

# Index

## Symbols

- & (ampersand) special character, 402
- && (ampersands) pattern character, 161
- \* (asterisk) pattern character, 155, 156
- @ (at) comment, 146
- \ (backslash) character strings, 123 pattern character, 156
- (\n) (backslash n) special character, 111, 124
- \t (backslash t) special character, 124–125
- <br /> tag, 112
- ^ (caret) pattern character, 156
- { } (curly braces) special character, 108, 117
- / (divide) arithmetic operator, 121
- \$ (dollar sign) pattern character, 156 special character, 113, 115 variables, 237
- . (dot) character strings, 125 pattern character, 156
- .=n (dot equal sign n) special character, 125
- "" (double quotes), special character, 116, 124–125, 402
- = (equal sign) loops, 176 special character, 114
- == (equal signs) comparison operator, 152, 153, 254 loops, 176
- === (equal signs), comparison operator, 153, 254
- ! (exclamation point) comparison operator, 164
- !== (exclamation point equal signs) comparison operator, 153
- !<, <> comparison operator, 153
- / (forward slash) delimiter, 158
- // (forward slashes) comment, 149
- /\* comment, 148
- > (greater than) comparison operator, 153, 402
- >= (greater than or equal to) comparison operator, 153
- (hyphen) pattern character, 156
- < (less than) comparison operator, 153
- <= (less than or equal to) comparison operator, 153
- % (modulus) arithmetic operator, 121
- \* (multiply) arithmetic operator, 121
- () (parentheses) pattern character, 156, 161
- % (percent) special character, 273
- + (plus sign) arithmetic operator, 121 comparison operator, 156
- # (pound sign) comparison operator, 149
- ? (question mark) comparison operator, 156
- ;(semicolon) special character, 170
- ' (single quote) special character, 116, 123, 124–125, 402
- [] (square brackets) comparison operator, 156 special character, 128
- (subtraction sign) arithmetic operator, 121
- \_ \_ (underscores) method, 239
- || (vertical lines) pattern character, 156, 161

## A

- a/A, date format symbol, 140
- a mode, 216
- a+ mode, 216
- abstract class, 248–249
- Access Control List (ACL), 387
- Access Denied error message, 64–65
- accessing
  - catalog databases, 558–559
  - customer databases, 536–537
  - files, 216–218
  - IMAP mailboxes, 465–466
  - mBox mailboxes, 465–466
  - MySQL data, 270
  - properties using `$this`, 237
  - shopping cart databases, 578–579
- account management (MySQL)
  - adding accounts, 278–281
  - changing privileges, 282–284
  - identifying current accounts, 277–278
  - overview, 275
  - passwords, 280–281
  - removing accounts, 284–285

- accounts
  - adding MySQL, 278–281
  - creating with
    - phpMyAdmin, 279–280
  - creating with SQL queries, 278–279
  - identifying current
    - MySQL, 277–278
  - MySQL, 267–268
  - names (MySQL), 272–273
  - privileges (MySQL), 274–275
  - removing MySQL, 284–285
  - setting up MySQL, 275–285
- ACL (Access Control List), 387
- activating
  - MySQL support on
    - Linux/Mac, 40
  - MySQL support overview, 39
  - MySQL support on Windows, 40–42
- adding
  - comments to PHP script, 148–149
  - data from data files with
    - phpMyAdmin, 326–327
  - data from data files with SQL queries, 325–326
  - data to shopping cart database, 579
  - information to databases, 320–327
  - information to URL, 515, 516–517
  - methods, 237–239
  - MySQL accounts, 278–281
  - rows of data with
    - phpMyAdmin, 322–324
  - rows of data in SQL queries, 321
  - tables to databases with
    - phpMyAdmin, 314–315
  - tables to databases with SQL queries, 311–313
  - text to images, 455–457
  - variables to the URL, 516–517
- administrative software, 264–267
- administrator
  - responsibilities, 269–270
  - user, 411
- Advanced Encryption Standard (AES), 412
- AES\_DECRYPT()
  - encryption scheme, 412
- AES\_ENCRYPT()
  - encryption scheme, 412
- all-in-one installation kits, 24, 51–52, 76–77
- ALL privilege, 275
- Allwhois Web site, 12
- alpha channel, 454
- ALTER privilege, 275
- ALTER query, 261–262, 316–317
- ALTER TABLE query, 316–317
- & (ampersand) special character, 402
- && (ampersands) pattern character, 161
- and comparison keyword, 159
- antivirus
  - policy, 366–367
  - software, 404
- Apache
  - changing port number, 85
  - changing settings, 84–85
  - changing Web space location, 85
  - configuring, 84–85, 97
  - configuring on Linux and Mac, 33–34
  - configuring on Windows, 34–35
  - document root, 38
  - downloading from Web site, 75
  - getting information on
    - Linux/Unix/Mac, 83–84
  - getting information on Windows, 83
  - information, 83
  - installing on a Mac, 79
  - installing from source code
    - on Linux/Mac, 79–80
  - installing on Windows, 77–79
  - mailing list, 19
  - mod\_ssl, 415
  - obtaining, 74–77
  - obtaining for Linux, 76
  - obtaining for Mac, 76
  - obtaining for Windows, 75
  - overview, 17–18
  - restarting on Linux/Unix/Mac, 82
  - securing, 383–385
  - starting on Linux/Unix/Mac, 81–82
  - starting on Windows, 81
  - stopping on Linux/Unix/Mac, 82–83
  - stopping on Windows, 81
  - versions, 18, 74
  - Web server, 73
- apache\bin directory, 96
- application
  - architect, 365
  - developers, 365
  - script, 566–570
- arithmetic operations, 120–121
- arithmetic operators, 121–122
- arrays
  - \$\_FILES, 506–507
  - \$\_SESSION, 528
  - assort statement, 132
  - creating, 128–129
  - current(\$arrayname) statement, 134
  - defined, 119
  - elements, 128
  - end(\$arrayname) statement, 135
  - getting values from, 133–134
  - iteration/traversing, 134
  - key pairs, 128
  - manually walking through, 134–135
  - multidimensional, 137–138

- next (*\$arrayname*)
    - statement, 134
  - \$\_POST*, 402–403
  - previous (*\$arrayname*)
    - statement, 135
  - print\_r statement, 116–118, 129
  - reading files into, 220–221
  - registering long, 472
  - removing values from, 130–131
  - reset (*\$arrayname*)
    - statement, 135
  - sort statement, 131
  - sorting, 131–132
  - using *foreach* to walk through, 135–136
  - value pairs, 128
  - var\_dump* statement, 129
  - viewing, 129–130
  - walking through, 134–136
  - arsort (*\$arrayname*)
    - statement, 132
  - ASCII code, 153
  - asort statement, 132
  - asort (*\$arrayname*)
    - statement, 132
  - assigning
    - strings to variables, 123–124
    - values to PHP variables, 114
  - \* (asterisk) pattern
    - character, 155, 156
  - attributes, 230, 231, 297
  - authentication
    - defined, 373
    - digital identities, 378–380
    - image recognition, 376–378
    - passwords, 374–376
  - auto entry, 352
  - AUTO\_INCREMENT
    - definition, 311
  - AVG (*columnname*) SQL
    - format, 330
  - avoiding
    - DoS (Denial of Service) attacks on the filesystem, 404
    - infinite loops, 175–176
- 
- ## B
- 
- \ (backslash)
    - character strings, 123
    - pattern character, 156
  - backticks, 205–206
  - backup
    - databases with *mysqldump*, 286–287
    - databases with *phpMyAdmin*, 288–290
    - defined, 14
    - recovery, 367–369
  - basename (" /t1/do.txt ")
    - function, 199
  - BetterWhois Web site, 12
  - BIGINT MySQL data
    - type, 303
  - binary files
    - defined, 50, 75
    - Web site, 24
  - bindir-*DIR* PHP configure
    - option, 31
  - blank pages, 45
  - block
    - catch, 252–253
    - conditional, 108
    - defined, 108, 151
    - try, 252–253
  - book
    - conventions, 2–3
    - icons, 6
    - organization, 4–5
    - Web site, 5
  - Boolean
    - data type, 127
    - defined, 119
  - break statement, 167, 177
  - browsing
    - data with *phpMyAdmin*, 327–328
    - data with SQL queries, 327
    - information in databases, 327–328
  - building
    - catalog database, 558
    - customer database, 536
    - database, 308–316
    - file lists in directories, 202–203
    - if statements, 162–164
    - lists of check boxes, 488–490
    - lists of radio buttons, 487–488
    - login script, 545–553
    - login Web page, 537–545
    - for loops, 168–169
    - online catalog application script, 566–570
    - selection lists, 480–486
    - shopping cart database, 577–578
    - SQL queries, 260–261
    - built-in extensions, 423
    - built-in functions, 189
    - Burmeister, Mary, *HTML 4 For Dummies*, 469
- 
- ## C
- 
- CA (Certificate Authority), 413
  - “Can’t connect to” error message, 65
  - CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart), 377–378
  - ^ (caret) pattern character, 156

- cast, 120
- catalog. *See* online catalog
- catalog database
  - accessing, 558–559
  - building, 558
  - creating, 556–559
  - designing, 556–558
- catch block, 252–253
- CERT, 420
- Certificate Authority (CA), 413
- Certificate Signing Request (CSR), 414
- changing
  - control process, 369–371
  - MySQL account privileges, 282–284
  - port numbers (Apache), 85
  - settings (Apache), 84–85
  - Web space location (Apache), 85
- channels
  - alpha, 454
  - overview, 453–454
- character strings. *See also* strings
  - assigning to variables, 123–124
  - defined, 262
  - joining, 125–126
  - single and double quotes with strings, 124–125
  - storing, 126–127
- characters
  - converting HTML special, 401–402
  - data, 301–302
  - escaping, 123, 353–354
- CHAR (*length*) MySQL data type, 303
- check boxes, 488–490
- checking
  - for empty fields in forms, 491–497
- format of information, 497–502
- image formats, 450–451
- MySQL installation, 48–49
- MySQL support, 42
- variable content, 154
- child class, 232
- class
  - abstract, 248–249
  - child, 232
  - defined, 230
  - defining a, 235–245
  - getting information about, 255
  - object-oriented programming (OOP), 230–231
  - overview, 230–231
  - parent, 232
  - preventing changes, 251
  - using in scripts, 246–247
- class statement, 235
- Client does not support authentication protocol error message, 65
- client software, 263
- \_\_clone method, 253
- close () function, 465
- closing
  - files, 218
  - PHP sessions, 527
- code
  - ASCII, 153
  - display, 190, 473–476
  - distribution, 431–433
  - logic, 190, 473
  - product categories Web page, 584–586
  - product information Web page, 586–588
  - reusing, 191
  - shipping form Web page, 591–596
  - standards, 432
- summary Web page, 596–600
- using PEAR Packages in your own, 440
- writing for index pages, 562–564
- writing for products page, 564–566
- writing for shopping cart Web page, 584–600
- color indexes, 454–455
- color manipulation
  - alpha channel, 454
  - channels, 453–454
  - color indexes, 454–455
  - overview, 452–453
- columns\_priv table, 276
- combining, information from more than one table, 334–338
- comma-delimited files
  - creating, 223
  - defined, 324
  - reading, 223
- comma-delimited format exchanging data in, 222–223
- overview, 222
- comma-separated value (CSV). *See* comma-delimited file
- command line, installing PEAR Packages from, 437–439
- Command Prompt window
  - creating digital certificates, 414
  - database backup, 287
  - getting Apache information, 83
  - manual shutdown, 62
  - restoring databases, 291
  - sending SQL queries, 263
  - starting MySQL, 49, 61
  - testing MySQL, 63–64

