - filter_var() function, 137
 - filter_var_array() function, 142
 - final function, 279
 - Firefox browser, 493, 526, 527, 531, 565
 - flag variables, 202
 - floating-point numbers, 37, 38, 98, 438–442

- folders, 48, 54
- fopen() function
 - CLI scripts, 426
 - file compression, 374, 380, 383–384
 - opening files, 316, 348, 374, 426, 554
 - Web site access, 348
- fopen() method, 362
- fopen() mode, 316
- for loop, 73, 122, 519
- foreach loop, 362, 413, 427, 560
- foreign keys, 9, 175, 176, 182
- formatting items
 - characters, 37–38
 - with CSS classes, 550–554
 - numbers, 37–38
 - strings, 38
 - XML output, 550–554
- forms. *See also* HTML forms
 - credit card processing, 229–231
 - default values, 461
 - filtering data, 461–465
 - hidden input, 131
 - login functionality, 144–156
 - logout functionality, 93–95, 154
 - processing data in, 466–468
 - registration. *See* registration forms
 - rules, 461–467
 - validating, 126–135, 131, 461–465
- fragment value, 356
- Frameworks, 72
- fscanf() function, 42, 437, 438, 442
- fsockopen() function, 355
- fsockopen() method, 356–362
- FTP files, 368, 372
- FTP port, 355
- functions
 - ActiveXObject(), 492–494
 - addUser(), 149
 - array_sum(), 303
 - autoload(), 260–262, 306
 - autoloading, 306
 - base64_decode(), 157, 166
 - base64_encode(), 157, 162
 - benchmarking, 451–456
 - calculate(), 387–388, 392
 - call_this_(), 495
 - Character Type, 127
 - checkdate(), 126, 132
 - check_url(), 359, 362
 - check_urls(), 361
 - check_username(), 498, 502, 505
 - clear(), 396
 - close_session(), 94
 - com(), 404
 - commenting, 45
 - CONCAT_WS(), 107
 - count_to_100(), 25
 - cryptType, 151
 - ctype_alnum(), 131
 - ctype_digit(), 132
 - curl_errno(), 372
 - curl_error(), 372
 - curl_getinfo(), 372
 - curl_init(), 368, 370
 - curl_setopt(), 368
 - date(), 423, 567
 - destruct(), 256, 258
 - dl(), 391
 - do(), 289
 - echo(), 37, 398, 447–454, 475
 - error_log(), 56, 188
 - escapeshellarg(), 125, 415
 - escapeshellcmd(), 125, 415
 - eval(), 513, 517, 519
 - exec(), 415
 - fclose(), 316, 352, 360, 383
 - fgetc(), 442
 - fggetcsv(), 352, 353, 442
 - fggets(), 437
 - file(), 426
 - file_exists(), 325
 - filter_has_var(), 142
 - filter_input(), 137
 - filter_list(), 137
 - filter_var(), 137
 - filter_var_array(), 142
 - final, 279
 - fopen(). *See* fopen() function
 - fscanf(), 42, 437, 438, 442
 - fsockopen(), 355

- fwrite(), 314, 355, 380
- getAuthData(), 151, 156
- get_class(), 270
- gethostbyaddr(), 367
- gethostbyname(), 363
- gethostbyname1(), 367
- get_parent_class(), 270
- get_perimeter(), 246
- get_stores(), 516, 521
- grade_sort(), 28
- gzfile(), 375
- gzopen(), 374, 377, 380
- gzwrite(), 380
- handle_character_data(), 554
- handle_check(), 498, 499
- handle_open_element(), 553
- handle_stores(), 517, 518
- header(), 74–76
- htmlentities(), 372, 567
- include_once(), 241
- is_a(), 284
- isset(), 560
- is_subclass_of(), 284
- is_writable(), 378
- mail(), 124, 469–471
- make_list(), 23, 24, 35
- MCRYPT_DEV_RANDOM, 158
- mcrypt_enc_get_iv_size(), 158
- mcrypt_end_get_key_size(), 163
- mcrypt_module_open(), 158
- MCRYPT_RAND, 158
- MD5(), 149, 158–161, 160, 163
- mdecrypt_generic(), 164, 166
- my_error_handler(), 189
- mysqli_real_escape_string(), 88, 189, 468, 490
- names, 243
- name_sort(), 26
- number(), 228
- ob_gzhandler(), 80
- open(), 523
- parse_url(), 355–356
- passthru(), 415
- pclose(), 415
- performance, 452–455
- popen(), 415
- printf(), 37–38, 398, 441
- print_r(), 6, 93
- readgzfile(), 375
- recursive, 18–24
- references and, 30
- registerRule(), 465
- require_once(), 241, 243
- return_distance(), 114, 507
- rtrim(), 164
- scanf(), 42
- session, 84–90
- session.auto_start, 90
- session_cache_limit(), 80
- session_destroy(), 89
- session_save_path(), 82
- session_set_save_handler(), 84, 90, 91
- session_start(), 90, 93
- session_write_close(), 91, 94, 95
- set_display(), 391–392, 396
- setlocale(), 127
- setMarker(), 445–449
- set_number(), 392–393, 396
- set_operator(), 396
- set_time_limit(), 362, 406–407
- SHA(), 174, 485
- shell_exec(), 368, 415
- show_login_form(), 154
- show_stores(), 517–519
- signature, 280
- sleep(), 269
- sprintf(), 38, 39–42, 490
- srand(), 158
- stored, 110–115
- strcasecmp(), 6, 352
- strcmp(), 3
- strip_tags(), 125
- strtolower(), 553
- system(), 415
- trigger_error(), 189
- trim(), 462
- uasort(), 3, 6
- uksort(), 3
- UNIX_TIMESTAMP(), 78
- unserialize(), 343

