Index

Symbols

& (ampersand), 211
&& (AND operator), 608
`` (backquotes/backticks), 143
& (bitwise AND) operator, 140
~ (bitwise NOT) operator, 141
| (bitwise OR) operator, 140
^ (bitwise XOR) operator, 140
^ (caret metacharacter), 518, 520–521
" (double quote), 211
@ (error control) operator, 143–145
> (greater than), 211, 608, 611-612
<< (left shift) operator, 141
< (less than), 211, 608, 611-612
* metacharacter, 536–538
!, NOT operator, 608
< operator, 608, 611–612
<= operator, 608
<>, != operator, 608–609
= operator, 608–609
> operator, 608, 611–612
-> operator, 743–744
>= operator, 608
|| (OR operator), 130–132, 608
>> (right shift) operator, 141
' (single quote), 211, 352–353, 609
% wildcard, 613–614
>>> (zero-fill right shift) operator, 141

A

abs() function, 117
Access methods, 53, 758–760
ACTION attribute, 91
action attribute, 389–390, 421
addcslashes() function, 206–208, 214
Addition (+), 112
addslashes() function, 206–208, 214
admin_art_edit.php, 784, 793–796
admin_art_list.php, 784
admin_artist_edit.php, 784, 788–789
admin_artist_insert.php, 784, 791–792
admin_artist_list.php, 784, 786–787, 796
admin_footer.php, 784
admin_header.php, 784, 803
admin_login.php, 784, 797, 803
Advisory locking, 474
Aliases, 630–631
Alphabetic sort of, 299–300
Alphanumeric word characters, metasymbols representing, 531–533
ALTER command, 631
ALTER TABLE statement, 620, 631–633
Alternation, metacharacters for, 543
Anchoring metacharacters, 520–523
  beginning-of-line anchor, 520–523
  end-of-line anchor, 522
  word boundaries, 523

index.fm Page 856 Wednesday, October 25, 2006 1:28 PM

AND operator, 612–613
&& (and) operator, 127–128
  compared to using the word AND, 132
Apache Web server, 14, 834
  configuring to use PHP module (all platforms), 851–852
  installing on Windows, 849–850
  testing installation, 852
Appending to files, 473–474
Application Programming Interface (API), 5
area.php, 146
Arguments, 27, 29, 339, *See also* Passing arguments
  defined, 341
  missing, 344–345
  names of, 341–342
  passing, 341–351
  variable number of, 345–347
Arithmetic operators, 112–113
array() function, 251–252, 330
(array) operator, 119
array_change_key_case() function, 333
array_chunk() function, 332
array_combine() function, 251
array_count_values() function, 278–279, 333
array_diff() function, 333
array_diff_assoc() function, 333
array_diff_key() function, 333
array_diff_ukey() function, 333
array_diff_unassoc() function, 333
array_fill() function, 251, 255–256, 330
array_filter() function, 332
array_flip() function, 332
array_intersect() function, 334
array_intersect_assoc() function, 334
array_intersect_key() function, 334
array_intersect_uassoc() function, 334
array_intersect_ukey() function, 334
array_key_exists() function, 331
array_keys() function, 281–282, 331
array_map() function, 332, 359, 359–360
array_merge() function, 323, 325, 331
array_merge_recursive() function, 326, 331
array_multisort() function, 298, 334
array_pad() function, 251
array_pop() function, 311–312, 332
array_product() function, 332
array_push() function, 315, 317–318, 332
array_rand() function, 306, 310, 332
array_reduce() function, 332
array_reverse() function, 332
array_search() function, 333
array_shift() function, 311, 313, 332
array_slice() function, 320–321, 332
array_splice() function, 311, 315, 318–321, 332
array_sum() function, 331
array_udiff() function, 333
array_udiff_assoc() function, 333
array_udiff_uassoc() function, 333
array_uintersect() function, 334
array_uintersect_assoc() function, 334
array_uintersect_uassoc() function, 334
array_unique() function, 314–315, 332
array_unshift() function, 315–316, 332
array_values() function, 283–284, 331, 333
array_walk() function, 332
array_walk_recursive() function, 332, 360
Arrays, 46, 95, 247–335
  accessing an array's elements (values), 257–262
  alphabetic sort, 299–300
  array() function, 251–252, 330
  array identifier [ ], 252–253
  array_combine() function, 251
  array_count_values() function, 278–279, 333
  array_fill() function, 251, 255–256, 330
  array_keys() function, 281–282, 331
  array_merge() function, 323, 325, 331
  array_pad() function, 251
  array_push() function, 315, 317–318, 332

array_slice() function, 320–321, 332
array_unshift() function, 315–316, 332
array_values() function, 283–284, 331, 333
associative, 248
    elements of, 259–260
    merging, 325
    sort by key, 305–306
    sort by values, 302–303
checking for the existence of, 274–275
combining/merging, 322–327
compact() function, 251
count() function, 278–279
creating and naming, 251–256
creating from strings, 275–277
creating strings from, 275–277
creating variables from array elements, 286–293
defined, 247
each() function, 284–285, 288, 331
elements, 247
    adding, 315–319
    copying, 320–321
    removing, 310–315
    extract() function, 97, 288–293, 331, 398–399, 788
extracting keys and values from, 281–286
finding the size of, 278–280
functions, 330–335
list() function, 286–288
and loops, 266–273
mixing elements in, 260–262
modifying, 310–335
multidimensional, 293–297
numeric, 248
    elements of, 257–258
    shuffling, 309–310
numeric sort, 300–301
numerically indexed, merging, 323–324
operators, 327–330
    equality operator, 329–330
    identical operator, 329–330
    union operator, 327–328
printing, 262–266
    print_r() function, 262–264
    var_dump() function, 265–266
randomizing, 306–310
range() function, 251, 253–255
removing, 310–315
removing duplicate values from, 314–315
reversed sort, 302
sizeof() function, 277–279, 333
sort by key, 305–306
sort by values, 302–303
    in reverse order, 303–304
sorting, 298–306
splicing, 318–321
arsort() function, 298, 303–304, 334
Art gallery, building (example), *See* Canvas Gallery Web project
artist_detail.php, 784, 798, 800–801
AS keyword, 630
ASCII character set, 125
asort() function, 298, 334
ASP, 14, 41
asp_tags configuration option, 853
Assignment operator, 107–108
Assignment statement, defined, 108
Associative arrays, 248
    and extract() function, 288–293
    merging, 325
    sort by key, 305–306
    sort by values, 302–303
Associativity, 108–111
    rules of, for PHP operators, 110
Attributes, 52, 675, 740–741
AUTH_TYPE, 98
AUTO_INCREMENT attribute, 625
Autoglobal arrays, 431
Autoincrement and autodecrement operators, 115–117
AVG() function, 634
Axmark, David, 7