- comments
    - adding to PHP script, 148–149
    - defined, 148
  - communicating
    - defined, 372
    - with MySQL, 260–267, 344–348
  - Community Server, MySQL
    - open source software, 49
  - community support (PHP), 432–433
  - company Web sites, 11
  - comparing
    - objects, 254–255
    - values, 152–154
  - comparisons, joining
    - multiple, 159–161
  - Completely Automated Public Tring test to tell Computers and Humans Apart (CAPTCHA), 377–378
  - complex statements, 108–109
  - Computer Browser
    - Properties dialog box, 386
  - concatenation, 125
  - conditional block, 108
  - conditional statements
    - defined, 151, 161
    - if statements, 161–165
    - switch statements, 165–167
  - conditions
    - defined, 152
    - setting up, 152–161
  - Configuration Wizard,
    - Windows MySQL, 53, 55–56
  - configuring
    - Apache, 84–85, 97
    - Apache on Linux and Mac, 33–34
    - Apache on Windows, 34–35
    - Development Environment, 95–97
    - GD extension, 449–451
    - IIS (Internet Information Server), 35–36
    - MySQL, 60–61, 97
    - PHP, 36–37, 96–97
    - PHP for MySQL support, 40
    - PHP to send e-mail, 460
    - Web Server for PHP, 33–35
    - Web Server on Windows, 34–36
  - confirmation Web page, 584
  - connect () function, 465
  - connecting to MySQL server, 345–347
  - connection
    - strings, 411
    - verification, 271
  - constructors, 242
  - continue statement, 177
  - Control Panel
    - checking MySQL installation, 48
    - controlling Windows server, 61
    - displaying error messages, 65
  - IIS (Internet Information Server), 35, 86
  - installing Apache, 77
  - installing PEAR packages, 438
  - securing Web root, 387
  - setting up MySQL support, 40
  - XAMPP, 91–95, 98
- control process, 369–371
  - controlling
    - data access, 271–275
    - MySQL Server on Linux/Mac, 63
    - MySQL Server on Windows, 61–62
  - conventions, book, 2–3
  - converting
    - HTML special characters, 401–402
    - mysqli functions to mysql functions, 354–355
  - cookies
    - defined, 522
    - overview, 415–416
    - PHP sessions without, 528–529
    - retrieving information, 523
    - saving information, 523
    - setting expiration time, 523–524
    - shopping cart, 573
    - storing information in, 522–526
  - copy statement, 200–201
  - copying
    - files, 200–201
    - objects, 253–254
  - COUNT (columnname) SQL format, 330
  - counting
    - affected rows, 353
    - rows returned by a query, 351–352
  - CREATE DATABASE query, 309
  - CREATE privilege, 275
  - CREATE query, 261–262, 311–313
  - CREATE TABLE query, 311–313
  - CREATE USER, 278–279
  - CREATE USER query, 278–279
  - createAttribute () function, 442
  - createCDATASection () function, 442
  - createElement () function, 442

`createProcessingInstruction()`  
 function, 442  
`createTextNode()`  
 function, 442  
 creating  
   accounts with  
     phpMyAdmin, 279–280  
   accounts with SQL  
     queries, 278–279  
   arrays, 128–129  
   catalog database, 556–559  
   Certificate Signing  
     Request (CSR), 414  
   comma-delimited files, 223  
   databases, 310–311  
   digital certificates, 414–415  
   directories, 201–202  
   empty databases with  
     phpMyAdmin, 310  
   empty databases with SQL  
     queries, 309  
   forms for uploading files,  
     505–509  
   functions, 179–180  
   a new database, 309–310  
   objects, 234  
   PHP variables, 114  
   relationships between  
     tables, 300–301  
   user database, 534–537  
   credit card handling,  
     572–573  
   cross-site scripting (XSS)  
     defined, 400  
     overview, 417–418  
   CSR (Certificate Signing  
     Request), 414  
   CSV (comma-separated  
     value). *See* comma-  
     delimited file  
   {} (curly braces) special  
     character, 108, 117  
   `current($arrayname)`  
     statement, 134

customer database  
   accessing, 536–537  
   building, 536  
   designing, 535–536  
 customer login, 572  
 CustomerOrder table, 575  
 CVS, installing PEAR via,  
   439–440

## D

d/D, date format symbol,  
   140  
 data. *See also* information  
   adding from data files with  
     phpMyAdmin, 326–327  
   adding from data files with  
     SQL queries, 325–326  
   adding rows with  
     phpMyAdmin, 322–324  
   adding rows in SQL  
     queries, 321  
   adding to shopping cart  
     database, 579  
   browsing with  
     phpMyAdmin, 327–328  
   browsing with SQL  
     queries, 327  
   character, 301–302  
   checking all form, 401  
   date, 302  
   default access to MySQL,  
     270  
   enumeration, 302  
   exchanging in comma-  
     delimited format,  
     222–223  
   exchanging in flat files,  
     221–222  
   exposing confidential, 362  
   loss, 362  
   numerical, 302  
   organization in database  
     design, 296–300  
   restoring, 290–293

  retrieving in a specific  
     order, 331  
   retrieving from specific  
     rows, 331–332, 331–334  
   selecting for database  
     design, 295–296  
   sending with Secure  
     Sockets Layer (SSL),  
     412–415  
   time, 302  
   transfer, 14  
   validation, 491  
 data access  
   account names, 272–273  
   account privileges,  
     274–275  
   controlling, 271–275  
   hostnames, 272–273  
   passwords, 273–274  
 data types  
   Boolean, 127  
   character strings, 123–127  
   floating-point numbers,  
     119–123  
   integers, 120–123  
   MySQL, 303  
   NULL, 127  
   overview, 119–120  
   storing, 301–303  
 database security  
   encrypting stored  
     passwords, 412  
   overview, 409–410  
   privileges, 411  
   selecting users, 410  
   storing connection  
     strings, 411  
 database structure  
   changing with  
     phpMyAdmin, 317–318  
   changing with SQL  
     queries, 316–317  
   overview, 316

- databases
  - adding information, 320–327
  - adding tables with phpMyAdmin, 314–315
  - adding tables with SQL queries, 311–313
  - backing up with `mysqldump`, 286–287
  - backing up with phpMyAdmin, 288–290
  - browsing data, 327–328
  - building, 308–316, 536, 558
  - building catalog, 558
  - building customer, 536
  - catalog, 556–559
  - creating empty with phpMyAdmin, 310
  - creating empty with SQL queries, 309
  - creating new, 309–310
  - creating user, 534–537
  - customer, 536–537
  - deleting with phpMyAdmin, 310–311
  - deleting with SQL queries, 310
  - designing, 295–308
  - designing a sample, 304–307
  - documenting your design, 307–308
  - MySQL security, 276
  - overview, 319
  - protecting MySQL, 267–268
  - removing information, 340–341
  - restoring with phpMyAdmin, 292–293
  - restoring using `mysql` client, 291–292
  - retrieving information, 328–338
  - securing, 409–412
  - selecting, 349
  - shopping cart, 577–579
  - structure, 260, 316–318
  - updating information, 339–340
  - user selection, 410
  - variables, 573
- date
  - data, 302
  - format symbols, 140
  - formatting, 139–140
- date function, 139–140
- DATE MySQL data type, 303
- DATETIME MySQL data type, 303
- db table, 276
- DECIMAL (*length*, *dec*) MySQL data type, 303
- DECODE () encryption scheme, 412
- DECRYPT () encryption scheme, 412
- default file, defined, 16
- DEFAULT *value* definition, 311
- define statement, 119
- defining, class, 235–245
- definitions, 311
- delete () function, 466
- DELETE privilege, 275
- DELETE query, 261–262, 340
- deleting
  - databases with phpMyAdmin, 310–311
  - databases with SQL queries, 310
  - files, 200–201
  - values from arrays, 130–131
- delimiters
  - defined, 158
  - using other, 223–225
- Denial of Service (DoS) attacks, avoiding on the filesystem, 404
- deprecated functions, 144
- DES\_DECRYPT () encryption scheme, 412
- DES\_ENCRYPT () encryption scheme, 412
- designing
  - advanced for loops, 169–171
  - catalog database, 556–558
  - customer database, 535–536
  - databases, 295–308
  - index page, 560
  - Login Application, 534
  - login Web page, 537–538
  - online catalog, 555–556
  - online catalog Web pages, 559–566
  - products page, 561–562
  - sample databases, 304–307
  - shopping cart databases, 574–577
  - shopping cart Web pages, 579–584
- destroying, objects, 255–256
- \_\_destruct method, 256
- development, setting up
  - local computer for, 17–19
- Development Environment
  - configuring Apache, 97
  - configuring MySQL, 97
  - configuring PHP, 96–97
  - integrated, 17
  - opening XAMPP Web page, 93–94
  - selecting, 16–17
  - testing PHP, 94–95
  - testing phpMyAdmin, 94
- dialog box
  - Computer Browser Properties, 386
  - Edit System Variable, 41
  - Environment Variables, 41

- Identity, 389–390
- Properties, 392
- Windows Features, 86
- differential, 368
- digital certificates
  - creating, 414–415
  - obtaining, 412–414
  - overview, 380
- digital identities
- digital certificates, 380, 412–415
- digital signatures, 379
- overview, 378–379
- digital signatures, 379
- directories
  - apache\bin, 96
  - building file lists, 202–203
  - creating, 201–202
  - defined, 198
  - FTP (File Transfer Protocol), 212
  - FTP (File Transfer Protocol) listings, 212
  - include, 195–196
  - directory handle, 203
  - dirname (" /t1/do.txt ")
    - function, 199
  - disable-libxml PHP
    - configure option, 31
  - disabling services, 385–386
  - disaster recovery, 367–369
  - disk space, 14
  - display code
    - defined, 473
    - script, 474–476
    - separating from logic code, 190
  - display\_errors = Off
    - setting, 396
  - display\_errors = On
    - setting, 145
  - displaying
    - account information from phpMyAdmin, 277–278
    - account information with SQL queries, 277
    - catalog Web pages, 566
    - dynamic HTML forms, 477–490
    - error messages, 65, 145–146
    - forms with two submit buttons, 503–504
    - login Web page, 544–545
    - selected messages, 145–146
    - static HTML forms, 470–476
    - values in text fields, 477–480
    - variable values, 116–118
    - variables with print\_r statements, 117–118
    - variables with var\_dump statements, 118
    - Web page content, 110–113
  - DISTINCT, 332, 334
  - / (divide) arithmetic operator, 121
  - DNS (domain name system), 10
  - Document Object Model. *See* DOM (Document Object Model)
  - document root, 38
  - DocumentRoot directive, 85
  - \$ (dollar sign)
    - pattern character, 156
    - special character, 113, 115
    - variables, 237
  - DOM (Document Object Model)
    - reading, 441–442
    - writing to the, 442–443
  - domain name
    - overview, 12
    - system (DNS), 10
    - Web hosting company, 14
  - DoS (Denial of Service)
    - attacks, avoiding on the filesystem, 404
  - . (dot)
    - character strings, 125
    - pattern character, 156
  - “” (double quotes) special character, 116, 124–125, 402
  - do . .while loops
    - defined, 167
    - using, 174–175
  - downloading
    - Apache from the Web site, 75
    - Apache from Web site, 75
    - files with FTP (File Transfer Protocol), 212–214
    - from the MySQL Web site, 50
    - PEAR Package Manager, 433–436
    - from the PHP Web site, 22–23
    - verifying files, 24–25, 52, 77
    - XAMPP, 88
  - DROP privilege, 275, 310–311
  - DROP query, 261–262
  - DROP USER query, 284
  - Dykes, Lucinda, *XML For Dummies*, 444
  - dynamic forms. *See* forms
  - dynamic HTML forms, displaying, 477–490
  - dynamic Web site, 469