usort(), 3, 6
 vprintf(), 42
 write_session(), 94
 xml_parse(), 549
 xml_parse_create(), 549, 554
 xml_parse_free(), 549
 xml_set_character_data_handler(), 549
 xml_set_element_handler(), 549
 fwrite() function, 314, 355, 380

G

garbage collection function, 89
 Garrett, Jesse James, 483
 general_widgets table, 173, 177–180
 get() method, 451, 470
 get_ methods, 250
 GET requests, 495, 526
 get_area() method, 248, 301, 304
 getAuth() method, 150
 getAuthData() function, 151, 156
 get_class() function, 270
 getCode() method, 311
 get_count() method, 298
 getFile() method, 311
 gethostbyaddr() function, 367
 gethostbyname() function, 363
 gethostbynameI() function, 367
 getMessage() method, 311, 315
 get_parent_class() function, 270
 get_perimeter() function, 246
 get_perimeter() method, 304
 getProfiling() method, 445
 getRecord() method, 364
 get_stores() function, 516, 521
 getTrace() method, 311
 getTraceAsString() method, 311
 GIMP (GNU Image Manipulation Program), 385
 GIMP Tool Kit (GTK+), 385–398
 global statement, 30
 global variables, 25, 55, 137, 391–394
 Gnope Web site, 385
 GNU Image Manipulation Program (GIMP), 385

Google, 530
 grade_sort() function, 28
 graphical calculator example, 386–398
 group element type, 456
 > entity, 536
 GTK+ (GIMP Tool Kit), 385–398
 GTK home page, 385
 GtkWindow() widget, 394
 gzfile() function, 375
 gzopen() function, 374, 377, 380
 gzwrite() function, 380

H

h mode, 374
 handle_character_data() function, 554
 handle_check() function, 498, 499
 handle_open_element() function, 553
 handle_stores() function, 517, 518
 header element type, 456
 header() function, 74–76
 header types, 74–75, 458
 headers() method, 470
 HelloWorld class, 237–244
 HelloWorld example, 237–244
 help files, 418, 420, 421, 475
 heredoc syntax, 31–36, 339, 445–451
 hexadecimal integer, 37
 hexadecimal numbers, 127
 hidden element type, 456
 hidden input, 131, 456
 hidden values, 135
 hiddenselect element type, 456
 hide() method, 397
 hierselect element type, 456
 hinting, type, 319
 home page, 69, 197–198
 host value, 356
 HTML
 Ajax applications, 500–503, 520–522
 clearing, 125
 entities, 539
 errors, 464
 vs. XML, 530
 HTML elements, 456, 457

HTML email, 469–479. *See also* email
HTML forms. *See also* forms
 Ajax applications, 501–503, 520–522
 creating, 134, 457–460, 501–503
 credit card processing, 229–231
 elements, 457
 hidden values in, 135
 HTML_QuickForm package, 456–468
 shopping cart, 337
HTML tags, 372, 496, 530
HTML templates, 58–61
HTML_AJAX package, 518
htmlentities() function, 372, 567
HTML_QuickForm class, 456, 457, 459
HTML_QuickForm package, 456–468
HTTP authentication, 157, 372
HTTP status codes, 356, 360, 361, 496
Huffman-only compression, 374

I

ID type, 542
IIS (Internet Information Server), 424
image element type, 456
image input, 456
images
 directory for, 48
 GD library, 135
 HTML email, 476–479
 as input, 456
IMAP port, 355
img tag, 476–478, 561
importing data, 98
IN clause, 220
included files, 125
include_once() function, 241
index page, 62–65, 197–198
indexes, 10, 99, 102, 519
infinite loop, 18
inheritance
 from classes, 266–270
 constructors, 271–275
 described, 264, 266
 destructors, 271, 274
 terminology, 265

inheritance theory, 274
innerHTML property, 496, 498, 499
InnoDB storage engine, 175, 176, 182
input
 hidden, 131, 456
 image, 456
 password, 456
 standard, 437–439, 442
 text, 456
 user, 438–442
instanceof keyword, 284
INT keyword, 102
integers, 37, 38
interfaces, 307
Internet Explorer browser, 74, 492, 526,
 527, 531
Internet service provider (ISP), 363
IP addresses, 356–367
IP geolocation, 363
is_a() function, 284
ISBN numbers, 182
is_empty() method, 333
ISP (Internet service provider), 363
isset() function, 560
is_subclass_of() function, 284
is_writable() function, 378
Iterate class, 451, 454