B

Backquotes, 143
Backslash (/), 62
Backticks, 74, 143
Base class, 763

index.fm Page 858 Wednesday, October 25, 2006 1:28 PM

base_convert() function, 117
basename() function, 485–486
BBEdit, 15
Beginning-of-line anchor, 520–523, 521
Berners-Lee, Tim, 1, 3
BETWEEN operator, 608, 615
BIG5, 214
BIG5-HKSCS, 214
Binary operators, 107
bindec() function, 117
binhex() function, 214
Bits, 140
Bitwise operators, 140–143
  shift operators, 141–142
  using, 140
Black box, *See* Functions
BLOB data type, 622
Block, 132
(bool) operator, 119
Boolean literals, 68
Boolean values, 44
break statement, 51, 239–243
Browser:
  and HTML forms, 380–383
  using to remove cookies, 693–694
Built-in constants, 103
Built-in functions, PHP, 27–28, 69
Bytes, 140, 149

C

Callback functions, 358, 358–362
  array_map() function, 359–360
  array_walk_recursive() function, 360
  defined, 358
Canvas Gallery Web project, 783–808
  administration pages, 786–788
  database, creating, 785–786
  files, 784
  installing, 805–808
  MySQL database, installing, 806–807
  PHP pages, editing, 808
  public pages, 797–803
    artist_detail.php page, 798, 800–801
    contact.php page, 802–803
    header.php page, 797–798
    index.php page, 798–799
  public/private pages, 783
  securing pages with login, 803–805
  Web site, creating, 785–805
  where to find the files for, 805
Capturing, 545–547
Caret (^) metacharacter, 518, 520–521
Carriage return, 24
Case sensitivity, SQL commands, 599–600
Case statement, 228
Casting operators, 118–120
C/C++, 41, 49
ceil() function, 117
CGI script, 2
Character classes, 525–526
Character list (charlist), 182, 512
charset, 211
CHAR(SIZE) data type, 622
checkbox input type, 388, 406–407
checkdate() function, 831–832
chgrp() function, 447
chmod() function, 447–448
chop function, 167fn, 182, 214
chown() function, 447
chr() function, 205, 214
chunk_split function, 214
Chunks, 814
Class constants, 774–775, 775
Class members, 742
__CLASS__, 103
Classes, 52–53
  accessor or instance methods (setters/getters), 758–760
  class constants, 774–775
  constructors, 753–756
  creating, 746–747
  defined, 740
  destructors, 756–758
  encapsulation, 749–750
  functions for getting class information, 748–749
  information hiding, 749–750
  inheritance, 741, 763–766
  instantiating, 741–742

magic methods, 753–762
members, 742
and scope, 750–752
methods, 742
and objects, 740
difference between, 742
overriding methods, 766–768
properties, 742
protected access, 768–771
reusing, 776–780
static members, 773–774
working with, 741–772
Classes and objects, 52–54
Client/server databases, 568
Client/server model, 7
clock.php, 354
Clustering, 544, 544–545
Codd, E.F. “Ted,” 595
ColdFusion, 14
Command, 597
Command-line applications, defined, 32
Command-line options, PHP, 34–35
Comments, 25–27, 42
Commercial license, MySQL, 570
Common Gateway Interface (CGI), 2–3
compact() function, 251
Comparison operators, 122–126
equality, 122–123
Composite data types, 59
Concatenation, 120
defined, 120
operators, 120–121
and variables, 79–80
CONCAT(string1,string2,...) function, 637
Conditional operator, 49, 138–139
Conditional returns, 353–355
Conditionals, 46–48, 222–231
if/else branching construct, 222–225
if/elseif construct, 225–227
switch statement, 228–231
constant() function, 102
Constants, 43, 62, 99–103, 774–775
built-in, 103
constant() function, 102
creating with the define() function, 100–101
defined, 99–100
magic, 102
predefined, 102–103
__construct() method, 753
Constructors, 753–756
and inheritance, 763–766
contact.php, 784
contact.php page, 802–803
continue control statement, 239–243
continue statement, 51
Control statements, loops, 239–243
Control string, 156
convert_cyr_string function, 214
convert_uudecode function, 214
convert_uuencode function, 214
converter() function, 377
$_COOKIE global array, 678–681
Cookie jar, 674
Cookies, 673–736
attributes, 675–677
buffering and HTTP headers, 689–693
defined, 673–674
deleting, 693–694
domain name, 676
expiration date, 676
extending the life of, 688–689
ingredients, 674–675
magic cookie, 674
name, 675
output buffering and php.ini, 693
path, 676
and PHP, 677–694
$_COOKIE global array, 678–681
serialization, 681–683
setcookie() function, 677–678
tracking visitors, 683–687
security, 677
sessions without, 717–720
starting a cookie-based session, 698–699
turning off, 674
use of term, 674
using the browser to remove, 693–694
copy() function, 481–482

index.fm Page 860 Wednesday, October 25, 2006 1:28 PM

count() function, 278–279, 296–297, 333
COUNT() function, 634
count_chars function, 214
COUNT(*expression*) function, 634
cp866, 213
cp1251, 213
cp1252, 214
crc32 function, 214
CREATE INDEX statement, 620
CREATE TABLE statement, 620, 623, 663
Cross joins, 628
crypt function, 214
curdate() function, 592
CURDATE() function, 638
Curly brace syntax, 200
Current directory, changing/getting, 486–487
current() function, 333
CURTIME() function, 638

D

Data Definition Language (DDL), 620–633
  altering a table, 631–632
  creating a key, 625–627
    foreign keys, 627
    primary keys, 625–626
  creating a table, 623–625
  creating the database, 620–621
  dropping a database, 633
  dropping a table, 633
  relations, 628–631
    aliases, 630–631
    two tables with a common key, 628–629
    using a fully qualified name and a dot to join tables, 629–630
  SQL data types, 621–622
Data hiding, 750
Data Manipulation Language (DML), 603–620
  DELETE statement, 619–620
  INSERT command, 617–618
  SELECT command, 604–617
    LIKE operator, 608, 613
    LIMIT, 606–607
    NOT LIKE operator, 608, 613
    NULL, 609–611
    AND operator, 612–613
    BETWEEN operator, 615
    ORDER BY clause, 616–617
    pattern matching and the % wildcard, 613–614
    quotes, 609
    SELECT DISTINCT statement, 605–606
    selecting all columns, 605
    selecting specified columns, 604
    WHERE clause, 608–609
  UPDATE command, 618–619
Data types, 43, 59–70
  Boolean literals, 68
  defined, 59
  numeric literals, 60
  special, 68–70
  SQL, 621–622
  string literals and quoting, 62–67
Database design, 575
database() function, 592
Database schema, 575
Database tables, 601–603
Database-driven Web site, example of building, 663–670
  designing the database, 663–664
  example summary, 670
  listing all messages, 669–670
  posting a message page, 665–668
Databases, 567–570, *See also* Relational databases
  client/server, 568
  connecting to, 575–581
  defined, 571
  dropping, 633
  graphical user tools, 579–581
  MySQL command-line options, 576–578
  relational, 570–575
  SQL (Structured Query Language), 568–569
  talking to, 568–570
databook file, 466, 468
data.txt, 520
Date and time functions:
  date and time parts, 641–642