---

## E

---

  - e-commerce
    - digital certificates, 380
    - programming applications, 409–419
    - security, 359–364, 372
    - SSL (Secure Socket Layer), 17
  - e-mail
    - basics, 459–460
    - configuring PHP to send, 460
    - queuing messages, 463–465
    - security, 462
    - sending with PHP, 459–465

- e-mail addresses
  - overview, 15
  - sanitizing, 402–403
- E\_ALL, 146
- echo statements
  - displaying content in Web pages, 110
  - displaying variables, 116–118
  - echoing HTML for a form, 470
  - using in PHP scripts, 106–108
  - using variables in, 116–117
- echoing links, 512
- E\_DEPRECATED, 144
- Edit System Variable dialog box, 41
- educational institution Web sites, 12–13
- E\_ERROR, 143, 146
- element, 128
- else, 162
- elseif, 162
- embedded scripting
  - language, 105
- empty(*\$varname*)
  - function, 154, 189
- enable-ftp PHP configure option, 31
- enable-magic-quotes
  - PHP configure option, 31
- ENCODE() encryption
  - scheme, 412
- ENCRYPT() encryption
  - scheme, 412
- encryption
  - hash functions, 382
  - one-way, 381
  - overview, 380–381
  - public key, 381–382
  - salt, 380
  - schemes, 412
  - stored passwords, 412
  - strength, 381
- end(*\$arrayname*)
  - statement, 135
- endingcondition, 168
- ENDSTRING, 126–127
- Enterprise Server, MySQL
  - open source software, 49
- entity, 297
- ENUM ("val1", "val2"...) MySQL
  - data type, 303
- enumeration, data, 302
- Environment Variables
  - dialog box, 41
- = (equal sign)
  - loops, 176
  - special character, 114
- == (equal signs)
  - comparison operator, 152, 153, 254
  - loops, 176
- === (equal signs)
  - comparison operator, 153, 254
- error messages
  - Access Denied, 64–65
  - Can't connect to, 65
  - Client does not support authentication protocol, 65
  - displaying, 65, 145–146
  - fatal, 142, 143, 144
  - handling, 397–401
  - handling with exceptions, 251–253
  - logging, 147–148
  - MySQL, 349–351
  - notice, 142
  - operating system
    - commands, 208–209
  - parse, 142, 142–143
  - phpMyAdmin, 71–72
  - strict, 142
  - suppressing single, 146
  - troubleshooting, 44, 64–66
  - turning off, 145
  - types of PHP, 142–144
  - warning, 142
- error\_log = *filename*
  - option, 396
- error\_log setting, 147
- error\_reporting =
  - setting, 145–146
- error\_reporting(E\_ALL) statement, 146
- error\_reporting(error\_reporting Setting) statement, 146
- escaping, characters, 123, 353–354
- E\_STRICT, 146
- E\_WARNING, 146
- example, patterns, 156–158
- ! (exclamation point)
  - comparison operator, 164
- exception, throwing an, 251
- exchanging
  - data in comma-delimited format, 222–223
  - data in flat files, 221–222
- exec function, 205, 207–208
- exec-prefix=EPREFIX
  - PHP configure option, 31
- expose\_php = off
  - option, 395
- exposing, confidential data, 362
- expressions
  - pattern matching with
    - regular, 155–159
  - regular (regexes), 155, 400
- expunge() function, 466
- Extensible Stylesheet Language
  - Transformation. *See* XSLT (Extensible Stylesheet Language Transformation)
- extension\_loaded()
  - function, 425
- extensions
  - Built-in, 423
  - checking loaded, 424–426
  - external, 424
  - GD, 449–457
  - loading, 426–427
  - mail, 459–466
  - Mail\_IMAP, 465–466

extensions (*continued*)  
 Mail\_Mbox, 466  
 PHP file, 16  
 PHP scripts, 104  
 types, 423–424  
 Zend engine, 423  
 external extensions, 424

## F

F, date format symbol, 140  
 F-Prot, 404  
 fatal errors  
   defined, 142  
   handling, 143  
 fault-tolerant, 359–360  
 fclose(\$fh) statement, 218  
 fgetcsv function, 223  
 fgets statement, 218  
 \$fh = fopen  
   ("filename",  
   "mode") statement, 216  
 field, 260  
 \$\_FILES array, 506–507  
 file extensions. *See*  
   extensions  
 file function, 220–221  
 file management  
   copying, 200–201  
   deleting, 200–201  
   information retrieval,  
     198–200  
   organizing, 201–204  
   renaming, 200–201  
 FILE privilege, 275  
 file system, 198  
 File Transfer Protocol. *See*  
   FTP (File Transfer  
   Protocol)  
 fileatime("stuff.  
 txt") function, 199  
 filectime("stuff.  
 txt") function, 199  
 file\_exists statement,  
   198  
 file\_get\_contents  
   function, 221  
 filegroup("stuff.  
 txt") function, 199  
 filetime("stuff.  
 txt") function, 199  
 fileowner("stuff.  
 txt") function, 199  
 files  
   accessing, 216–218  
   binary, 24, 50, 75  
   building lists in  
     directories, 202–203  
   closing, 218  
   comma-delimited, 223, 324  
   copying, 200–201  
   creating forms for  
     uploading, 505–509  
   default, 16  
   defined, 198  
   deleting, 200–201  
   downloading with FTP  
     (File Transfer  
     Protocol), 212–214  
   exchanging data in flat,  
     221–222  
   extensions, 16  
   flat, 215  
   forms for uploading,  
     505–509  
   getting information about,  
     198–200  
   include, 194–195  
   libmysql.dll, 40  
   log, 147–148  
   Mail\_Queue\_Config.  
     php, 464  
   managing, 198–204  
   modes for opening, 216  
   opening on another Web  
     site, 217  
   opening in read mode,  
     216–217  
   opening in write mode,  
     217  
   organizing, 201–204  
   php.ini, 96  
   PKG, 57–58  
   processing overview, 104  
   processing uploaded,  
     506–507  
   reading comma-  
     delimited, 223  
   reading from, 218–221  
   reading into arrays,  
     220–221  
   reading into strings, 221  
   reading piece by piece,  
     219–220  
   renaming, 200–201  
   tab-delimited, 224–225, 324  
   text, 215, 573  
   uploading, 403–408  
   uploading with FTP (File  
     Transfer Protocol),  
     212–214  
   uploads with FTP  
     functions, 405–406  
   validating, 404  
   verifying downloaded,  
     24–25  
   writing to, 218  
 filesize("stuff.txt")  
   function, 199  
 filesystem  
   avoiding DoS (Denial of  
     Service) attacks on  
     the, 404  
   maintenance, 406–408  
 filetype("stuff.txt")  
   function, 199  
 file\_uploads = On  
   setting, 403  
 fixed-length format  
   character data, 301  
 flat files  
   defined, 215  
   exchanging data in,  
     221–222

- float, 120
- floating-point numbers
  - arithmetic operators, 121–122
  - defined, 119
  - formatting numbers as dollar amounts, 122–123
  - performing arithmetic operations, 120–121
- folders, 198
- font types, 451
- footprint, 385
- `fopen()` function, 405
- // (forward slashes)
  - comment, 149
- / (forward slash) delimiter, 158
- `<form>` tag, 493
- for loops
  - building, 168–169
  - defined, 167
  - designing advanced, 169–171
  - nesting, 169
- foreach, using to walk through an array, 135–136
- format
  - checking image, 450–451
  - checking information, 497–502
  - comma-delimited, 222–223
  - date symbols, 140
  - defined, 140
- formatting
  - dates, 139–140
  - numbers as dollar amounts, 122–123
- forms
  - checking for empty fields, 491–497
  - creating, 505–509
  - creating for uploading files, 505–509
  - displaying dynamic HTML, 477–490
  - getting information from, 470–472
  - multiple submit buttons, 503–505
  - organizing scripts that display, 473–476
  - processing information from, 490–502, 490–505
  - static HTML, 469–476
  - using, 512–513
- FTP (File Transfer Protocol)
  - directory listings, 212
  - downloading files with, 212–214
  - functions, 214–215, 405–406
  - installing, 435–436
  - logging in to server, 211
  - overview, 210
  - uploading files with, 212–214
- FTP\_ASCII, 212
- `ftp_cdup($connect)` function, 214
- `ftp_chdir($connect, "directoryname")` function, 214
- `ftp_close($connect)` function, 214
- `ftp_close($connect)` statement, 213
- `ftp_connect` function, 211
- `ftp_connect("servername")` function, 214
- `ftp_delete($connect, "path/filename")` function, 214
- `ftp_exec($connect, "command")` function, 214
- `ftp_fget($connect, $fh, "data.txt", FTP_ASCII)` function, 214
- `ftp_fput($connect, "new.txt", $fh, FTP_ASCII)` function, 214
- `ftp_get` function, 212–214
- `ftp_get($connect, "d.txt", "sr.txt", FTP_ASCII)` function, 214
- `ftp_login` function, 211
- `ftp_login($connect, $userID, $password)` function, 214
- `ftp_mdtm($connect, "filename.txt")` function, 215
- `ftp_mkdir($connect, "directoryname")` function, 215
- `ftp_nlist` statement, 212
- `ftp_nlist($connect, "directoryname")` function, 215
- `ftp_put($connect, "d.txt", "sr.txt", FTP_ASCII)` function, 215
- `ftp_pwd($connect)` function, 215
- `ftp_rename($connect, "oldname", "newname")` function, 215
- `ftp_rmdir($connect, "directoryname")` function, 215
- `ftp_size($connect, "filename.txt")` function, 215
- `ftp_systype($connect)` function, 215
- functionality, shopping cart, 573–574
- functions (basics)
  - built-in, 189
  - creating, 179–180
  - date, 139–140
  - defined, 151
  - deprecated, 144
  - `empty($varname)`, 189
  - `isset($varname)`, 189
  - overview, 178–179
  - passing values to, 181–186
  - `print_r`, 117–118
  - returning values from, 186–188
  - using variables, 180–181
  - `var_dump`, 118, 120