J

Java language, 494
JavaScript
 Ajax applications, 482, 492–499, 513–519
 asynchronous transactions and, 483
 clearing, 125
 errors/debugging, 464, 523–528
JavaScript alerts, 519, 525
JavaScript console, 527
JavaScript debugger, 527
JavaScript events, 500
JavaScript Object Notation (JSON), 506–521
JavaScript-enhanced text boxes, 456
joins, 100, 102
JSON (JavaScript Object Notation), 506–521

K

keys

- associating buttons with, 398
- foreign, 9, 175, 176, 182
- primary, 9, 83, 88, 149, 173–178

keywords, 110, 300

L

labels, 395, 456

Last-Modified header, 74–75, 79

LDAP port, 355

lettersonly rule, 462

libmcrypt, 157

libraries

- GD, 135
- MCrypt, 157
- PHP, 345
- SimpleXML, 557
- SPL, 345
- zlib, 374–376

libxml parser, 548

link element type, 456

links

- categories as, 194
- “Log Out,” 93–95
- product catalog, 205–211
- to shopping cart, 210

LOAD DATA INFILE query, 98

local variables, 110

login functionality, 144–156

logout functionality, 93–95, 154

logout() method, 154

loops

- for, 73, 122, 519
- foreach, 362, 413, 427, 560
- infinite, 18
- parsed files, 554, 559
- while, 121, 413, 511

< entity, 536

Lynx browser, 399

M

Mac OS X

- binary files and, 382

MySQL Administrator, 112–115

remote server connections, 422

running scripts, 424–425, 431

Terminal application, 400, 421

testing CLI installation on, 421

macros, 415

“magic” constants, 465

Magic Quotes, 189

mail. *See* email

Mail class, 469–471

mail() function, 124, 469–471

Mail_Mime class, 469, 471, 474

Mail_Mime package, 469–479

main() method, 397

main module, 66–69

main page, 62–65

main_quit() method, 397

make_list() function, 23, 24, 35

markup language, 530, 531

max-age directive, 75

maxfilesize rule, 462

maxlength rule, 462

MCrypt application, 157–167

MCrypt library, 157

MCRYPT_DEV_RANDOM function, 158

mcrypt_enc_get_iv_size() function, 158

mcrypt_end_get_key_size() function, 163

mcrypt_module_open() function, 158

MCRYPT RAND function, 158

MD5() function, 149, 158–161, 163

MD5() method, 154

mdcrypt_generic() function, 164, 166

member access, 281–288

META tags, 80

metadata, 171, 175

meta-properties, 45

methods

- abstract, 300–307
- access control, 281–288
- accessibility of, 281–288
- addAttachment(), 478
- addAttribute(), 556
- addBcc(), 470
- addCc(), 470
- addChild(), 556

- addHTMLImage(), 476–477, 479
- add_item(), 333–334, 344
- addRule(), 461–467
- addUser(), 145
- applyFilter(), 461–463, 468
- arguments, 235, 240, 300, 301
- asXML(), 561
- attach(), 395
- call(), 280
- checkAuth(), 145, 154
 - within classes, 235
- close(), 326
- connect_simple(), 386, 390, 398
- construct(), 251–252, 255, 271
- delete_item(), 335–336
- display(), 445, 466–468
- display_cart(), 346
- do(), 244
- encodeURIComponent(), 499
- escape(), 499
- Exception class, 311
- factory(), 470–471, 474
- fopen(), 362
- fsockopen(), 356–362
- get(), 451, 470
- get_area(), 248, 301, 304
- getAuth(), 150
- getCode(), 311
- get_count(), 298
- getFile(), 311
- getMessage(), 311, 315
- get_perimeter(), 304
- getProfiling(), 445
- getRecord(), 364
- getTrace(), 311
- getTraceAsString(), 311
- headers(), 470
- hide(), 397
- is_empty(), 333
- logout(), 154
- main(), 397
- main_quit(), 397
- MD5(), 154
- names, 243
- OOP, 277–281, 300–307
 - open(), 495, 513, 523
 - overloading, 280
 - overriding, 277–281
 - print_var(), 284, 285, 288
 - run(), 455
 - send(), 523
 - set_, 250
 - setAdvancedSecurity(), 156
 - setAuthData(), 156
 - set_default_size(), 394
 - set_size(), 245, 247, 248, 255
 - set_title(), 394
 - show(), 397
 - show_all(), 397
 - start(), 154
 - static, 295–299
 - testing, 455
 - timeElapsed(), 445, 449
 - toString(), 311, 318
 - update_item(), 333, 335, 345
 - visibility of, 281–288
- Microsoft Script Debugger, 527
- Microsoft.XMLHTTP object, 494
- mimetype rule, 462
- minlength rule, 462
- modularity, 49–73, 234
- month menu, 456
- Msxm12.XMLHTTP object, 494
- multidimensional arrays, 2–24
- must-revalidate directive, 75
- my_error_handler() function, 189
- MySQL
 - authentication, 145–149, 152
 - connecting to. *See* database connections
 - database access, 83
 - database creation, 10–11
 - distance calculation, 104–109
 - errors. *See* errors
 - importing data, 98
 - joins, 102
 - stored functions, 110–115
 - versions, 81, 183
- MySQL Administrator, 112, 113
- mysql client, 111
- MySQL GUI Tools package, 112