index.fm Page 861 Wednesday, October 25, 2006 1:28 PM

EXTRACT command, 641–642
formatting the date and time, 639–641
SQL, 638–642
DATE data type, 622
date() function, 29, 31, 38, 73, 77, 819–821, 829
DATE_FORMAT() function, 639–641
Date/time:
checkdate() function, 831–832
date() function, 819–821, 829
formatting dates and times, 819–824
getdate() function, 829–830
mktime() function, 825–827, 829
and PHP, 819–832
strftime() function, 822–824
time() function, 824–825
UNIX timestamp, 819
creating from a string, 828–829
validating dates, 831–832
DATETIME data type, 622
DAYNAME(date) function, 638
DAYOFMONTH(date) function, 638
DAYOFWEEK(date) function, 638
DAYOFYEAR(date) function, 638
db.sql, 784
DDL, *See* Data Definition Language (DDL)
Debugging, 841–847
diagnostic print statements, 843–845
error reporting, turning on, 841
parse errors, 841–842
SQL errors, fixing, 845–847
decbin() function, 117
dechex() function, 117
DECIMAL(m,d) data type, 622
Decision-making constructs, 222
decoct() function, 117
Default condition, 227
define() function, 43, 100–101
creating constants with, 100–101
DELETE statement, 619–620
DESCRIBE command, 602–603, 664
__destruct() method, 753, 757
Destructors, 756–758
Diagnostic print statements, 843–845
Digits and spaces, metasymbols representing, 530–531
Directories, 483–487
current, changing and getting, 486–487
opendir() function, 483–484
path information, getting, 485–486
readdir() function, 484–485
dirname() function, 485–486
display_errors directive, 840, 853
DISTINCT keyword, 605–606
Division (/), 112
DML, *See* Data Manipulation Language (DML)
DOCUMENT_ROOT, 98, 433
Dot metacharacter, 524–525
Double data type, 59
Double word, 140
DOUBLE(m,d) data type, 622
Doubles, 60
do-while loop, 49–50, 234
drop command, 633
drop database command, 633
DROP INDEX statement, 620
DROP TABLE statement, 620, 633, 664
Drop-down lists, 387
Dropping a database, 633
Dropping a table, 633
Dynamic function calls, 351
Dynamic variable, 82
Dynamic Web sites, 2–3
process of creating, 3

E
each() function, 284–285, 288, 331
eBay, 7
echo() function, 42, 76, 215
Edit distance, 177
Element, 247
emacs, 15
E-mail, and PHP, 809–817
empty() function, 88–89
Encapsulation, 749–750
enctype attribute, 421
end() function, 334

index.fm Page 862 Wednesday, October 25, 2006 1:28 PM

Ending sessions, 727–730
End-of-line anchor, 522
ENT_COMPAT, 211
ENT_NOQUOTES, 211
ENT_QUOTES, 211
Environment variables, 84, 437–438
Epoch, 819
Equality, 122–123
Equality operator, arrays, 329–330
ereg() function, 499
ereg_replace() function, 499
eregi() function, 499
eregi_replace() function, 499
Error control operators, 143–145
Error reporting settings, 839–840
error_log directive, 841
error_reporting directive, 839–840, 853
Escape sequences, 62, 66–67
EUC-JP, 214
Exclusive OR operator, 134
Execution operators, 143
Expiration date, cookies, 676
explode() function, 179, 215, 275, 276–277, 331, 466–467, 510
Expressions:
  defined, 107
  testing with logical operators, 126–127
extension_dir configuration option, 854
EXTR_IF_EXISTS, 289
EXTR_OVERWRITE, 289
EXTR_PREFIX_ALL, 289, 292
EXTR_PREFIX_IF_EXISTS, 289
EXTR_PREFIX_INVALID, 290
EXTR_PREFIX_SAME, 290
EXTR_REFS, 290
EXTR_SKIP, 290–291
extract() function, 97, 288–293, 331, 398–399, 788
  and associative arrays, 288–293

F

FALSE, 103
fclose() function, 450, 474
feof() function, 450, 451
fflush() function, 450
fgetc() function, 450, 455–456
fgets() function, 450, 451–452
fgetscsv() function, 450
fgetss() function, 450, 453–454
Fibonacci, Leonardo, 370
Fibonacci sequence, 370
Fields, 567, 572–573, 601
File checks, 475–481
file() function, 450, 465–466
file input type, 389
File pointer, 441, 448
  positioning, 458–462
__FILE__, 103
file_exists() function, 450, 451, 476–477
file_get_contents() function, 450, 464–465
file_put_contents() function, 472
Files, 54–55, 441–446
  appending to, 473–474
  chgrp() function, 447
  chmod() function, 447–448
  chown() function, 447
  filehandle, 448
  fopen() function, 448–450
  locking implications, 474–475
  opening, 448–450
    for reading, 451–458
    for writing/appending, 471–475
  opening a URL for reading, 463–464
  useful file functions, 450
  without a filehandle, reading from, 464–465
final classes and methods, 772–773
Firefox, 14
Flat files, 441
(float) operator, 119
Floating-point numbers, 60
FLOAT(m,d) data type, 622
flock() function, 474–475
floor() function, 117
Flow chart, defined, 221
footer.php, 784
fopen() function, 448–450, 471, 474
  defined, 450

index.fm Page 863 Wednesday, October 25, 2006 1:28 PM

modes for, 449–450
for loop, 50, 235–238
and repetitive form fields, 236
foreach loop, 50–51, 239
Foreign key, 627–628
Form validation with PHP, 556–562
regular expressions used for (table), 557
valid e-mail address, checking for, 557–562
Form variables, 90–99
extracting data by request, 95–97
how PHP handles form input, 91–95
php.ini file and register_globals, 90–91
predefined variables, 98–99
fprintf() function, 20, 42, 160, 215
fputs() function, 472
fread() function, 456–458
Free form language, defined, 23–24
Free Software Foundation, 3
fseek() function, 459–460
ftell() function, 461–462
Full joins, 628
func_get_arg($arg_num) function, 345–347
func_get_args() function, 345–347
func_num_args() function, 345–347
function construct, 52
function keyword, 338–339
Function libraries, 373–376
include() construct, 373–375
require() construct, 373–375
__FUNCTION__, 103
Functions, 52
compared to miniscripts, 338
declaring, 338
defined, 52, 337
definition, 338
how to call, 339–341
how to define, 338–339
naming conventions, 339
nesting, 368–370
recursive, 370–372
fwrite() function, 472

G
gallerydb.sql, 784
Garbage collection, 729–730
GATEWAY_INTERFACE, 432
GB2312, 214
gc_maxlifetime configuration directive, 729–730
GET method, 381–382
$_GET array, 400–402
__get() method, 753, 758
get_class() function, 744–745, 748–749
get_class_vars() function, 748–749
get_declared_classes() function, 748–749
get_html_translation_table function, 215
get_magic_quotes_gpc() function, 206
get_object_vars() function, 748–749
get_parent_class() function, 748–749
getrandmax() function, 117
Getters, 758–760
gettype() function, 69–70, 73, 277, 744–745, 748–749
global command, 52
Global scope, 363–364
Global variables, 84
GoDaddy, 834
Google, 2
Grant tables, 582
Greed factor, 534–535
Greedy match, 534
Greedy metacharacters, 533–534
* metacharacter, 536–538
+ metacharacter, 539
Grouping/clustering, 544–545
Gutmans, Andi, 4