- functions (communication)
  - mysqli\_affected\_rows, 353
  - mysqli\_connect, 345–346
  - mysqli\_multi\_query (\$cxn, \$query), 348
  - mysqli\_num\_rows, 351–352
  - mysqli\_query, 347
  - mysqli\_real\_escape\_string, 354
  - mysql\_select\_db, 349
- functions (email)
  - close(), 465
  - connect(), 465
  - delete(), 466
  - expunge(), 466
  - getBody(), 466
  - getHeaders(), 465
  - mail(), 460–461
  - messageCount(), 466
- functions (extensions)
  - extension\_loaded(), 425
  - get\_loaded\_extensions(), 424
  - include(), 426–427
  - php -m, 425
  - require(), 426–427
- functions (file management)
  - basename("/t1/do.txt"), 199
  - dirname("/t1/do.txt"), 199
  - file, 220–221
- file retrieval information, 199
  - fileatime("stuff.txt"), 199
  - filectime("stuff.txt"), 199
  - file\_get\_contents, 221
  - filegroup("stuff.txt"), 199
  - filemtime("stuff.txt"), 199
  - fileowner("stuff.txt"), 199
  - filesize("stuff.txt"), 199
  - filetype("stuff.txt"), 199
  - is\_dir("stuff.txt"), 199
  - is\_executable("do.txt"), 199
  - is\_file("stuff.txt"), 199
  - is\_readable("stuff.txt"), 199
  - is\_writable("stuff.txt"), 199
  - pathinfo(), 200
  - readdir, 203
  - rtrim, 219
- functions (FTP)
  - FTP (File Transfer Protocol) overview, 214–215
  - ftp\_cdup(\$connect), 214
  - ftp\_chdir(\$connect, "directoryname"), 214
  - ftp\_close(\$connect), 214
  - ftp\_connect, 211
  - ftp\_connect ("servername"), 214
  - ftp\_delete(\$connect, "path/filename"), 214
  - ftp\_exec(\$connect, "command"), 214
  - ftp\_fget(\$connect, \$fh, "data.txt", FTP\_ASCII), 214
  - ftp\_fput(\$connect, "new.txt", \$fh, FTP\_ASCII), 214
  - ftp\_get, 212–214
  - ftp\_get(\$connect, "d.txt", "sr.txt", FTP\_ASCII), 214
  - ftp\_login(\$connect, \$userID, \$password), 214
  - ftp\_mdtm(\$connect, "filename.txt"), 215
  - ftp\_mkdir(\$connect, "directoryname"), 215
  - ftp\_nlist(\$connect, "directoryname"), 215
  - ftp\_put(\$connect, "d.txt", "sr.txt", FTP\_ASCII), 215
  - ftp\_pwd(\$connect), 215
  - ftp\_rename(\$connect, "oldname", "newname"), 215
  - ftp\_rmdir(\$connect, "directoryname"), 215
  - ftp\_size(\$connect, "filename.txt"), 215
  - ftp\_systype(\$connect), 215
- functions (image manipulation)
  - ImageColorAt(), 455
  - ImagePSText(), 455–457
  - ImageString(), 455–457
  - ImageStringUp(), 455–457
  - ImageTTFText(), 455–457
- functions (operating system)
  - exec, 205, 207–208
  - fgetcsv, 223
  - mkdir, 201–202
  - passthru, 205, 208
  - sqlite\_query, 226
  - system, 205, 207
- functions (programming), get\_class, 255
- functions (script organization)
  - getStateCodes(), 191
  - getStateNames(), 191

is\_float(\$number), 154  
 is\_int(\$number), 154  
 organizing scripts with,  
   191–192  
 preg\_match, 158, 499  
**functions (security)**  
   fopen(), 405  
   hash, 382  
   htmlentities(),  
     400–402  
   is\_binary(), 404  
   rename(), 408  
   script\_tags(), 399, 400  
   session\_regenerate\_  
     id(), 417  
**functions (Web site)**  
   header(), 513  
   session, 514  
   session\_start, 527  
   setcookie, 523–524  
**functions (XML)**  
   createAttribute(), 442  
   createCDATASection(),  
     442  
   createElement(), 442  
   createProcessing  
     Instruction(), 442  
   createTextNode(), 442  
   getElementById(), 442  
   getElementsByTagName  
     (), 442  
 fwrite statement, 218

---

## G

---

g/G, date format symbol, 140  
**GD extension**  
   configuring, 449–451  
   image manipulations,  
     451–457  
   overview, 449  
 GET method, 471  
 getBody() function, 466  
 get\_class\_ function, 255  
 getElementById()  
   function, 442

getElementsByTagName  
   () function, 442  
 getHeaders() function,  
   465  
 get\_loaded\_extensions  
   () function, 424  
 getStateCodes()  
   function, 191  
 getStateNames()  
   function, 191  
 global variables, 180  
**Google**  
   Code, 431  
   Web site, 12  
**GRANT privilege**, 275,  
   282–283  
**granting privileges**, 411  
**> (greater than)**  
   comparison operator,  
   153  
**>= (greater than or equal  
 to) comparison  
 operator**, 153  
**GROUP BY**, 331

---

## H

---

h/H, date format symbol,  
   140  
**handling**  
   credit card, 572–573  
   error messages, 397–401  
   error messages with  
     exceptions, 251–253  
     parse errors, 142–143  
     warnings, 143–144  
**hash functions**, 382  
 header() function, 513  
**headers**, 513  
**help, software**, 19  
**heredoc statement**, 126  
**host, Web**, 10  
**host table**, 276  
**hostnames**  
   defined, 16, 271  
   MySQL, 272–273  
**htdocs**, 94, 195

**HTML**  
   converting special  
     characters, 401–402  
   source code, 111  
   special characters, 401–402  
   troubleshooting output, 45  
*HTML 4 For Dummies Quick  
 Reference* (Ray), 4  
*HTML 4 For Dummies* (Tittel  
 and Burmeister), 469  
**HTML forms**  
   passing information using,  
     515–516  
   static, 469–476  
 htmlentities() function,  
   400–402  
**HTTP header, relocating  
 users with a**, 513–515  
 httpd.conf file, 84  
**- (hyphen) pattern  
 character**, 156

---

## I

---

i, date format symbol, 140  
**icons, book**, 6  
**IDE (integrated  
 development  
 environment)**, 17  
**identifying current MySQL  
 accounts**, 277–278  
**Identity dialog box**, 389–390  
**if statements**  
   building, 162–164  
   components, 162  
   defined, 161  
   negating, 164–165  
   nesting, 165  
**IIS (Internet Information  
 Server)**  
   configuring, 35–36  
   Control Panel, 35, 86  
   defined, 18  
   installing, 86  
   securing, 385–395  
   Web server, 73

- image manipulations
  - adding text to images, 455–457
  - color manipulation, 452–455
  - resizing images, 452
- image recognition
  - accessibility, 377
  - implementing, 377–378
  - overview, 376–377
- ImageColorAllocate
  - Alpha() method, 454
- ImageColorAt() function, 455
- ImageCopyResampled(), 452
- ImageCopyResized(), 452
- ImagePSText() function, 455–457
- images
  - adding text to, 455–457
  - checking formats, 450–451
  - resizing, 452
- ImageString() function, 455–457
- ImageStringUp() function, 455–457
- ImageTTFText() function, 455–457
- IMAP (Internet Message Access Protocol)
  - accessing mailboxes, 465–466
  - defined, 459, 465
- implementing image recognition, 377–378
- include directories, 195–196
- include files
  - naming with .php extensions, 194–195
  - storing, 194–195
- include() function, 426–427
- include statements
  - types, 193
  - variables in, 193–194
- include\_once, 193
- increment, 168
- index page
  - designing, 560
  - writing code for, 562–564
- indexes
  - color, 454–455
  - defined, 453
- infinite loops, 175–176
- infodir=DIR PHP
  - configure option, 31
- information. *See also* data
  - adding to databases, 320–327
  - adding to URL, 515, 516–517
  - browsing in databases, 327–328
  - checking format of, 497–502
  - class, 255
  - combining from more than one table, 334–338
  - getting from forms, 470–472
  - making available to all pages on Web site, 522–532
  - passing between Web pages, 515–522
  - processing from forms, 490–505
  - removing from databases, 340–341
  - removing with phpMyAdmin, 341
  - removing with SQL queries, 340
  - retrieving in cookies, 523
  - retrieving from databases, 328–338
  - retrieving file, 198–200
  - retrieving specific, 329–331
  - saving in cookies, 523
  - storing in cookies, 522–526
  - updating in databases, 339–340
  - updating with phpMyAdmin, 339–340
  - updating with SQL queries, 339
- inheritance, 232
- INSERT privilege, 275
- INSERT query, 261–262, 321
- inserting tabs, 124–125
- installing
  - Apache on a Mac, 79
  - Apache from source code on Linux/Mac, 79–80
  - Apache on Windows, 77–79
  - FTP (File Transfer Protocol), 435–436
  - IIS (Internet Information Server), 86
  - MySQL, 18, 52–60
  - MySQL GUI Administration Programs, 66
  - MySQL on Linux from an RPM file, 57
  - MySQL on Mac from a PKG file, 57–58
  - MySQL from source files, 58–60
  - MySQL on Windows, 52–56
  - options for Unix/Linux/Mac, 31–32
  - PEAR Package Manager via Web front end, 433–435
  - PEAR Packages from the command line, 437–439
  - PEAR via CVS, 439–440
  - PHP on Mac OS X, 28–30
  - PHP overview, 18–19, 25–26
  - PHP on Unix and Linux, 26–28
  - PHP on Windows, 32–33
  - phpMyAdmin, 67–69
  - Web servers, 17–18
  - XAMPP, 88–91
- instantiation, 230
- integers
  - arithmetic operators, 121–122
  - defined, 119

formatting numbers as  
 dollar amounts, 122–123  
 performing arithmetic  
 operations, 120–121  
 integrated development  
 environment (IDE), 17  
 interfaces, 249–251  
 Internet Information Server.  
*See* IIS (Internet  
 Information Server)  
 Internet Protocol (IP), 272  
 INT (*length*) MySQL data  
 type, 303  
 INT (*length*) UNSIGNED  
 MySQL data type, 303  
 IP address  
 connecting to FTP (File  
 Transfer Protocol)  
 server, 211  
 overview, 12  
 IP (Internet Protocol), 272  
 is\_array(*\$var2*), 154  
 is\_binary() function, 404  
 is\_dir("stuff.txt")  
 function, 199  
 is\_executable("do.  
 txt") function, 199  
 is\_file("stuff.txt")  
 function, 199  
 is\_float(*\$number*), 154  
 is\_int(*\$number*), 154  
 is\_null(*\$var1*), 154  
 is\_numeric(*\$string*),  
 154  
 is\_readable("stuff.  
 txt") function, 199  
 isset(*\$varname*)  
 function, 154, 189  
 is\_string(*\$string*), 154  
 is\_writable("stuff.  
 txt") function, 199  
 iteration, 134

---

## J

j, date format symbol, 140  
 JOIN, 334, 336–338

joining  
 character strings, 125–126  
 multiple comparisons,  
 159–161

---

## K

key pairs, 128  
 krsort(*\$arrayname*)  
 statement, 132  
 ksort(*\$arrayname*)  
 statement, 132