MySQL Improved extension, 14, 183, 489
 MySQL users, 145
`mysqli_real_escape_string()` function, 88,
 189, 468, 490

N

`\n` (newline), 124, 423, 427, 438
 name attribute, 288
 name element, 458
 name fields, 140, 464
`name_sort()` function, 26
 nested items

- conditionals, 310
- tasks, 19, 24
- to-do lists, 8–17
- XML elements, 532, 534

 .NET objects, 415
`Net_Geo` PEAR class, 363
`Net_GeoIP` PEAR class, 363
 networking, 347–372

- `cURL` utility, 368–372
- IP geolocation, 363–367
- PEAR and, 354
- sockets, working with, 355–362
- Web site access, 348–354

 network-related classes, 354
 new keyword, 240
 newline (`\n`), 124, 423, 427, 438
`NMTOKEN` type, 542
`NMTOKENS` type, 542
 no-cache directive, 75
 nodes, 499
 nonzero rule, 462
`nopunctuation` rule, 462
 normalization, 172, 173, 176
 NULL values, 99, 176, 510
`number()` function, 228
 numbers

- binary, 37–38
- character type functions, 127
- floating-point, 37–38, 98, 438–442
- formatting, 37–38
- hexidecimal, 127
- integers, 37–38
- ISBN, 182

SKU, 172, 178, 182
 type specifiers, 37
 numeric rule, 462
 numeric sorts, 2–7

O

`ob_gzhandler()` function, 80
 Object Browser, 405
 object type hinting, 319
 object-oriented programming. *See* OOP
 objects

- `ActiveXObject`, 492–494
- COM, 410
- creating, 240–243, 492
- deleting, 240, 242, 249, 259
- `Microsoft.XMLHTTP`, 494
- `Msxml2.XMLHTTP`, 494
- .NET, 415
- OOP, 234
- `$this` attribute, 244–250
- `XMLHttpRequest`, 492–496

 octal integer, 37
`onblur()` event, 500
`onchange()` event, 500
`onclick()` event, 521
`onfocus()` event, 500
`onload()` event, 500
`onreadystatechange` property, 495
`onreset()` event, 500
`onsubmit()` event, 500
 OOP (object-oriented programming),
 233–346

- abstract classes/methods, 300–307
- access control, 281–288
- advanced techniques, 263–308
- autoloading classes, 260–262
- cart class, 340–346
- constructors, 251–255
- creating objects, 240–243
- defining classes, 235–239
- design patterns, 235
- destructors, 256–259
- errors, 234, 310, 327
- exceptions. *See* exceptions
- inheritance. *See* inheritance

- object type hinting, 319
- overriding methods, 277–281
- performance and, 234
- pros and cons, 234
- real-world uses of, 309–346
- scope resolution (::) operator, 289–293
- shopping cart class, 328–339
- static class attributes, 294–299
- theories, advanced, 264–265
- theories, basic, 233–262
- `$this` attribute, 244–250
- versions, 233
- visibility, 281–288
- OOP classes
 - abstract, 300–307
 - autoloading, 260–262
 - considerations, 234
 - defining, 235–239
- `open()` function, 523
- “open” function, 84
- `open()` method, 495, 513, 523
- Open Source Web Design, 58, 190
- `$operator` variable, 392–393
- operators, 398
- optimizing joins, 102
- or symbol (`|`), 541
- Oracle database, 83
- `order_contents` table, 172, 173, 175
- `orders` table, 171, 172, 175
- “out” clause, 18
- overloading methods, 280
- overriding methods, 277–281

P

- padding specifier, 37
- page caching, 80
- parent classes, 264–270
- parent keyword, 289
- `parent_id` attribute, 9
- `parent_id` value, 10, 14
- parse errors, 32, 422
- parsers
 - described, 548
 - DOM, 548, 556–561
 - Expat, 548, 549–551, 555

- PHP, 548, 550–555
- SAX, 548, 561
- SimpleXML, 556–561
- `parse_url()` function, 355–356
- parsing XML, 548–561
- pass value, 356
- `passthru()` function, 415
- password element type, 456
- password elements, 460
- password field, 465
- password input, 456
- passwords. *See also* authentication
 - email, 131
 - encrypted, 149, 154
 - input, 456
 - rules, 465
 - validating, 131, 465
- path value, 356
- paths, 82, 359
- patterns, design, 235
- payments, credit card, 224–232
- PayPal, 530
- `#PCDATA` type, 541
- `pclose()` function, 415
- `pcntl` (process control) extension, 440
- PEAR (PHP Extension and Application Repository), 443–479
 - errors, 446, 461, 464
 - installing, 446
 - networking and, 354
- PEAR HTML_AJAX package, 518
- PEAR `Net_Geo` class, 363
- PEAR packages
 - Auth, 143–156
 - Benchmark, 444–456
 - Cache/Cache_Lite, 80
 - Console_Getargs, 436
 - Console_Getopt, 436
 - described, 443
 - HTML_AJAX, 518
 - HTML_QuickForm, 456–468
 - installing, 446
 - Mail_Mime, 469–479
 - Text_CAPTCHA, 135
- PECL (PHP Extension Code Library), 80