H
Haystack, 192, 198
header() function, 417–418
header.php, 784, 797–798
hebrev function, 215
hebrevc function, 215
Helper program, 2

index.fm Page 864 Wednesday, October 25, 2006 1:28 PM

Here-documents (here-docs), 62, 64
  rules for, 64–65
hexdec() function, 118
Hidden fields, using, 415–417
hidden input type, 389
Homophones, 175
House class, 740–742
htdocs, 384
HTML forms:
  $_REQUEST array:
    extract() function, 398–399
    form parameters in, 396–399
  action attribute, 389–390
  browser's role, 380–383
  creating, 384–390
  dots and spaces in html input device names, 393–394
  form input types, 388–389
  form parameters and the medium style, 400–405
    $_GET array, 400–402
    $_POST array, 403–405
    security, 402
  GET method, 381–382
  image button, 410–412
  method attribute, 389
  POST method, 382–383
  processing forms with multiple selections, 406–409
    creating the form with select and check boxes, 406–408
    processing the form with multiple selections, 408–409
  request method, 393
  review of, 379–390
  self-processing, 413–420
  server information, how to get, 432–436
  server's role, 383–384
  slashes, stripping out, 395–396
  steps in producing, 385–386
  sticky forms, 427–431
  superglobal arrays, where to get information about, 431–432
  uploading files, 421–427
HTML (Hypertext Markup Language), 1–2
  compared to PHP, 21–23
HTML special characters, 210–214
html_entity_decode function, 215
htmlentities() function, 210, 213–214, 215
htmlspecialchars() function, 210, 210–215
htmlspecialchars_decode(), 210–215
HTTP (Hypertext Transfer Protocol), 1, 14, 379, 673
  Set-Cookie header, 675
HTTP request methods, 381
HTTP_ACCEPT, 433
HTTP_CONNECTION, 433
HTTP_HOST, 433
HTTP_REFERER, 98, 433
HTTP_USER_AGENT, 98, 433
$_HTTP_COOKIE_VARS, 392
httpd.conf, 18fn
$_HTTP_ENV_VARS, 392
$_HTTP_GET_VARS, 392
$_HTTP_POST_VARS, 392
$_HTTP_POST_FILES, 392
$_HTTP_SERVER_VARS, 392

I
Identical operator, arrays, 329–330
Identity test, with strings and numbers, 123
if statement, 46
if/else branching construct, 222–225
if/elseif construct, 47, 225–227
Image button, 410–412
image input type, 389
Image map, 410
implode() function, 179, 215, 275–276, 331, 466–467
in_array() function, 331
include() construct, 100, 373–375, 487–488
include_path directive, php.ini configuration file, 487
Index, 247
Index position of the substring, 189

index.fm Page 865 Wednesday, October 25, 2006 1:28 PM

index.php, 784, 798–799
INFO_ALL, 432
INFO_CONFIGURATION, 432
INFO_CREDITS, 432
INFO_ENVIRONMENT, 432
INFO_GENERAL, 432
INFO_LICENSE, 432
INFO_MODULES, 432
INFO_VARIABLES, 432
Information hiding, 749–750
Ingres, 595
Inheritance, 741, 763–766
  and constructors, 763–766
  defined, 763
Inner joins, 628
INSERT command, 617–618
Instance methods, 53
instanceof (PHP 5) function, 748–749
(int) operator, 119
INT UNSIGNED data type, 622
INTEGER data type, 622
Integers, 60
interface_exists() function, 748–749
Internet Explorer, 14
Internet Information Services (IIS), 14
Interpreter, 15
Invocation, 338
IS [NOT] NULL operator, 608
is_a() function, 748–749
is_array() function, 331
is_dir(), 476
is_file(), 476
is_file() function, 477–478
is_finite() function, 118
is_infinite() function, 118
is_link(), 476
is_nan() function, 118
is_readable() function, 451, 476, 478
is_subclass_of() function, 748–749
is_uploaded_file(), 476
is_writeable(), 476, 479
ISO-8859-1, 213
ISO-8859-15, 213
isset() function, 85–87, 340

J
Java, 41
JavaScript, 64
Join, 628
join() function, 275, 331, 628
Joins, 628

K
key() function, 331
Keys, randomizing, 306
KOI8-R, 214
krsort() function, 298, 334
ksort() function, 298, 305, 334

L
Labels, 228
Left joins, 628
Lerdorf, Rasmus, 4
levenshtein() function, 176–178, 215
Library, 373
Lifetime, 362
LIKE operator, 608, 613
LIMIT clause, 606–607, 619
Line breaks, 23–24
__LINE__, 103
list() function, 331
  and numeric arrays, 286–288
Literals, 62, 64
Local scope, 362–363
Local variables, 84
localeconv function, 215
LOCK_EX, 474
LOCK_NB, 474
LOCK_SH, 474
LOCK_UN, 474
log_errors directive, 841
Logical AND operator, 127, 127–128
  examples, 128
Logical operators, 126–137
  && (and) operator, 127
  || (or) operator, 130–132
  logical XOR operator, 134–135
  NOT (!) operator, 135–137
  testing expressions with, 126–127

index.fm Page 866 Wednesday, October 25, 2006 1:28 PM

Logical OR operator, 130–132
  examples, 130–131
Logical XOR operator, 134–135
  examples, 134
Lookahead, 550–551, 551
Loop control, 51
Looping constructs, 222
Loops, 49–51, 232–243
  and arrays, 266–273
  `break` control statement, 239–243
  `continue` control statement, 239–243
  control statements, 239–243
  `do/while` loop, 234
  `foreach` loop, 239
  `for` loop, 235–238
    and repetitive form fields, 236
  nested loops, 241–243
  `while` loop, 232–233
`LOWER(string)` function, 637
`ltrim()` function, 167fn, 182, 215

M

Magic constants, 102
Magic cookie, 674
`magic_quotes_gpc` directive, 206, 395, 853
`MagicCookie` file, 674
`mail()` function, 811
Markup language, defined, 21
Mask, 173
Math functions, 117–118
`max()` function, 118
`MAX()` function, 634
`mb_convert_case()` function, 165–167
`md5` function, 215
`md5_file` function, 215
Members, classes, 742
  and scope, 750–752
Metacharacters, 517–554, *See also* Anchoring metacharacters
  for alternation, 543
  anchors (assertions), 520–523
  character classes, 525–526
  dot metacharacter, 524–525
  greedy, 533–534
  matching one character from a set, 526–527
  matching one character in a range, 527–528
  matching one character not in a set, 529
  metasymbols, 530–533
  to repeat pattern matches, 533–544
  table of, 518–519
  turning off greed, 541–542
`metaphone()` function, 174–176, 215
Metasymbols, 526, 530–533, 533
  representing alphanumeric word characters, 531–533
  representing digits and spaces, 530–531
`METHOD` attribute, 91
`method` attribute, 389, 421
`__METHOD__`, 103
`method_exists()` function, 748–749
Methods, 52, 338, 740–741, 745, 747
  classes, 742
  defined, 743
Microsoft Internet Information Server, 834
Microsoft SQL Server, 595
`min()` function, 118
`MIN()` function, 634
Missing arguments, 344–345
Mixed data types, and variables, 78–79
`mktime()` function, 689, 829
Modulus operator, 114
`money_format()` function, 162, 216
`MONTHNAME(date)` function, 638
`move_uploaded_file()` function, 425
Multidimensional arrays, 293–297
  finding the size of, 296–297
Multipart MIME message, 810, 814
Multiple alternative delimiters, splitting on, 512–515
Multiple-line comments, 25
Multiplication (*), 112
Multiplier, 181
`mylibrary.php`, 373
MySQL, 3, 7–8, 14, 834
  advantages of, 9–10
  choosing the database, 649–650