---

## L

l, date format symbol, 140  
 < (less than) comparison  
 operator, 153, 402  
 <= (less than or equal to)  
 comparison operator,  
 153  
 libmysql.dll file, 40  
 LIMIT keyword, 331, 334  
 links, echoing, 512  
 Linux  
 activating MySQL  
 support, 40  
 checking MySQL  
 installation, 48–49  
 checking PHP  
 installation, 22  
 configuring Apache, 33–34  
 configuring PHP, 36  
 controlling MySQL  
 Server, 63  
 controlling MySQL Server  
 on, 63  
 getting Apache  
 information, 83–84  
 installing Apache from  
 source code, 79–80  
 installing MySQL from an  
 RPM file, 57  
 installing PHP on, 26–28  
 obtaining Apache for, 76  
 obtaining MySQL for, 50–51  
 obtaining PHP for, 23

PHP installation options,  
 31–32  
 restarting Apache, 82  
 starting Apache, 81–82  
 stopping Apache, 82–83  
 troubleshooting error  
 messages, 44

Listen directive, 85  
 listing examples  
 Basic FTP Functions,  
 405–406  
 Building a Date Selection  
 List, 484–485  
 Building a List of Check  
 Boxes, 489–490  
 Building a List of Radio  
 Buttons, 487–488  
 Building a Selection List,  
 482–483  
 Checking for Blank Fields,  
 495–497  
 Checking for Invalid  
 Formats in Form Fields,  
 500–502  
 Displaying a Form with  
 Two Submit Buttons,  
 503–504  
 Displaying an HTML Form  
 with Information,  
 478–480  
 Displays a Form with a  
 Hidden Field, 493–495  
 File that Defines the  
 Product Categories  
 Web Page, 584–585  
 File that Defines the  
 Product Information  
 Web Page, 586–587  
 File that Defines the  
 Shipping Form, 591–596  
 File that defines the  
 Shopping Cart Web  
 Pages, 588–591  
 File that Defines the  
 Summary Page,  
 596–600

- listing examples (*continued*)
  - File that Defines Two Side-by-Side HTML Forms, 539–544
  - File that Displays the File Upload Form, 507–508
  - File that Displays the Index Page, 562–564
  - File that Displays a Login Form, 518
  - File that Displays the Product Page, 564–566
  - Hello World HTML script, 106
  - Hello World PHP Script, 106–107
  - Login Application Code, 546–553
  - Login Script that Stores Information in Cookies, 524–525
  - Login Script that Stores Information in Sessions, 531
  - Mail\_Queue\_Config.php file, 464
  - Online Catalog Application Script, 567–570
  - Processing Two Submit Buttons, 504–505
  - Sample Code for a MIME-Encoded e-mail, 462–463
  - Sample XML Schema Document, 443–444
  - Script to Create an Image Gallery, 203–204
  - Script to Download Files via FTP, 213–214
  - Script for Sending Queued Messages, 464–465
  - Script that Contains a Class for a Form Object, 244–245
  - Script that Converts a CSV file into a Tab-Delimited File, 224–225
  - Script that Creates a Form, 246–247
  - Script That Defines a Form, 474–475
  - Script that Displays all the Fields from a Form, 472–473
  - Script That Displays a Form, 474
  - Script that Displays a Form with No Fields, 512–513
  - Script that Displays and Processes the Login Form, 519–521
  - Script that Gets Information from Cookies, 525–526
  - Script that Gets Information from Sessions, 531–532
  - Script that Gets Information from the URL, 521–522
  - Script that Manages the Shopping Cart, 607–609
  - Script that Processes the Order, 610–616
  - Script that Provides product Information, 602–606
  - SQL Query for Creating a Table, 312
  - Uploading a File with a POST Form, 508–509
- lists
  - building check box, 488–490
  - building radio button, 487–488
  - building selection, 480–486
- LOAD query, 261–262, 325–326
- loading, extensions, 426–427
- local variables, 180
- log file, specifying, 147–148
- log\_errors =on option, 396
- log\_errors setting, 147
- logging
  - error messages, 147–148
  - in to FTP (File Transfer Protocol) servers, 211
- logic code
  - defined, 473
  - separating from display code, 190
- Login Application
  - building login script, 545–553
  - building login Web page, 537–545
  - creating user database, 534–537
  - designing, 534
  - overview, 533
  - protecting Web pages, 553–554
  - that stores information in cookies, 524–526
  - that stores information in a session, 530–532
  - login script, building, 545–553
  - login Web page
    - building, 537–545
    - designing, 537–538
    - displaying, 544–545
    - listings, 518–521
    - overview, 517
    - writing code, 538–544
- loops
  - avoiding infinite, 175–176
  - breaking out of, 177–178
  - building with for, 168–169
  - defined, 151
  - do..while, 167, 174–175
  - for, 168–171
  - overview, 167
  - while, 203
- loss, data, 362

## M

- M/m, date format symbol, 140
- Mac
- activating MySQL support, 40
  - checking MySQL installation, 48–49
  - checking PHP installation, 22
  - configuring Apache, 33–34
  - configuring PHP, 36
  - controlling MySQL Server, 63
  - getting Apache information, 83–84
  - installing Apache, 79
  - installing Apache from source code, 79–80
  - installing MySQL from a PKG file, 57–58
  - installing PHP, 28–30
  - obtaining Apache for, 76
  - obtaining PHP for, 24
  - PHP installation options, 31–32
  - restarting Apache, 82
  - starting Apache, 81–82
  - stopping Apache, 82–83
  - troubleshooting error messages, 44
- mail, extensions, 459–466
- mail() function, 460–461
- mailboxes, accessing IMAP and mBox, 465–466
- Mail\_IMAP extension, 465–466
- mailing lists, 19
- Mail\_Mbox extension, 466
- Mail\_Mime, 462
- Mail\_Queue Package, 463–464
- Mail\_Queue\_Config.php file, 464–465
- MAMP, 24, 52, 77
- managing, files, 198–204
- mandir=DIR PHP configure option, 31
- master class, 232
- MAX(columnname) SQL format, 330
- mBox
- accessing mailboxes, 465–466
  - defined, 459, 465
- MD5() encryption scheme, 412
- MD5 signature checkers, 25, 77
- messageCount() function, 466
- messages
- displaying selected, 145–146
  - error, 44, 64–65, 71–72, 142–148
  - queuing, 463–465
- methods
- adding, 237–239
  - \_\_clone, 253
  - \_\_destruct, 256
  - GET, 471
  - ImageColorAllocate Alpha(), 454
  - object-oriented programming (OOP), 231–232
  - overview, 231–232
  - POST, 471, 490
  - preventing changes, 251
  - public/private, 240–242
  - schemaValidate(), 444–445
  - selecting foreach object, 233–234
- MIN(columnname) SQL format, 330
- mkdir function, 201–202
- % (modulus) arithmetic operator, 121
- modes, for opening files, 216
- ModSecurity, 384–385
- mod\_ssl, Apache, 415
- monitor, 263
- \* (multiply) arithmetic operator, 121
- multidimensional arrays, 137–138
- multiple comparisons, joining, 159–161
- multiple inheritance, 232
- MySQL
- account management, 275–285
  - accounts, 16, 267–268
  - adding accounts, 278–281
  - Administrator, 66
  - administrator responsibilities, 269–270
  - advantages, 9
  - building SQL queries, 260–261
  - checking installation, 48–49
  - communicating with, 260–267, 344–348
  - Components Wizard, 86
  - Configuration Wizard, 53, 55–56
  - configuring, 60–61, 97
  - data access, 270–275
  - data types, 303
  - data types Web site, 303
  - database access, 15
  - database backup, 285–290
  - defined, 1
  - error log, 66
  - errors, 349–351
  - GUI Administration Programs, 66
  - Improved, 354
  - installing on Linux from an RPM file, 57
  - installing on Mac from a PKG file, 57–58
  - installing overview, 18
  - installing from source files, 58–60
  - installing on Windows, 52–56
  - mailing list, 19
  - mysql client, 263–264
  - obtaining, 49–52

MySQL (*continue*)  
 obtaining for Mac, 51  
 online manual, 274  
 overview, 259–260  
 permissions, 268  
 PHP functions that  
   communicate with, 344  
   PHP working with, 343–344  
   protecting databases,  
     267–268  
 Query Browser, 66  
 removing accounts,  
   284–285  
 reserved words Web  
   site, 299  
 restoring data, 290–293  
 security database, 276  
 sending SQL queries,  
   262–267  
 setting up accounts,  
   275–285  
 software, 49  
 starting, 49, 61  
 testing, 63–64  
 troubleshooting, 64–66  
 troubleshooting functions,  
   44–45  
 updates, 19–20  
 upgrade information, 293  
 upgrading, 293–294  
 versions, 50, 293–294  
 Web site, 20  
 Web site downloading, 50  
 mysql client  
   restoring databases using,  
     291–292  
   sending SQL queries,  
     263–264  
 mysql function syntax, 355  
 MySQL Server  
   connecting to the, 345–347  
   controlling on Linux/  
     Mac, 63  
   controlling on Windows,  
     61–62  
   manual shutdown, 62

MySQL support  
 activating, 39–42  
 checking, 42  
 setting up files, 40–42  
 mysqldump, backing up  
   databases with,  
     286–287  
 mysqli function syntax, 355  
 mysqli\_affected\_rows  
   function, 353  
 mysqli\_close  
   (\$connectionname),  
   346–347  
 mysqli\_connect function,  
   345–346  
 mysqli\_multi\_query  
   (\$cxn, \$query)  
   function, 348  
 mysqli\_num\_rows  
   function, 351–352  
 mysqli\_query function,  
   347  
 mysqli\_real\_escape\_  
   string function, 354  
 mysql\_select\_db  
   function, 349

## N

n, date format symbol, 140  
 naming  
   include files with .php  
   extensions, 194–195  
   MySQL accounts, 272–273  
   PHP variables, 113–114  
 navigating Web sites,  
   511–515  
 negating, if statement,  
   164–165  
 nesting  
   if statement, 165  
   for loops, 169  
 Netbios, 385  
 NetCraft, 17, 103, 105, 383  
 network administrators, 365

next (\$arrayname)  
   statement, 134  
 NOT NULL definition, 311  
 notices, 142, 144  
 NULL  
   data type, 127  
   defined, 119  
 number\_f, 122–123  
 numerical data, 302

## O

object-oriented  
 programming (OOP)  
 abstract class, 248–249  
 classes, 230–231  
 comparing objects,  
   254–255  
 copying objects, 253–254  
 defined, 229  
 defining a class, 235–245  
 destroying objects,  
   255–256  
 developing object-  
   oriented scripts,  
   232–234  
 getting information about  
   objects and classes,  
   255  
 handling errors with  
   exceptions, 251–253  
 inheritance, 232  
 interfaces, 249–251  
 methods, 231–232  
 objects, 230–231  
 overview, 229–230  
 preventing changes to a  
   class or method, 251  
 properties, 231  
 using a class in a script,  
   246–247  
 object-oriented script  
   choosing objects, 233  
   creating an object, 234  
   overview, 232–233  
   selecting methods, 233–234