-
- PECL Filter, 136–142
 - PECL json package, 507
 - percent sign (%), 37, 42
 - performance
 - client/server, 80
 - functions, 452–455
 - OOP and, 234
 - permissions
 - crontab file, 401
 - directories, 378, 380–381
 - files, 314, 315, 348, 381
 - PHP
 - accessing Web sites with, 348–354
 - distance calculation, 103
 - errors. *See* error messages; errors
 - networking with, 347–372
 - parsing XML with, 548, 550–555
 - testing, 491, 524–527
 - using COM with, 404–415
 - versions, 81, 418
 - PHP CLI help file, 418, 420, 421
 - PHP CLI man page, 421
 - PHP Coding Standard, 46
 - php executable, 403, 430
 - .php extension, 48, 424, 565
 - PHP Extension and Application Repository.
 - See* PEAR
 - PHP Extension Code Library. *See* PECL
 - PHP files, 199, 400–401
 - PHP Library, 345
 - PHP scripts. *See* scripts
 - PHP tags, 48, 555
 - phpDocumentor, 46
 - PHP-GTK, 385–398, 427
 - PHP-GTK Web site, 385
 - phpinfo() scripts, 125, 157, 374
 - php.ini file, 406, 424
 - phpMyAdmin, 10–11, 97, 174
 - pipes, 415
 - plus sign (+), 541
 - polymorphism, 264–265
 - POP port, 355
 - popen() function, 415
 - port value, 356
 - ports, 355, 359
 - POST data, 370, 372
 - POST method, 135
 - POST requests, 523, 526
 - PostgreSQL database, 83
 - Pragma header, 75
 - precision specifier, 37
 - primary keys, 9, 83, 88, 149, 173–178
 - printf() function, 37–38, 398, 441
 - printing
 - array items, 21
 - categories as links, 194
 - error messages, 56
 - items in cart, 222
 - with printf(), 37–38, 398, 441
 - with print_r(), 6, 93
 - with print_var(), 284, 285, 288
 - with sprintf(), 38, 39–42, 490
 - with vprintf(), 42
 - print_r() function, 6, 93
 - print_var() method, 284, 285, 288
 - private directive, 75
 - process control (pcntl) extension, 440
 - product catalog, 205–211
 - product ID, 172, 208, 213–219, 344, 345
 - product page, 205–211
 - product tables, 172, 177, 178
 - products. *See also* widgets
 - adding to shopping cart, 217, 218, 333–334, 344
 - browsing by category, 199–204
 - calculating prices, 221
 - deleting in shopping cart, 219, 223, 335–336
 - displaying in shopping cart, 219–223
 - displaying online, 205–211
 - invalid ID, 211
 - names, 209
 - out-of-stock, 210
 - printing items in cart, 222
 - retrieving information about, 217
 - updating in shopping cart, 218–223, 333, 335, 345, 346
 - properties, 235, 281–288. *See also specific properties*
 - protected variable, 286

proxy servers, 74
proxy-revalidate directive, 75
public directive, 75
public variables, 285
punctuation, 127
PuTTY application, 422

Q

query results
 displaying horizontally, 116–122
 reporting on, 15
 sent to browser, 8
query value, 356
question mark (?), 541
QuickForms. *See* HTML_QuickForm
 package
" entity, 536
quotation marks (“), 284, 444, 445–455
quote marks (’), 444, 445–455

R

\r (carriage returns), 124
r mode, 374
r+ mode, 374
radio button, 456
radio element type, 456
ranglength rule, 462
Ray, Arpad, 518
read access, 374
“read” function, 84
readgzfile() function, 375
readyState property, 495, 496, 498, 499
Really Simple Syndication. *See* RSS
records. *See also* databases
 adding to database, 101, 178–181, 485–486
 backing up, 375–384
 displaying horizontally, 116–122
 retrieving, 379, 384, 413
Rectangle class, 246–250
Rectangle example, 244–250
recursive functions, 18–24
references, 30
regex rule, 462

register_globals, 124
registerRule() function, 465
registration forms
 creating with Ajax, 484–505
 creating with HTML_QuickForm, 456–468
 default form values, 461
 filtering data, 461–465
 processing data, 466–468
 validating (Ajax), 484, 488–489, 499
 validating (QuickForm), 128–131, 138–139
regular expressions, 124, 126, 131, 132
Remote Procedure Calls (RPC), 530
remote servers, 422
renderers, 460
Representational State Transfer (REST), 530
required rule, 462
require_once() function, 241, 243
reset button, 456
reset element type, 456
responseText attribute, 513, 517
responseText property, 495, 496, 525
responseXML property, 495
REST (Representational State Transfer), 530
return_distance() function, 114, 507
RETURNS clause, 111
Rico software, 518
Rijndael algorithm, 158
root element, 532, 534–535, 562
root folder, 54
routines, stored, 110–115
rows, 116–122, 395, 413
RPC (Remote Procedure Calls), 530
RSS (Really Simple Syndication), 562–568
RSS feeds, 529, 562–568
rtrim() function, 164
Ruby technology, 518
rules
 adding to form elements, 461–467
 email addresses, 462, 464, 465
 error messages, 464
 passwords, 465
 validation, 461–465
run() method, 455