index.fm Page 867 Wednesday, October 25, 2006 1:28 PM

client/server model, 7
command-line options, 576–578
commercial license, 569–570
community of developers, 569
connecting to the database server, 647–649
database-driven Web site, example of building, 663–670
  designing the database, 663–664
  example summary, 670
  listing all messages, 669–670
  posting a message page, 665–668
databases, 567–570
defined, 7
developers, 10
documentation, 7
Documentation page, 8
ease of learning, 10
ease of use, 569
executing SQL statements (INSERT, UPDATE, DELETE), 650–652
extensibility, 10
features, 7–8
GPL (GNU General Public License), 9
installing, 9
installing on Linux/UNIX, 854
installing on Mac OS X, 854
installing on Windows, 854
low cost, 10
open source license, 569
open source software, 9
performance, 9
portability, 10
privilege system, 582–593
  creating and dropping a database, 590–591
  “db” table, 586–587
  finding the databases, 583–585
  GRANT command, 589–590
  “host” table, 587–588
  logging into the database server, 582–583
  primary purpose of, 582
  real-world example, 588–589
  REVOKE command, 589–590
  useful functions, 592–593
  “user” table, 585–586
query results, retrieving, 653–657
  mysql_fetch_assoc() function, 655–657
  mysql_fetch_row() function, 653–655
scalability, 570
strengths/weaknesses, 569–570
terminology, 583
where to get, 7
MySQL AB, 7
mysql console, 596
MySQL Query Browser, 580–581, 596, 597, 806
mysql_affected_rows() function, 657
mysql_change_user() function, 657
mysql_connect() function, 657
mysql_errno() function, 658
mysql_error() function, 658–659
mysql_fetch_assoc() function, 657
mysql_fetch_assoc() function, 655–657
mysql_fetch_field() function, 658
mysql_fetch_object() function, 658
mysql_fetch_row() function, 653–655, 657
mysql_field_len() function, 658
mysql_field_name() function, 658, 661–662
mysql_field_seek() function, 658
mysql_field_table() function, 658
mysql_field_type() function, 658
mysql_list_dbs function, 657
mysql_list_fields() function, 658
mysql_list_tables function, 657
mysql_num_fields() function, 658, 661
mysql_num_rows() function, 658, 660
mysql_pconnect() function, 657
mysql_selectdb() function, 657
mysql_tablename() function, 658

N

Naming conventions, functions, 339
Naming sessions, 715–717
natcasesort() function, 298, 334

natsort() function, 298, 334
Needle, 192
Nested function, 368–370
Nested loops, 241–243
Netscape, 14, 674
Network security, and SSL, 833–834
new keyword, 742
Newline (line feed), 24
next() function, 334
nl_langinfo function, 215
nl2br() function, 200–202, 215
Noncapturing parentheses, 519
Nonprogrammer notes, 41–42
NOT (!) operator, 135–137
NOT LIKE operator, 608, 613
! operator, 135–137
Notepad, 15
now() function, 592
NOW() function, 638
Nul-byte, 24
NULL, 68
null() function, 85–86
Null string, 62
Null value, 45
number_format() function, 160, 161–162, 215
Numbers, 43
- comparing, 124
- equality test with, 123
- identity test with, 123

Numeric arrays, 248
- and list() function, 286–288
- shuffling, 309–310

Numeric functions, SQL, 634–636
- GROUP BY clause, 635–636

Numeric literals, 60
Numeric sort, 300–301
Numerically indexed arrays:
- alphabetic sort of, 299–300
- merging, 323–324
- numeric sort, 300–301

NUMERIC(m,d) data type, 622

O

ob_end_flush() function, 691–692
ob_start() function, 691–692
Object-oriented programming (OOP), 739–740
- encapsulation and information hiding, 750

Objects, 52–53, 739–781
- and classes, 740
  - difference between, 742
- defined, 739
- displaying, 747–748
- properties, 740

octdec() function, 118
Offset, 318
Open source license, MySQL, 570
Open Source movement, 3–4
Opening files, 448–450
- mode and permissions, 449–450
- for reading, 451–458
- for writing/appending, 471–475

Operands, 79, 107–108
- defined, 107

Operators, 45, 107–108
- arithmetic, 112–113
- arrays, 327–330
- assignment, 107–108
- associativity, 108–111
- autoincrement and autodecrement, 115–117
- bitwise, 140–143
- casting, 118–120
- comparison, 122–126
- concatenation, 120–121
- conditional, 138–139
- defined, 107
- error control, 143–145
- execution, 143
- logical, 126–137
- precedence, 108–111
- short circuit assignment, 113–114
- SQL, 608
- type, 145

Oracle, 14, 595
ord() function, 206, 215
ORDER BY clause, 616–617, 619
Overriding methods, 766–768

index.fm Page 869 Wednesday, October 25, 2006 1:28 PM

P
pages.php, 494
Parent class, 763
Parent function, 368
Parent–child relationship, 763
Parse errors, 841–843
parse_str function, 215
Passing arguments, 341–351
  default parameters, setting, 347–348
  dynamic function calls, 351
  missing arguments, 344–345
  by reference, 349–351
  by value, 342–344
password() function, 592
password input type, 388
Path, cookies, 676
PATH_TRANSLATED, 434
Pattern, finding, 500–503
  captured patterns, 502–503
  case sensitivity, 501–502
Pattern matching, 499–565
  evaluating the replacement side with the e modifier, 509–510
  functions, 499–517
  modifiers, 505–506
  multiple alternative delimiters, splitting on, 512–515
  preg_grep() function, 515–517
  preg_match() function, 499, 503–504
  preg_match_all() function, 499, 503–504
  preg_split() function, 179, 499, 510–511
    flags for, 511
  searching and replacing, preg_replace_callback() function, 499, 506–509
  searching for patterns in text files, 554–556
Patterns, searching for, in text files, 554–556
PCRE (Perl Compatible Regular Expression) library, 498
Period (.) metacharacter, 518
Perl, 2, 10, 41, 64
Perl style, regular expressions, 498
Permissions, 446
PHP, 2–3, 14
  advantages of, 9–10
  built-in functions, 10, 27–28
  on the command line, 32
  command-line options, 34–35
  comments, 25–27
  compared to HTML, 21–23
  and cookies, 677–694
    $_COOKIE global array, 678–681
    serialization, 681–683
    setcookie() function, 677–678
    tracking visitors, 683–687
  and date/time, 819–832
  defined, 4
  developers, 10
  documentation, 5–6
  download page, 6
  ease of learning, 10
  and e-mail, 809–817
    chunks, 814
    mail() function, 811
    mail server, 809
    MIME (Multipurpose Internet Mail Extensions), 810
    runtime configuration options, 810
    sending a message with an attachment, 814–817
    sending a simple message, 811–812
    sending an HTML message, 813–817
    Simple Mail Transfer Protocol (SMTP) server, 809
  extensibility, 10
  finding the version of, 32
  form validation with, 556–562
  and forms, 390–438
    superglobals for retrieving form data, 392
  as hypertext preprocessor, 15
  installing, 9
  installing on Linux/UNIX, 851
  installing on Mac OS X, 851
  installing on Windows, 850–851
  latest release of, 5
  line breaks, 23–24