- selecting properties, 233–234
  - objects
    - choosing, 233
    - comparing, 254–255
    - copying, 253–254
    - creating, 234
    - defined, 119, 297
    - destroying, 255–256
    - getting information about, 255
    - object-oriented programming (OOP), 230–231
    - overview, 230–231
  - obtaining
    - all-in-one installation kits, 24, 51–52, 76–77
    - Apache information, 83
    - Apache for Linux, 76
    - Apache for Mac, 76
    - Apache for Windows, 75
    - digital certificates, 412–414
    - MySQL for Linux/Unix, 50–51
    - MySQL for Mac, 51
    - MySQL for Windows, 50
    - PHP for Linux, 23
    - PHP for the Mac OS, 24
    - PHP for Windows, 23
    - phpMyAdmin, 67
    - values from arrays, 133–134
    - XAMPP, 88
  - OCR (optical character recognition), 377
  - one-way encryption, 381
  - online catalog
    - building application script, 566–570
    - building catalog Web pages, 559–566
    - creating catalog database, 556–559
    - designing, 555–556
    - designing Web pages, 559–566
    - displaying Web pages, 566
    - overview, 555
  - open source software, 19–20
  - open\_basedir = directory option, 395
  - opendir statement, 202–203
  - opening
    - files on another Web site, 217
    - files in read mode, 216–217
    - files in write mode, 217
    - PHP sessions, 527
    - XAMPP Web page, 93–94
  - operating system
    - commands
      - backticks, 205–206
      - error messages, 208–209
      - exec function, 205, 207–208
      - overview, 204–205
      - passthru function, 205, 208
      - security issues, 209–210
      - system function, 205, 207
    - operations, arithmetic, 120–121
    - operators, arithmetic, 121–122
    - optical character recognition (OCR), 377
    - options
      - error\_log = filename, 396
      - expose\_php = off, 395
      - register\_globals = off, 395
  - or, 159
  - order
    - processing, 600–601
    - shopping cart scripts, 609–616
  - ORDER BY, 331
  - OrderItem table, 575–576
  - organization
    - book, 4–5
    - database design data, 296–300
  - organizing
    - files, 201–204
    - scripts, 189–196
    - scripts with functions, 191–192
    - scripts with include files, 192–196
    - scripts that display forms, 473–476
  - outputitem, 110
  - OWASP, 420
- 
- p***
- 
- package maintenance, 431–433
  - () (parentheses) pattern character, 156, 161
  - parent class, 232
  - parse errors
    - defined, 142
    - handling, 142–143
  - passthru function, 205, 208
  - passwords
    - account management (MySQL), 280–281
    - authentication, 374–376
    - changing with phpMyAdmin, 280–281
    - changing with SQL queries, 280
    - data access, 273–274
    - encrypting stored, 412
    - guessed, 375–376
    - lost, 374
    - MySQL, 273–274
    - overview, 374
    - stolen, 375–376
    - storing, 376, 411–412

- patch management policy, 366–367
- pathinfo() function, 200
- patterns
  - example, 156–158
  - matching with PHP
    - functions, 158–159
    - special characters, 155–156
- PEAR (PHP Extension and Application Repository)
  - code distribution, 431–433
  - downloading/installing
    - PEAR Package Manager, 433–436
  - FTP installation, 435–436
  - installing a PEAR Package
    - from the command line, 437–439
  - installing a PEAR Package via CVS, 439–440
  - library, 430–431
  - mailing lists Web site, 433
  - Mail\_Mime package, 462–463
  - overview, 429–430
  - package maintenance, 431–433
  - using a PEAR Package in your own code, 440
  - Validate class, 419
- PECL (PHP Extension Community Library), 432
- % (percent) special character, 273
- performing, arithmetic operations, 120–121
- permissions, 268
- phishing, 380, 400
- PHP
  - advantages, 9
  - applications secured with SuExec, 383–384
  - community support, 432–433
  - configure options, 31–32
  - configuring, 36–37, 96–97
    - configuring for MySQL support, 40
    - configuring to send e-mail, 460
    - configuring Web Server for, 33–36
    - constants, 118–119
    - defined, 1, 103
    - displaying error messages, 145–146
    - file extensions, 16
    - file processing, 104
    - functions that
      - communicate with MySQL, 344
    - how it works, 103–105
    - installing on Mac OS X, 28–30
    - installing overview, 18–19
    - installing on Unix and Linux, 26–28
    - installing on Windows, 32–33
    - logging error messages, 147–148
    - mailing list, 19
    - MySQL working with, 343–344
    - naming variables, 113–114
    - obtaining for Linux, 23
    - obtaining for the Mac OS, 24
    - obtaining for Windows, 23
    - online manual, 139
    - pattern matching
      - functions, 158–159
    - Security Blog, 420
    - Security Consortium, 420
    - sending e-mail with, 459–465
    - syntax, 107–109
    - testing, 38–39, 94–95
    - Tidy, 419
    - troubleshooting
      - settings, 43
    - types of error messages, 142–144
    - updates, 19–20
    - versions, 13, 22
      - Web site, 20
      - writing code, 109–110
- php -re extensions, 425–426
- php -ri extension, 426
- php -m function, 425
- \$PHPSESSID, 528
- <?php ?> statement, 105
- PHP error messages
  - fatal errors, 142, 143
  - notices, 142, 144
  - parse errors, 142–143
  - strict messages, 142, 144
  - warnings, 142, 143–144
- PHP Extension and Application Repository. *See* PEAR (PHP Extension and Application Repository)
- PHP extensions. *See* extensions
- PHP installation
  - checking for, 22
  - process overview, 21
  - troubleshooting, 42–45
- PHP programming
  - error handling, 397–401
  - uploading files, 403–408
  - variables, 401–403
- PHP scripts, 111
  - adding comments to, 148–149
  - breaking out of a loop, 177–178
  - checking variable content, 154
  - comparing values, 152–154
  - conditional statements, 161–167
  - do...while loops, 174–175
  - extensions, 104
  - functions, 178–189
  - if statements, 161–165
  - infinite loops, 175–176
  - joining multiple
    - comparisons, 159–161
  - loops, 167–178
    - for loops, 168–171

- organizing, 189–196
- pattern matching, 155–159
- setting up conditions, 152–161
- structure, 105–107
- switch statements, 165–167
- while loops, 171–173
- PHP sessions
  - closing, 527
  - opening, 527
  - overview, 526–527
  - using without cookies, 528–529
  - variables, 528
- PHP variables. *See also* variables
  - assigning values to, 114
  - creating, 114
  - defined, 113
  - displaying variable values, 116–118
  - naming, 113–114
  - removing information from, 114
  - uncreating, 114
  - variable, 115–116
- phpinfo() statement, 38, 195, 450–451
- php.ini
  - file, 96
  - setting security options, 395–396
- phpMyAdmin, 24
  - adding data from data files with, 326–327
  - adding rows of data with, 322–324
  - adding tables to databases with, 314–315
  - backing up databases with, 288–290
  - browsing data with, 327–328
  - changing database structure with, 317–318
  - changing passwords with, 280–281
  - changing privileges with, 283–284
  - creating accounts with, 279–280
  - creating empty databases with, 310
  - deleting databases with, 310–311
  - displaying account information from, 277–278
  - installing, 67–69
  - obtaining, 67
  - overview, 67
  - removing accounts with, 284–285
  - removing information with, 341
  - removing tables with, 316
  - restoring databases with, 292–293
  - sending SQL queries with, 265–267
  - testing, 69–70, 94
  - troubleshooting, 71–72
  - updating information with, 339–340
- PKG file, installing MySQL on Mac from, 57–58
- + (plus sign)
  - arithmetic operator, 121
  - comparison operator, 156
- POP3, 465
- port number, 85
- \$\_POST array, 402–403
- POST form, uploading files with a, 508–509
- POST method, 471, 490
- # (pound sign), 149
- prefix=*PREFIX* PHP
  - configure option, 31
- preg\_match function, 158, 499
- preventing
  - changes to classes, 251
  - changes to methods, 251
  - cross-site scripting, 417–419
- previous(\$arrayname) statement, 135
- primary key, 299
- print\_r statement, 116–118, 129
- private, 236–237
- private
  - methods/properties, 240–242
- privileges
  - ALL, 275
  - ALTER, 275
  - changing MySQL, 282–284
  - changing with
    - phpMyAdmin, 283–284
  - changing with SQL queries, 282–283
  - CREATE, 275
  - DELETE, 275
  - DROP, 275, 310–311
  - FILE, 275
  - granting, 411
  - INSERT, 275
  - MySQL account, 274–275
  - SELECT, 275
  - SHUTDOWN, 275
  - USAGE, 275
- processing
  - files, 104
  - information from forms, 490–505
  - uploaded files, 506–507
- product information
  - providing, 600–601
- shopping cart scripts, 601–606
- Web page, 581, 586–588

products page  
 designing, 561–562  
 Web page, 580–581, 584–586  
 writing code for, 564–566  
 programming applications, e-commerce, 409–419  
 properties  
 accessing using `$this`, 237  
 defined, 230  
 object-oriented programming (OOP), 231  
 overview, 231  
 public/private, 240–242  
 selecting for each object, 233–234  
 setting, 235–237  
 Properties dialog box, 392  
 protected, 236–237  
 protecting  
 MySQL databases, 267–268  
 Web pages, 553–554  
`ps -A` command, 83  
`public`, 236–237  
 public key encryption, 381–382  
 public methods/properties, 240–242  
 purchasing methods, shopping cart, 572

## Q

? (question mark) comparison operator, 156  
 queries  
 adding data from data files with SQL, 325–326  
 adding database tables with SQL, 311–313  
 adding rows of data in SQL, 321  
 adding tables to databases with SQL, 311–313  
`ALTER`, 261–262, 316–317

`ALTER TABLE`, 316–317  
 browsing data with SQL, 327  
 building SQL, 260–261  
 changing database structure with SQL, 316–317  
 counting rows returned by a, 351–352  
`CREATE DATABASE`, 309  
`CREATE TABLE`, 311–313  
`DELETE`, 261–262, 340  
 deleting databases with SQL, 310  
 displaying account information with SQL, 277  
`DROP`, 261–262  
`DROP USER`, 284  
`INSERT`, 261–262, 321  
`LOAD`, 261–262, 325–326  
 removing accounts with SQL, 284  
 removing information with SQL, 340  
`SELECT`, 261–262, 328–331  
`SELECT * FROM tablename`, 327  
 sending multiple, 348  
 sending SQL, 262–267, 263  
`SHOW`, 261–262  
`SHOW DATABASES`, 309  
`UPDATE`, 261–262, 339  
 queuing messages, 463–465

## R

`r` mode, 216  
`r+` mode, 216  
 radio buttons, 487–488  
 Ray, Deborah and Eric J., *HTML 4 For Dummies Quick Reference*, 4  
 RDBMS (Relational Database Management System), 9, 260  
 read mode, opening files in, 216–217