S

- Safari browser, 527, 531
- SAJAX software, 518
- salutation element, 458
- SAX (Simple API for XML), 548
- SAX parsers, 548, 561
- scalar values, 111
- scanf() function, 42
- Scheduled Tasks (Windows), 402–403
- schema, 540, 546
- scheme value, 356
- scientific notation, 37
- scope resolution (::) operator, 289–293
- script.aculo.us, 518
- scripts. *See also* code
 - Ajax examples, 487–490, 507–512
 - benchmarking, 444–450
 - CLI. *See* CLI scripts
 - comments, 44–46
 - phpinfo(), 125, 157, 374
 - running (Mac OS X), 424–425, 431
 - running (Unix systems), 424–425, 431
 - running (Windows), 403, 424–425, 429–430
 - testing, 491, 524–527
 - time limits for, 362, 406–407
- search module, 70–73
- security, 123–167. *See also* authentication; passwords
 - basics, 124–125
 - captcha test, 135
 - database connections, 422
 - encryption, 149, 154, 157–167
 - frameworks and, 72
 - mail abuses, 124
 - MCrypt, 157–167
 - modularization and, 64, 65
 - PEAR Auth package, 143–156
 - session storage and, 82
 - setAdvancedSecurity() method, 156
 - validating forms, 126–135
- select element type, 456
- select menu, 456
- self keyword, 289
- semicolon (;), 111
- send() method, 523
- serialized arrays, 89
- server requests, 80, 482, 496
- servers, 373–415
 - Ajax and, 482, 496
 - Apache, 424
 - cron utility, 399–401
 - file compression, 374–384
 - IIS (Internet Information Server), 424
 - live vs. test, 53, 54, 187
 - local, 53
 - multiple, 82
 - PHP and, 373
 - PHP-GTK, 385–398
 - proxy, 74
 - remote, 422
 - running commands on, 125, 415
 - scheduling tasks (Mac OS X), 399, 400–401
 - scheduling tasks (Unix), 399–401
 - scheduling tasks (Windows), 402–403
 - SMTP, 469
 - using COM with PHP, 404–415
- server-side caching, 80
- server-side validation, 462, 466
- server-specific constants, 53–54, 187
- \$_SESSION array, 88, 89
- session caching, 80
- session data, 82–95, 125
- session directory, 82, 125
- session functions, 84–90
- session handlers, 91–95
- session ID, 83, 88
- session table, 82
- session.auto_start function, 90
- session_cache_limit() function, 80
- session_destroy() function, 89
- sessions
 - arguments, 84
 - caching, 80
 - closing, 94
 - deleting, 89
 - opening, 88
 - security, 82
 - storing in databases, 82–95

- sessions table, 83
- session_save_path() function, 82
- session_set_save_handler() function, 84, 90, 91
- session_start() function, 90, 93
- session_write_close() function, 91, 94, 95
- set_ method, 250
- setAdvancedSecurity() method, 156
- setAuthData() method, 156
- set_default_size() method, 394
- set_display() function, 391–392, 396
- setlocale() function, 127
- setMarker() function, 445–449
- set_number() function, 392–393, 396
- set_operator() function, 396
- set_size() method, 245, 247, 248, 255
- set_time_limit() function, 362, 406–407
- set_title() method, 394
- SGML (Standard Generalized Markup Language), 529
- SHA() function, 174, 485
- shebang line, 424–425
- shell scripting, 417
- shell_exec() function, 368, 415
- shopping cart
 - adding items to, 217, 218, 333–334, 344
 - calculating item prices, 221
 - deleting items in, 219, 223, 335–336
 - displaying contents, 219–223, 337–339, 346
 - HTML form for, 337–339
 - links to, 210
 - printing items in, 222
 - retrieving product information, 217
 - templates, 190, 191–196
 - updating items in, 218–223, 333, 335, 345, 346
- shopping cart application, 212–223
- shopping cart class, 328–339
- shopping cart page, 213–223
- show() method, 397
- show_all() method, 397
- show_login_form() function, 154
- show_stores() function, 517–519
- sign specifier (+/-), 37
- signals, 440
- signature, 280
- signed integer, 38
- Simple API for XML (SAX), 548, 561
- simple authentication, 144–150
- SimpleXML, 556–561
- sizes table, 172, 177, 179
- SKU numbers, 172, 178, 182
- slash (/), 189
- sleep() function, 269
- Smarty system, 58
- s-maxage directive, 75
- SMTP port, 355
- SMTP server, 469
- SOAP, 530
- sockets, 355–362
- sorting, 2–7, 26, 28
- specific_widgets table, 173, 178–181, 196
- SPL (Standard PHP Library), 345
- sprintf() function, 38, 39–42, 490
- SQL injection attacks, 125
- SQLite database, 83
- srand() function, 158
- SSH application, 422
- SSH port, 355
- SSL port, 355
- Standard Generalized Markup Language (SGML), 529
- standard input (STDIN) constant, 437–439, 442
- standard integer, 37
- Standard PHP Library (SPL), 345
- start() method, 154
- static class attributes, 294–299
- static element type, 456
- static keyword, 294
- static members, 294–299
- static methods, 295–299
- static statement, 25–29
- static variables, 25–29, 294
- status property, 495, 496, 499
- stdClass class, 239
- STDIN (standard input) constant, 437–439, 442
- stored functions, 110–115
- stored routines, 110–115