index.fm Page 870 Wednesday, October 25, 2006 1:28 PM

PHP *(continued)*:
  low cost, 10
  and MySQL integration, 647–672
  performance, 9
  `php.ini` file, 35–37
  portability, 10
  quick start, 41–57
  running interactively, 34
  script execution at the command line, 32–33
  sessions, 694
  statements, 23–24
  syntax and constructs, 42–56
  testing installation, 852
  where to get, 5–6
  whitespace, 23–24
PHP expressions, 107, *See also* Expressions
PHP operators, 107, *See also* Operators
PHP regular expressions, *See* Regular expressions
PHP script:
  anatomy of, 15–21
  executing, 20
  `.php` extension, 16
  printing strings, 20
  quoting in strings, 19
  script content, 18
  steps of writing, 15
  tags, 16–17
  text editor, finding, 15
  where to save, 18
PHP Security Consortium (PHPSC), 841
PHP superglobal arrays, 431
PHP syntax and constructs:
  arrays, 46
  Boolean values, 44
  classes and objects, 52–54
  comments, 42
  conditional operator, 49
  conditionals, 46–48
  constants, 43
  data types, 43
  files, 54–55
  functions, 52
  loop control, 51
  loops, 49–51
  null value, 45
  numbers, 43
  operators, 45
  predefined variables, 43
  printing output, 42
  quotes, 44
  regular expressions, 55–56
    metacharacters, 56
  strings, 44
  tags, 42
  variables, 43
`PHP_OS`, 103
`PHP_SELF`, 432
`PHP_VERSION`, 103
PHP/FI (Personal Home page/Forms Interpreter), 4
`php.info()` file, 76
`phpinfo()` function, 38, 76, 98–99, 431–432, 498, 852
`php.ini`, 17, 35–37, 76, 90
  configuring, 853–854
phpMyAdmin tool, 579–580
php.net, 5–6
`pi()` function, 118
pico, 15
Portmanteau, defined, 4fn
`pos()` function, 333
Positive lookahead, 550–551
Positive lookbehind, 552
POSIX style, regular expressions, 498
`POST` method, 382–383
`$_POST` array, 403–405
Postincrement operator, 115–116
`pow()` function, 118
Precedence, 108–111
  rules of, for PHP operators, 110
Predefined constants, 102–103
Predefined variables, 43, 98–99
`preg_grep()` function, 499, 515–517
`preg_match()` function, 499, 500–501, 503–504
`preg_match_all()` function, 499, 503–504
`preg_quote()` function, 499
`preg_replace()` function, 499

index.fm Page 871 Wednesday, October 25, 2006 1:28 PM

preg_replace_callback() function, 499, 506–509
preg_split() function, 179, 499, 510–511
  flags for, 511
PREG_SPLIT_DELIM_CAPTURE, 511
PREG_SPLIT_NO_EMPTY, 511
PREG_SPLIT_OFFSET_CAPTURE, 511
Preincrement operator, 115–116
prev() function, 334
Primary keys, 625–626
print() function, 19, 42, 77, 80, 216
print_r() function, 262–264, 330
printf() function, 42, 147, 156–159, 162, 216
  format specifier modifiers, 159
  format specifiers, 158
Printing output, 42
Private scope, 750
Privilege system, MySQL, 582–593
  creating and dropping a database, 590–591
  "db" table, 586–587
  finding the databases, 583–585
  GRANT command, 589–590
  "host" table, 587–588
  logging into the database server, 582–583
  primary purpose of, 582
  real-world example, 588–589
  REVOKE command, 589–590
  useful functions, 592–593
  "user" table, 585–586
Properties, 52, 740, 745, 746
  classes, 742
  objects, 740
property_exists() function, 748–749
Proprietary software, examples of, 3fn
Protected access, 768–771
Protected scope, 751
public keyword, 742
Public scope, 750

Q

Quantifiers, 533
QUARTER(date) function, 638
Queries, 567, 568, 595
Query results, retrieving in SQL, 653–657
  mysql_fetch_assoc() function, 655–657
  mysql_fetch_row() function, 653–655
Query string, 91, 382
QUERY_STRING, 433
quoted_printable_decode function, 216
quotemeta function, 216
Quotes, 44, 609
  strings, 150–152

R

radio input type, 389
rand(start,finish) function, 118
range() function, 251, 253–255, 330
readfile() function, 470
Reals, 60
Records, 601
Recursion, 296
Recursive functions, 370–372
Redirection, using the correct URI for, 418–420
register_globals directive, 90, 391, 835–837, 853
Regular expressions, 55–56, 497, 497–565, *See also* Pattern matching
  commenting, and the x modifier, 553–554
  defined, 497–498
  grouping/clustering, 544–545
  metacharacters, 56, 517–554
  Perl style, 498
  positive lookahead, 550–551
  positive lookbehind, 552
  POSIX style, 498
  regexlib.com, 562–564
  remembering/capturing, 545–547
  searching, capturing, and replacing, 548–550
Relation, 567
Relational databases, 570–575
  columns/fields, 573
  database schema, 575
  databases, defined, 571
  fields, 572–573
  indexes, 574
  primary key, 574

index.fm Page 872 Wednesday, October 25, 2006 1:28 PM

Relational databases *(continued)*:
  records, 572–573
  rows/records, 574
  tables, 571–572
Relations, 567, 628–631
  aliases, 630–631
  two tables with a common key, 628–629
  using a fully qualified name and a dot to join tables, 629–630
Remembering/capturing, 545–547
REMOTE ADDRESS, 98
REMOTE_ADDR, 433
REMOTE_HOST, 433
REMOTE_PORT, 433
rename() function, 481–482
Repeating characters, matching for, 540–541
request method, 393
$_REQUEST[] superglobal array, 836
REQUEST_METHOD, 433
REQUEST_TIME, 433
REQUEST_URI, 434
Request/response loop, 14
require() construct, 373–375, 487–488
Reserved words, SQL, 599
reset() function, 284, 334
reset input type, 389
Resources, 68
Result-set, 604
Return values, 27, 351–358
  conditional returns, 353–355
  returning a single value, 352–353
  returning multiple values, 355–358
Reusing classes, 776–780
Reversed sort, 302
rewind() function, 459
round() function, 27, 118
rsort() function, 298, 302, 334
rtrim() function, 167fn, 182, 216
Runtime configuration, sessions, 730–731