`readdir` function, 203  
 reading  
 comma-delimited files, 223  
 DOM (Document Object Model), 441–442  
 from files, 218–221  
 files into arrays, 220–221  
 files into strings, 221  
 files piece by piece, 219–220  
 reducing, server's footprint, 385–386  
 regenerate, sessions IDs, 417  
 regexes (regular expressions), 155, 400  
`register_globals = off` option, 395  
 registering, long arrays, 472  
 regressions tests, 431  
 regular expressions (regexes), 155, 400  
 reinstalling XAMPP, 97–98  
 Relational Database Management System (RDBMS), 9, 260  
 relocating, users with an HTTP header, 513–515  
 removing. *See also* deleting accounts with phpMyAdmin, 284–285  
 accounts with SQL queries, 284  
 information from databases, 340–341  
 information with phpMyAdmin, 341  
 information with SQL queries, 340  
 MySQL accounts, 284–285  
 tables, 316  
 values from arrays, 130–131  
`rename()` function, 408  
 rename statement, 201  
 renaming files, 200–201  
 request verification, 271  
`require()` function, 193, 426–427  
`require_once`, 193

`$result` variable, 347  
`reset($arrayname)`  
    statement, 135  
resizing, images, 452  
resource, 119  
Responsible Security  
    Personnel, 365  
restarting Apache on  
    Linux/Unix/Mac, 82  
restoring  
    data, 290–293  
    databases with  
        phpMyAdmin, 292–293  
    databases using mysql  
        client, 291–292  
retrieving  
    data in a specific order, 331  
    data from specific rows,  
        331–334  
    file information, 198–200  
    information in cookies, 523  
    information from  
        databases, 328–338  
    specific information,  
        329–331  
return statement,  
    179–180, 186–188  
reusing code, 191  
roles, security, 359–360  
root, 198  
root account, 270  
RPM (Red Hat Package  
    Manager) file, installing  
    MySQL on linux from, 57  
`rsort($arrayname)`  
    statement, 132  
`rtrim` function, 219

## S

s, date format symbol, 140  
safe-mode, 206  
Safe\_HTML, 419  
`safe_mode = on` option,  
    395

`safe_mode_gid = off`  
    option, 395  
salt, 380  
sample security policy,  
    365–371  
sanitizing e-mail addresses,  
    402–403  
SANS, 420  
saving information in  
    cookies, 523  
Schema  
    defined, 443  
    XML validation using,  
        443–445  
`schemaValidate()`  
    method, 444–445  
scripting language,  
    embedded, 105  
scripts  
    adding comments to PHP,  
        148–149  
    application, 566–570  
    building login, 545–553  
    building online catalog  
        application, 566–570  
    cross-site, 400, 417–418  
    defined, 189  
    display code, 474–476  
    object-oriented, 232–234  
    organizing with functions,  
        191–192  
    organizing with `include`  
        files, 192–196  
    organizing that display  
        forms, 473–476  
    reusing code, 191  
    separating display code  
        from logic code, 190  
    using a class in a, 246–247  
`script_tags()` function,  
    399, 400  
searching  
    whois, 12  
    XML documents with  
        XPath, 446–447  
Secure Hash Algorithm  
    (SHA1), 412

Secure Sockets Layer (SSL)  
    digital certificates,  
        412–415  
    e-commerce, 17  
    sending encrypted data  
        with, 412–415  
security  
    Apache, 383–385  
    communication, 372  
    cookies, 415–416  
    cross-site scripting (XSS),  
        417–419  
    database, 409–412  
    e-commerce, 359–364, 372  
    e-mail, 462  
    ensuring physical, 366  
    IIS (Internet Information  
        Server), 385–395  
    mission statement, 365  
    MySQL database, 276  
    operating system  
        commands, 209–210  
    options in `php.ini`  
        setting, 395–396  
    PHP applications with  
        SuExec, 383–384  
    policy, 363–371  
    roles, 359–360  
    session IDs, 417  
    session timeouts, 416  
    sessions, 415–417  
    setting options in  
        `php.ini`, 395–396  
    software updates, 419–420  
    threats, 361–363  
    Web root, 387–395  
    Web sites, 420  
security policy  
    components, 364  
    development, 363–364  
    sample, 365–371  
SecurityFocus, 420  
SELECT \* FROM  
     query, 327  
SELECT privilege, 275  
SELECT query, 261–262,  
    328–331

- selecting
  - data for database design, 295–296
  - database, 349
  - database users, 410
  - development
    - environment, 16–17
    - methods for each object, 233–234
  - objects, 233
  - properties for each object, 233–234
- selection lists
  - building, 480–486
  - defined, 480
- ;(semicolon) special character, 170
- sending
  - e-mail with PHP, 459–465
  - encrypted data with Secure Sockets Layer (SSL), 412–415
  - multiple queries, 348
  - SQL queries, 262–267, 347–348
- sendmail\_from, 460
- sendmail\_path, 460
- SERIALAUTO\_INCREMENT
  - MySQL data type, 303
- server
  - administrator, 360
  - footprint, 385–386
- services, disabling, 385–386
- \$\_SESSION array, 528
- session function, 514
- session\_destroy()
  - statement, 527
- session\_regenerate\_id() function, 417
- sessions
  - closing PHP, 527
  - security, 415–417
  - shopping cart variables, 573
  - timeouts, 416
- sessions IDs, regenerate, 417
- session\_start function, 527
- setcookie function, 523–524
- setcookie statement, 514
- setting up
  - Access Control list (ACL), 387
  - conditions, 152–161
  - include directories, 195–196
  - local computer for development, 17–19
  - MySQL accounts, 275–285
- settings
  - changing Apache, 84–85
  - default time zones, 139
  - display\_errors = Off, 396
  - display\_errors = On, 145
  - error\_log, 147
  - error\_reporting = , , 145–146
  - expiration time on cookies, 523–524
  - file\_uploads = On, 403
  - local time, 139
  - log\_errors, 147
  - properties, 235–237
  - security options in php.ini, 395–396
  - troubleshooting PHP, 43
- Setup Wizard, Windows MySQL, 52–54
- SHA1() encryption scheme, 412
- SHAI (Secure Hash Algorithm), 412
- shipping fees, shopping cart, 573
- shipping form Web page, 583, 591–596
- shopping cart
  - building scripts, 600–616
  - building web pages, 579–600
  - confirmation Web page, 584
  - cookies, 573
  - creating the database, 574–579
  - credit card handling, 572–573
  - database variables, 573
  - defined, 571
  - designing, 571–574
  - designing Web pages, 579–584
  - functionality, 573–574
  - management, 600–601
  - overview, 571
  - product categories Web page, 580–581, 584–586
  - product information Web page, 581, 586–588
  - session variables, 573
  - shipping fees, 573
  - shipping form Web page, 582–583, 591–596
  - shopping cart scripts, 606–609
  - shopping cart Web page, 588–591
  - summary Web page, 583–584, 596–600
  - text file, 573
  - Web page, 582, 588–591
- shopping cart database
  - accessing, 578–579
  - adding data, 579
  - building, 577–578
  - CustomerOrder table, 575
  - designing, 574–577
  - OrderItem table, 575–576

- shopping cart scripts
  - order, 600–601, 609–616
  - product information, 600–606
  - responsibilities, 600–601
  - shopping cart, 600–601, 606–609
- SHOW DATABASES query, 309
- SHOW query, 261–262
- SHUTDOWN privilege, 275
- signature, 25, 248
- simple statements, 107–108
- ' (single quote), 116, 123, 124–125, 402
- SMTP, 460
- SMTP\_port, 460
- software
  - administrative, 264–267
  - antivirus, 404
  - client, 263
  - help, 19
  - open-source, 19–20
  - SQLite, 225–227
  - tools, 10
  - updating, 419–420
  - Web hosting company, 15
- sort statement, 131
- sort(\$arrayname) statement, 132
- sorting, arrays, 131–132
- source code, installing
  - Apache on Linux/Mac from, 79–80
- source files, installing
  - MySQL from, 58–60
- SourceForge, 431
- specifying log files, 147–148
- sprintf, 122
- SQL (Structured Query Language)
  - building queries, 261–262
  - defined, 260
  - injection, 397–399
  - sending queries, 262–267
- SQL (Structured Query Language) formats
  - AVG(*columnname*), 330
  - COUNT(*columnname*), 330
- SQL (Structured Query Language) queries
  - adding data from data files with, 325–326
  - adding rows of data in, 321
  - adding tables to databases with, 311–313
  - browsing data with, 327
  - building, 260–261
  - changing database structure with, 316–317
  - changing passwords with, 280
  - changing privileges with, 282–283
  - creating accounts with, 278–279
  - creating empty databases with, 309
  - defined, 66
  - deleting databases with, 310
  - displaying account information with, 277
  - removing accounts with, 284
  - removing information with, 340
  - removing tables with, 316
  - sending, 263, 347–348
  - updating information with, 339
- SQLite software, 225–227
- sqlite\_query function, 226
- [] (square brackets)
  - comparison operator, 156
  - special character, 128
- SSL. *See* Secure Sockets Layer (SSL)
- standards (code), 432
- starting
  - Apache on Linux/Unix/Mac, 81–82
  - Apache on Windows, 81
  - MySQL, 49, 61
- startingvalue, 168
- stateless, 511
- statements
  - \$fh = fopen("filename", "mode"), 216
  - arsort(\$arrayname), 132
  - asort, 132
  - assert, 132
  - break, 167, 177
  - class, 235
  - complex, 108–109
  - conditional, 151, 161–167
  - continue, 177
  - copy, 200–201
  - current(\$arrayname), 134
  - define, 119
  - defined, 151
  - echo, 106–108, 110, 116–118, 470
  - end(\$arrayname), 135
  - error\_reporting(E\_ALL), 146
  - error\_reporting(errorSetting), 146
  - fclose(\$fh), 218
  - fgets, 218
  - file\_exists, 198
  - ftp\_close(\$connect), 213
  - ftp\_nlist, 212
  - fwrite, 218
  - header, 514
  - heredoc, 126
  - if, 161–165
  - include, 192–196
  - krsort(\$arrayname), 132

statements (*continued*)

- ksort(*\$arrayname*), 132
- next(*\$arrayname*), 134
- opendir, 202–203
- <?php?>, 105
- phpinfo(), 38, 195, 450–451
- previous(*\$arrayname*), 135
- print\_r, 116–118, 129
- rename, 201
- reset(*\$arrayname*), 135
- return, 179–180, 186–188
- rsort(*\$arrayname*), 132
- session\_destroy(), 527
- setcookie, 514
- simple, 107–108
- sort, 131
- sort(*\$arrayname*), 132
- switch, 161, 165–167, 546
- unlink, 201
- usort(*\$arrayname*, *functionname*), 132
- var\_dump, 116–118, 129

static HTML forms

- displaying, 470–476
- overview, 469

stopping

- Apache on
  - Linux/Unix/Mac, 82–83
  - Apache on Windows, 81

storing

- character strings, 126–127
- connection strings, 411
- data types, 301–303
- encrypted passwords, 412
- include files, 194–195
- information in cookies, 522–526
- passwords, 376, 411–412
- timestamps in variables, 141–142
- (string), 120
- strict, 142

- strict messages, 144
- strings. *See also* character strings
  - assigning to variables, 123–124
  - character, 262
  - connection, 411
  - defined, 119
  - joining, 125–126
  - reading files into, 221
  - storing, 126–127
  - text, 262
- strtotime, 141
- structure, database, 260, 316–318
- Structured Query Language. *See* SQL (Structured Query Language)
- (subtraction sign)
  - arithmetic operator, 121
- subclass, 232
- subdomain, 12
- SuExec, securing PHP
  - applications with, 383–384
- SUM(*columnname*) SQL
  - format, 330
- summary Web page, 583–584, 596–600
- superglobal arrays, 471
- suppressing single error messages, 146
- switch statement
  - building login script, 546
  - defined, 161
  - using, 165–167
- syntax
  - mysql function, 355
  - mysqli function, 355
  - PHP, 107–109
  - system function, 205, 207