- stores table, 100–102
 - strcasecmp() function, 6, 352
 - strcmp() function, 3
 - strings
 - formatting, 38
 - heredoc syntax, 31–36
 - specifiers, 37
 - user input, 438
 - XML, 561
 - strip_tags() function, 125
 - strtolower() function, 553
 - subclass constructors, 272–275
 - subclasses, 265, 276, 288, 298
 - submit button, 456, 457, 460, 516, 520
 - submit element type, 456
 - superclass, 265
 - switch case, 64
 - switch conditional, 64–65, 238, 324
 - synchronous transactions, 483
 - system() function, 415
- ## T
- tables
 - addresses, 182
 - backing up, 375–384
 - categories, 172, 176–178, 193, 194
 - colors, 172, 176, 179
 - columns, 83, 98–100, 102, 413
 - compression, 374–384
 - creating with CREATE TABLE, 10, 151–152, 174–178
 - creating with GTK, 394–395
 - creating with HTML, 116, 457
 - customers, 171, 174, 182
 - general_widgets, 173, 177–180
 - joining, 102
 - metadata, 171, 175
 - order, 171, 172, 175
 - order_contents, 172, 173, 175
 - populating, 173, 178–180, 486
 - product, 172, 177, 178
 - retrieving, 379, 411
 - sessions, 83
 - sizes, 172, 177, 179
 - specific_widgets, 173, 178–181, 196
 - stores, 100–102
 - testing, 11
 - zip code, 96–99
 - task attribute, 9
 - task_id attribute, 9, 10
 - tasks
 - adding to arrays, 23
 - adding to database, 12–18
 - nested to-do lists, 8–17
 - recursion and, 18–24
 - scheduling (Mac OS X), 399–401
 - scheduling (Unix), 399–401
 - scheduling (Windows), 402–403
 - Telnet port, 355
 - templates
 - applications, 190–196
 - e-commerce example, 190–196
 - HTML, 58–61
 - shopping cart, 190, 191–196
 - Web pages, 58–61
 - temporary directory, 82
 - Terminal application, 400, 421
 - testing
 - Ajax applications, 504–505
 - captcha test, 135
 - CLI installation (Mac/Unix), 421
 - CLI installation (Windows), 419–420
 - code snippets, 422–423
 - methods, 455
 - PHP scripts, 491, 524–527
 - scripts, 491, 524–527
 - tables, 11
 - validate() test, 466
 - text blocks, 456
 - text boxes, 456
 - text element type, 456
 - text input, 456
 - textarea element type, 456
 - textarea field, 456
 - \$this attribute, 244–250
 - \$this variable, 245–246
 - throw syntax, 311
 - time
 - current, 378

elapsed, 445, 449
 script time limits, 362, 406–407
 time zones, 228
 timeElapsed() method, 445, 449
 Timer class, 445–450
 timers, 445–450
 timestamp, 76, 78, 83, 175, 423
 TINYINT keyword, 102
 to-do list, 8–17
 toString() method, 311, 318
 transactions
 Ajax, 482, 483, 494
 synchronous vs. asynchronous, 483
 tree-based parsers, 548, 559
 Triangle class, 302–307
 trigger_error() function, 189
 trim() function, 462
 try block, 311, 314, 318
 try statement, 310
 try...catch blocks, 310, 311, 315, 318, 410
 type hinting, 319
 type specifiers, 37

U

uasort() function, 3, 6
 uksort() function, 3
 Uniform Resource Indicators (URIs), 540
 Uniform Resource Locators. *See* URLs
 Unix servers, 399–401
 Unix-based systems
 binary files and, 382
 cron utility, 399–401
 remote server connections, 422
 running scripts, 424–425, 431
 Terminal application, 400, 421
 testing CLI installation on, 421
 UNIX_TIMESTAMP() function, 78
 unserialize() function, 343
 unsigned decimal integer, 37
 update_item() method, 333, 335, 345
 uploadedfile rule, 462
 URIs (Uniform Resource Indicators), 540
 URL links, 456
 URLs (Uniform Resource Locators)
 information passed via, 135

redirects, 348
 running applications through, 524
 unable to open, 352
 validating, 355–362
 user events, 386, 397
 user input, 438–442
 user value, 356
 usernames
 checking availability of, 485–505
 registered, 491
 users
 adding, 145, 149
 events, 386, 397
 finding location of, 363–367
 input, 438–442
 MySQL, 145
 usort() function, 3, 6
 UTF-8 characters, 499

V

validate() test, 466
 validation
 categories, 202
 client-side, 462, 464, 465–466
 conditionals, 202–204
 credit card, 224–232
 email addresses, 131, 462
 errors, 133
 form data (Ajax), 461–465
 form data (QuickForm), 128–131, 138–139
 passwords, 131, 465
 product ID, 208
 rules, 461–465
 server-side, 462, 466
 URL, 355–362
 usernames, 485–505
 var keyword, 493
 variable scope, 498
 variables
 checking values of, 525
 checking with JavaScript alerts, 525
 commands and, 125
 commenting, 45
 \$debug, 54
 declaring, 110