S
Sambar, 14
Scalability, MySQL, 570
Scalar variable, 247
  defined, 247
Scalars, 59
Scope, 362–367
  $GLOBALS[] array, 365
  global, 363–364
  local, 362–363
  private, 750
  protected, 751
  public, 750
  static variables, 365–367
SCRIPT_FILENAME, 434
SCRIPT_NAME, 434
Secure Sockets Layer (SSL), 833–834
Security, 833–841
  cookies, 677
  defined, 833
  error reporting settings, 839–840
  file names, 838
  file system, 839
  form field validation, 835
  logging everything, 839
  network security and SSL, 833–834
  operating system, web server, and file system security, 834–835
  PHP and MySQL, 834–835
  PHP Security Consortium (PHPSC), 841
  register_globals directive, 835–837
  safe mode, 839
  SQL injections, 837
seek() function, 459
SELECT * FROM command, 601
SELECT command, 604–617
  < operator, 611–612
  <>, != operator, 608–609
  = operator, 608–609
  > operator, 611–612
  LIKE operator, 608, 613
  LIMIT clause, 606–607
  NOT LIKE operator, 608, 613
  NULL, 609–611
  AND operator, 612–613
  BETWEEN operator, 615
  ORDER BY clause, 616–617
  pattern matching and the % wildcard, 613–614
  quotes, 609

index.fm Page 873 Wednesday, October 25, 2006 1:28 PM

SELECT DISTINCT statement, 605–606
selecting all columns, 605
selecting specified columns, 604
WHERE clause, 608, 608–609
SELECT DISTINCT statement, 605–606
select input type, 389, 406–407
Selection criteria, 608
Self-processing HTML forms, 413–420
checking for form submission, 413–414
hidden fields, using, 415–417
redirecting the user, 417–420
location header, 417–418
using the correct URL for redirection, 418–420
SEQUEL (Structured English Query Language), 595
Serialization, 681–683
serialize() function, 681–682
SERVER_NAME, 433
SERVER_PORT, 434
SERVER_PROTOCOL, 433
SERVER_SIGNATURE, 434
SERVER_SOFTWARE, 433
$_SERVER array, 432–436
Session ID number, 695
session_cache_expire() function, 730
session_cache_limiter() function, 730
session_commit() function, 730
session_decode() function, 730
session_destroy() function, 727–729, 730
session_encode() function, 730
session_get_cookie_params() function, 730
session_id() function, 730
session_is_registered() function, 702, 730
session_module_name() function, 731
session_name() function, 715, 731
session_regenerate_id() function, 731
session_register() function, 702, 731
session_save_path() function, 731
session_set_cookie_params() function, 731
session_set_save_handler() function, 731
session_start() function, 698–699, 715, 731
session_unregister() function, 702, 731
session_unset() function, 731
session_write_close() function, 728, 731
$_SESSION array, 702–706
session.cookie_domain directive, 707
session.cookie_lifetime directive, 707
session.cookie_path directive, 707
session.cookie_secure directive, 707
Sessions, 673, 694–736
defined, 694
ending, 727–730
garbage collection, 729–730
hidden form element, 717–720
implementing a login system with, 731–736
naming, 715–717
registering, 699–702
runtime configuration, 730–731
saving arrays in, 702–706
session cookie functions and configuration options, 706–707
session files, cleaning up, 729–730
session IDs:
changing, 726–727
passing with a link, 721–726
session variables, deleting, 727–728
session_register() function, 702
session_start() function, 698–699
setting preferences with, 708–715
starting a cookie-based session, 698–699
where to store, 696–698
without cookies, 717–720
session.save_path() function, 696
session.use_cookies directive, 707
session.use_only_cookies directive, 707
__set() method, 753, 758
setcookie() function, 677–678, 681–682, 727
setlocale function, 216

Setters, 758–760
sha1 function, 216
sha1_file function, 216
Shell terminal, 33
shell_exec() function, 143
Shift_JIS, 214
Short circuit assignment operators, 113–114
short_open_tag configuration option, 853
SHOW FIELDS IN command, 602–603
SHOW TABLES IN command, 601
ShowDate.php, 33
shuffle() function, 298, 309–310, 332, 335
similar_text function, 216
similar_text() function, 176–178
Simple Mail Transfer Protocol (SMTP) server, 809
Single quotes, 150, 352–353, 609
Single-line comment, 25
sizeof() function, 277–279, 333
Slashes, stripping out, 395–396
SMALLINT data type, 622
Smarty, 487
SMTP configuration option, 854
sort() function, 298, 298–301, 335
SORT_LOCALE_STRING flag, 299
SORT_NUMERIC flag, 299
SORT_REGULAR flag, 299
SORT_STRING flag, 299
Sorting arrays, 298–306
soundex() function, 174–176, 216
Space, 24
Special characters:
  HTML, 210–214
  strings and, 200–210
Special data types, 68–70
Splicing arrays, 318–321
split() function, 179, 275, 331, 499, 510
spliti() function, 179, 331, 499, 510
sprintf() function, 20, 42, 159–160, 216
SQL (Structured Query Language), 568–569 *See also* Data Definition Language (DDL); Data Manipulation Language (DML):
  commands/queries, 597–600
    case sensitivity, 599–600
    English-like grammar, 597–598
    naming conventions, 599
    reserved words, 599
    result-set, 600
    semicolons, use of, 598
  Data Definition Language (DDL), 620–633
  Data Manipulation Language (DML), 603–620
  data types, 621–622
  and the database:
    show databases command, 600–601
    USE command, 601
  database tables, 601–603
    DESCRIBE command, 602–603, 664
    SHOW FIELDS IN command, 602–603
  defined, 595
  executing SQL statements, 596–597
  functions, 633–642
    date and time, 638–642
    numeric, 634–636
    string, 637–638
  operators, 608
  standardization of, 596
  tutorial, 595–646
SQL errors, fixing, 845–847
sqrt() function, 118
srand() function, 118
sscanf() function, 216
Stale cookie, 676
Statements, 23–24
Static members, 773–774
Static variables, 365–367
Static Web sites, 1–2
Sticky forms, 427–431
str_pad() function, 184–185, 216
str_repeat() function, 181, 216
str_replace() function, 186, 216
str_rot13() function, 216
str_shuffle() function, 216
str_split() function, 179, 216
str_word_count() function, 164, 216
strcasecmp() function, 216
strchr() function, 191–193, 216
strcmp() function, 168–169
strcmp() function, 216