## T

---

- tab-delimited file, 224–225, 324
- tables
  - adding to databases with phpMyAdmin, 314–315
  - adding to databases with SQL queries, 311–313
  - columns\_priv, 276
  - combining information
    - from, 334–338
  - creating relationships
    - between, 300–301
  - CustomerOrder, 575
  - data organization into, 298–300
  - db, 276
  - defined, 297
  - host, 276
  - OrderItem, 575–576
  - removing, 316
  - tables\_priv, 276
  - user, 276
  - tables\_priv table, 276
- tabs, inserting, 124–125
- tags
  - <br />, 112
  - <form>, 493
- technical support, Web
  - hosting company, 14
- T\_ECHO, 143
- terminal monitor, 263
- testing
  - Development
    - Environment, 92–95
    - MySQL, 63–64
    - PHP, 38–39, 94–95
    - phpMyAdmin, 69–70, 94
  - for unexpected input, 399–400
  - Web server, 73–74

- text
    - adding to images, 455–457
    - displaying values in fields, 477–480
    - files, 215, 573
    - strings, 262
  - TEXT MySQL data type, 303
  - Text\_CAPTCHA, 378
  - Text\_CAPTCHA\_Numeral, 378
  - Thawte, 413
  - `$this`, accessing
    - properties using, 237
  - threats, security, 361–363
  - throwing an exception, 251
  - time
    - data, 302
    - setting local, 139
  - TIME MySQL data type, 303
  - time zone
    - codes Web site, 139
    - setting a default, 139
  - timestamp
    - defined, 138
    - storing in a variable, 141–142
  - Tittel, Ed
    - HTML 4 For Dummies*, 469
    - XML For Dummies*, 444
  - token, 143
  - tools, software, 10
  - `trans-sid`, 528–530
  - transfer, data, 14
  - transparent session ID, 416
  - traversing, 134
  - troubleshooting
    - blank pages, 45
    - HTML output only, 45
    - Mac error messages, 44
    - MySQL, 64–66
    - MySQL error messages, 64–66
    - MySQL function
      - activation, 44–45
    - PHP installation, 42–45
    - phpMyAdmin, 71–72
    - XAMPP, 98–99
    - try block, 252–253
    - TSV (tab-separated values)
      - file. *See* tab-delimited file
    - turning off, error messages, 145
    - type hinting, 238
- 
- U
- uncreating PHP variables, 114
  - `--` (underscores) method, 239
  - undefined function,
    - troubleshooting error message, 44
  - uninstalling XAMPP, 97–98
  - UNION, 334–336
  - Unix
    - checking MySQL
      - installation, 48–49
    - checking PHP installation, 22
    - configuring PHP, 36
    - getting Apache
      - information, 83–84
    - installing PHP on, 26–28
    - obtaining MySQL for, 50–51
    - PHP installation options, 31–32
    - restarting Apache, 82
    - starting Apache, 81–82
    - stopping Apache, 82–83
    - Timestamp, 139
  - unlink statement, 201
  - UNSIGNED definition, 311
  - UPDATE query, 261–262, 339
  - updating
    - defined, 369–370
    - information in databases, 339–340
    - information with
      - phpMyAdmin, 339–340
  - information with SQL
    - queries, 339
  - MySQL, 19–20
  - PHP, 19–20
  - software, 419–420
  - upgrading MySQL, 293–294
  - uploading
    - files, 403–408
    - files with FTP (File Transfer Protocol), 212–214
    - files with FTP (File Transfer Protocol)
      - functions, 405–406
    - files with a POST form, 508–509
  - URL
    - adding information to, 515, 516–517
    - adding variables to, 516–517
  - USAGE privilege, 275
  - user
    - database, 534–537
    - hijacking, 400
  - user table, 276
  - usort(`$arrayname`, `functionname`)
    - statement, 132
- 
- V
- validating
    - data, 491
    - files, 404
  - value pairs, 128
  - values
    - assigning to PHP
      - variables, 114
    - comparing, 152–154
    - displaying in text fields, 477–480
    - displaying variable, 116–118
    - obtaining from arrays, 133–134

values (*continued*)  
 passing to functions, 181–186  
 removing from arrays, 130–131  
 returning from functions, 186–188

VARCHAR(*length*) MySQL data type, 303

var\_dump function, 118, 120

var\_dump statement, 116–118, 129

variable-length format character data, 301

variables. *See also* PHP variables  
 \$result, 347  
 adding to the URL, 516–517  
 assigning strings to, 123–124  
 assigning values to PHP, 114  
 content checking, 154  
 converting HTML special characters, 401–402  
 creating PHP, 114  
 database, 573  
 defined, 113  
 displaying with print\_r statements, 117–118  
 displaying with var\_dump statements, 118  
 global, 180  
 local, 180  
 naming PHP, 113–115  
 PHP programming, 401–403  
 PHP sessions, 528  
 sanitizing e-mail addresses, 402–403  
 storing timestamps in, 141–142  
 using in echo statements, 116–117

using in functions, 180–181  
 using in include statements, 193–194

|| (vertical lines) pattern character, 156, 161

verifying connections, 271  
 downloaded files, 24–25, 52, 77

Verisign, 413

versions  
 Apache, 18, 74  
 MySQL, 50, 293–294  
 PHP, 13, 22  
 XAMPP, 87

viewing arrays, 129–130

## W

w, date format symbol, 140

w mode, 216

w+ mode, 216

WAMP5, 24, 52, 77

warnings  
 defined, 142  
 handling, 143–144

Web front end, installing  
 PEAR Package Manager via, 433–435

Web hosting  
 company Web site, 11  
 defined, 10  
 educational institution, 12–13  
 overview, 10–11

Web hosting company considerations, 14–15

Web pages  
 building login, 537–545  
 confirmation, 584  
 delivery stages, 111–112  
 displaying catalog, 566  
 displaying content, 110–113

product categories, 584–586

product information, 586–588

production stages, 111

protecting, 553–554

shipping form, 591–596

summary, 596–600

Web root, securing, 387–395

Web servers  
 configuring for PHP, 33–36  
 configuring on Windows, 34–36  
 defined, 73, 103  
 installing, 17–18  
 PHP file processing, 104  
 testing, 73–74

Web site developer, 360

Web sites (features and extensions)  
 binary files, 24  
 downloading from the MySQL, 50  
 downloading from the PHP, 22–23  
 F-Prot, 404  
 GD, 451  
 Google Code, 431  
 MAMP, 24  
 MD5 signature checkers, 25  
 mod\_ssl, 415  
 PEAR library, 430  
 PEAR mailing lists, 433  
 PEAR package browser, 437  
 PEAR's Validate class, 419  
 PHP Tidy, 419  
 Safe\_HTML, 419  
 SourceForge, 431  
 WAMP, 24  
 winMd5Sum, 25  
 XAMPP, 24, 88

- Web sites (general)
  - company, 11
  - dynamic, 469
  - educational institution, 12–13
  - host selection, 10–16
  - making information available to all pages, 522–532
  - navigating, 511–515
  - opening files on another, 217
  - passing information between pages, 515–522
  - security, 420
  - time zone codes, 139
  - using a hosted, 15–16
- Web sites (informational)
  - appendix of tokens, 143
  - book, 5
  - CERT, 420
  - coding standards, 432
  - editor information, 142
  - MySQL, 20
  - MySQL data types, 303
  - MySQL online manual, 274
  - MySQL reserved words, 299
  - MySQL upgrade information, 293
  - Netcraft survey, 103
  - OWASP, 420
  - PHP, 20
  - PHP online documentation, 189
  - PHP Security Blog, 420
  - PHP Security Consortium, 420
  - SANS, 420
  - SecurityFocus, 420
  - WindowSecurity.com, 420
  - XSS page, 418
- Web sites (search tools)
  - Allwhois, 12
  - BetterWhois, 12
  - Google, 12
- Web space
  - changing location, 85
  - defined, 94
- WHERE clause, 331–334
- while loops
  - defined, 167
  - file management, 203
  - using, 171–173
- whois searches, 12
- Windows
  - activating MySQL support, 40–42
  - checking MySQL installation, 48–49
  - checking PHP installation, 22
  - Components Wizard, 86
  - configuring Apache, 34–35
  - configuring PHP, 36
  - configuring Web Server, 34–36
  - controlling MySQL Server on, 61–62
  - Features dialog box, 86
  - getting Apache information on, 83
  - installing Apache on, 77–79
  - installing MySQL, 52–56
  - installing PHP, 32–33
  - MySQL Configuration Wizard, 53, 55–56
  - MySQL Setup Wizard, 52–54
  - obtaining Apache for, 75
  - obtaining MySQL for, 50
  - obtaining PHP for, 23
  - starting/stopping Apache, 81
  - troubleshooting error messages, 44
  - troubleshooting MySQL function activation, 44–45
- winMd5Sum, 25
- with-apxs2=*FILE* PHP
  - configure option, 31
- with-apxs=*FILE* PHP
  - configure option, 31
- with-config-file-path=*DIR* PHP
  - configure option, 31
- with-mysql=*DIR* PHP
  - configure option, 32
- with-mysqli=*DIR* PHP
  - configure option, 32
- with-oci8=*DIR* PHP
  - configure option, 32
- with-openssl=*DIR* PHP
  - configure option, 32
- with-oracle=*DIR* PHP
  - configure option, 32
- with-pgsql=*DIR* PHP
  - configure option, 32
- with-servlet=*DIR* PHP
  - configure option, 32
- Wizard
  - Windows Components, 86
  - Windows MySQL Configuration, 53, 55–56
- write mode, opening files in, 217
- writing
  - class statements, 235
  - code for index pages, 562–564
  - code for login Web page, 538–544
  - code for products page, 564–566
  - code for shopping cart web pages, 584–600
  - constructors, 242
  - to DOM (Document Object Model), 442–443
  - to files, 218
  - PHP code, 109–110

## X

---

### XAMPP

- all-in-one installation kit, 24, 52, 76–77
  - Control Panel, 91–95, 98
  - downloading, 88
  - installing, 88–91
  - obtaining, 88
  - opening Web pages, 93–94
  - overview, 87–88
  - reinstalling, 97–98
  - troubleshooting, 98–99
  - uninstalling, 97–98
  - versions, 87
- XML documents, searching with XPath, 446–447

### XML extension

- Document Object Model (DOM), 441–443
  - XML validation, 443–445
  - XPath, 446–447
  - XSLT, 445
- XML For Dummies* (Dykes and Tittel), 444
- XML validation, using Schema, 443–445
- xor, 159
- XPath
- defined, 446
  - searching XML documents with, 446–447

### XSLT (Extensible Stylesheet Language

- Transformation), styling documents with, 445
- XSS. *See* cross-site scripting (XSS)

## Y

---

- Y, date format symbol, 140

## Z

---

- Zend engine extensions, 423