- filtering, 136–142
 - flag, 202
 - global, 25, 55, 137, 391–394
 - included files and, 125
 - initializing, 124
 - local, 110
 - names, 45, 110, 196
 - `$operator`, 392–393
 - passing by reference, 30
 - protected, 286
 - public, 285
 - static, 25–29, 294
 - `$this`, 245–246
 - vs. attributes, 236
 - `$window`, 394
 - Venkman debugger, 527
 - visibility, 234, 236, 265, 281–288
 - Visual Basic editor, 404, 405
 - `vprintf()` function, 42
- W**
- w mode, 374
 - w+ mode, 374
 - W3C (World Wide Web Consortium), 530
 - Web applications, 43–80. *See also* applications
 - caching and, 74–80
 - code style/structure, 47–48
 - configuration files, 50–57
 - content modules, 66–69
 - documenting code, 44–46
 - HTML templates, 58–61
 - index page, 62–65
 - main module, 66–69
 - main page, 62–65
 - modularization and, 49–73
 - search modules, 70–73
 - Web browsers
 - Ajax applications, 482–528
 - caching Web pages, 74–80
 - Firefox, 493, 526, 527, 531, 565
 - history feature, 528
 - Internet Explorer, 74, 492, 526, 527, 531
 - Lynx, 399
 - query results sent to, 8
 - RSS feeds, 562–568
 - Safari, 531
 - Wget, 399
 - XML and, 531
 - Web pages
 - Ajax applications, 482–528
 - caching in, 74–80
 - home page, 69, 197–198
 - HTML templates, 58–61
 - login functionality, 144–156
 - logout functionality, 93–95
 - opening, 355, 369–372
 - ports, 355
 - posting data to, 369–372
 - product page, 205–211
 - shopping cart page, 213–223
 - titles, 60
 - updates to, 482–483
 - Web port, 355
 - Web Services Description Language (WSDL), 530
 - Web sites. *See also specific Web sites*
 - accessing with PHP, 348–354
 - authentication system, 143–156
 - companion site to book, 170
 - distance calculations on, 96–109
 - e-commerce home page, 197–198
 - home page, 69, 197–198
 - HTML templates, 58–61
 - modularizing, 49–73
 - reading data from, 348–354
 - sessions, 82
 - structure, 48
 - Wget browser, 399
 - WHERE clause, 106
 - while loop, 121, 413, 511
 - white space, 44, 48, 533
 - white space characters, 127
 - widget ID, 343
 - widgets, 385–386. *See also* products
 - `WidgetShoppingCart` class, 329–339, 340–346
 - `$window` variable, 394
 - Windows Vista systems, 420
 - Windows XP systems, 410, 419–420
 - Windows-based systems
 - binary files and, 382

- Component Object Module (COM), 404–415
- MySQL Administrator, 112–113, 114
- remote server connections, 422
- running scripts, 403, 424–425, 429–430
- Scheduled Tasks, 402–403
- testing CLI installation on, 419–420
- Word documents, 404
- workbooks, Excel, 405, 406–415
- World Wide Web Consortium (W3C), 530
- write access, 374
- “write” function, 84
- write_session() function, 94
- WriteToFile class, 326
- WSDL (Web Services Description Language), 530

X

- x mode, 374
- XAJAX software, 518
- x-coordinates, 395, 396
- XHTML, 483, 529, 550
- XML (Extensible Markup Language), 529–568
 - Ajax and, 481, 483
 - attributes, 536–539
 - browsers and, 531
 - Document Type Definitions, 540–547
 - empty elements, 536–539
 - entities, 536–539
 - naming conventions, 533
 - overview, 529, 530–531
 - parsing with Expat, 548, 549–551, 555
 - parsing with PHP, 548, 550–555
 - parsing with SimpleXML, 556–561
 - purpose of, 530
 - RSS feeds, 562–568
 - syntax, 532–535
 - versions, 532
 - vs. HTML, 125
 - white space and, 533
 - XPath language, 561
- XML data, 531, 544, 555, 556
- XML documents
 - associating DTDs with, 540–547

- browser-viewable, 531
- comments in, 533
- creating, 531, 534–535
- examples, 532–535
- nested elements in, 534
- structure of, 532
- valid, 544
- well-formed, 544
- white space and, 533
- XML elements, 530, 536–539
- XML prolog, 532
- XML Schema, 546
- XML strings, 561
- XML tags, 48, 530–534, 555
- XMLHttpRequest object, 492–496, 513
- xml_parse() function, 549
- xml_parse_create() function, 549, 554
- xml_parse_free() function, 549
- xml_set_character_data_handler() function, 549
- xml_set_element_handler() function, 549
- XPath language, 561
- XSLT (Extensible Stylesheet Language Transformation), 555

Y

- Yahoo!, 530
- y-coordinates, 396
- year menu, 456
- Yeung, Anthony, 58

Z

- ZCTA (Zip Code Tabulation Areas), 96
- Zend, 72, 80
- Zend Framework, 354
- Zend Optimizer, 80
- zip code search example, 506–522
- zip code table, 96–99
- Zip Code Tabulation Areas (ZCTA), 96
- zip codes, 96–109
- zips database, 96–109, 507–521
- zlib library, 374–376
- zlib tool, 80, 374–376, 382