index.fm Page 875 Wednesday, October 25, 2006 1:28 PM

strcoll() function, 216
strcspn() function, 216
strfmon() function, 162
strftime() function, 822–824
stri_ireplace() function, 187, 216
stri_replace() function, 185–189
strichr() function, 191–193
String constant, 62
String functions, 155–164
  addcslashes() function, 206–208
  addslashes() function, 206–208
  ASCII character values:
    and the chr() function, 205
    and the ord() function, 206
  changing the case of strings, 164–167
  chop function, 182
  comparing strings, 167–174
  curly brace syntax, 200
  explode() function, 179
  finding a position in a string, 189–191
  finding similarities, 174–178
  finding the length of a string, 163
  formatting and printing strings, 155–160
  formatting numbers and money, 160–162
  fprintf() function, 160
  HTML special characters, 210–214
  htmlentities(), 210, 213–214
  htmlspecialchars() function, 210–214
  htmlspecialchars_decode(), 210–214
  implode() function, 179
  levenshtein() function, 176–178
  ltrim() function, 182
  mb_convert_case() function, 165–167
  metaphone() function, 174–176
  money_format() function, 162
  nl2br() function, 200–202
  number_format() function, 161–162
  preg_split() function, 179
  printf() function, 156–159
    format specifier modifiers, 159
    format specifiers, 158
  quote constants, 211
  repeating a string, 181
  rtrim() function, 182
  search and replace, 185–189
  similar_text() function, 176–178
  soundex() function, 174–176
  special characters, 200–210
  split() function, 179
  spliti() function, 179
  splitting up a string, 179–181
  sprintf() function, 159–160
  SQL, 637–638
  str_pad() function, 184–185
  str_repeat() function, 181
  str_replace() function, 185–189
  str_split() function, 179
  str_word_count() function, 164
  strcasecmp() function, 168, 170–172
  strcmp() function, 168–169, 216
  strcspn() function, 169
  strcspn() function, 174
  stri_ireplace() function, 187
  stripslashes() function, 208, 217
  strlen() function, 163
  strnatcasecmp(str1, str2) function, 171–172
  strnatcmp(str1, str2) function, 169
  strncasecomp() function, 169, 170
  strpos() function, 189–190
  strripos() function, 191
  strrpos() function, 191
  strspn() function, 169, 173
  strstr() function, 206
  strtok() function, 179–181
  strtolower() function, 165–166
  strtoupper() function, 165–166
  substrings, 191–199
  trim() function, 182
  trimming and padding strings, 181–185
  ucfirst() function, 165–166
  ucwords() function, 165–166
  wordwrap() function, 200–205, 218
String literals and quoting, 62–67
(string) operator, 119
String operators, 152–155
  concatenation, 152–153
  equality (==) operator, 154
  functions, 155–164
  identical (===) operator, 154

Strings, 19, 44, 149–220, *See also* String functions; Substrings
  comparing, 125–126
  defined, 149
  equality test with, 123
  identity test with, 123
  quotes, 150–152
    quoting errors, 151–152
    single quotes, 150
    string operators, 152–155
  special characters and, 200–210
`strip_tags()` function, 216, 453
`stripcslashes()` function, 208, 217
`stripos()` function, 217
`stripslashes()` function, 208, 217, 395
`stristr()` function, 217
`strlen()` function, 163, 217
`strnatcasecap()` function, 217
`strnatcmp()` function, 217
`strncasecmp()` function, 217
`strncmp()` function, 217
"Strongly typed" languages, 71
`strpbrk()` function, 217
`strpos()` function, 189–190, 217
`strrchr()` function, 191–193
`strripos()` function, 191
`strrpos()` function, 191, 217
`strspm()` function, 217
`strstr()` function, 217
`strtok()` function, 179–181
`strtolower()` function, 165–166, 217
`strtotok()` function, 217
`strtoupper()` function, 165–166, 217
`strtr()` function, 217
Structured Query Language (SQL), *See* SQL
`style.css`, 784
Subclasses, 763
Subject, 500
`submit` input type, 389
Subpatterns, 502
`substr()` function, 191, 193–194, 218
`substr_compare()` function, 217
`substr_count()` function, 191, 198–199, 217
`substr_replace()` function, 191, 196–198, 217
Substrings:
  defined, 191
  functions, 191–199
  `strchr()` function, 191–193
  `strichr()` function, 191–193
  `strrchr()` function, 191–193
  `substr()` function, 191, 193–194
  `substr_count()` function, 191, 198–199
  `substr_replace()` function, 191, 196–198
`SUBSTRING(string, position)` function, 637
Subtraction (–), 112
`SUM()` function, 634
Super class, 763
Superglobal arrays, 43
  PHP, 392
  where to get information about, 431–432
    `phpinfo()` function, 431–432
Superglobal variables, 84
Superglobals, 43, 431
Superuser, 582
Suraski, Zeev, 4
`switch` statement, 48, 228–231
Sybase, 14, 595
Syntax, 842
Syntax errors, 143
System/R, 595

T
Tab, 24
Table, 567
  altering, 631–632
  dropping, 633
Tags, 42
Ternary operator, 49, 107, 138
`TEXT` data type, 622
Text files, searching for patterns in, 554–556
`text` input type, 388
`textarea` input type, 388
TextPad, 15
Thawte, 834

index.fm Page 877 Wednesday, October 25, 2006 1:28 PM

thinkgeek.com, 598
$this pseudo-variable, 743
TIME data type, 622
time() function, 688–689
TIME_FORMAT() function, 639–641
TIMESTAMP data type, 622
TINYINT data type, 622
TINYTEXT data type, 622
Tokens, 179
touch() function, 481
Tracking visitors, with cookies, 683–687
  tracking a visitor's last visit, 685–687
  visitor count example, 683–685
TRANSLATE function, 637
trim() function, 167fn, 182, 218
TRIM(' string '); function, 637
TRUE, 103
Type operators, 145

U
uasort() function, 298, 335
ucfirst() function, 165–166, 218
ucwords() function, 165–166, 218
uksort() function, 298, 335
Unary operator, 107, 135
  defined, 107
Union operator, arrays, 327–328
UNIX timestamp, 688, 819
  creating from a string, 828–829
unlink() function, 481–483
unserialize() function, 681
unset() function, 45, 89–90, 311, 332
UPDATE command, 618–619
upload_max_filesize, 421
upload_tmp_dir directive, 422
Uploading files, 421–427
  $_FILES array, 422–424
  <form> tag attributes, 421–422
  moving the uploaded file, 425–426
UPPER(string) function, 637
URL, opening for reading, 463–464
user() function, 592
User-defined functions, 337–377
  callback functions, 358–362
  declaring, 338
  function libraries, 373–376
  nesting functions, 368–370
  return values, 351–358
  scope, 362–367
  where to put, 338
User-defined setters and getters, 760–761
usort() function, 298, 335
UTF-8, 213

V
var keyword, 742
var_dump() function, 265–266, 330
VARCHAR(SIZE) data type, 622
Variable variables, 351
Variables, 43, 70–99
  and concatenation, 79–80
  declaring/initializing, 71–72
  defined, 70
  displaying, 75–78
  double, single, and backquotes in assignment statements, 72–73
  empty() function, 88–89
  environment, 84
  form, 90–99
  global, 84
  isset() function, 85–87
  local, 84
  managing, 85–90
  and mixed data types, 78–79
  print and echo constructs, 75–76
  references, 80–81
  scope of, 84
  shortcut tags, 76–78
  superglobal, 84
  types of, 70
  unset() function, 89–90
  valid names, 71
  variable variables, 82–83
VeriSign, 834
version() function, 592
Vertical tab, 24
vfprintf function, 218
vi, 15

Visual Basic, 41
vprintf function, 218
vsprintf function, 218

W

Web page:
  analysis of, 14–15
  life cycle of, 13
Web servers, 446, 849
WEEKDAY(date) function, 638
WHERE clause, 608–609, 619, 630
while loop, 49, 232–233
Whitespace, 23–24
Widenius, Michael (Monty), 7
WinEdit, 15
Wordpad, 15
wordwrap() function, 200–205, 218
World Wide Web (WWW), invention of, 1–2

X–Z

XOR (exclusive OR) operator, 608
Yahoo!, 2, 7
YEAR data type, 622
YEAR(date) function, 638
Zend engine, 4–5, 9
Zend Technologies, 4
Zero-width assertions, 520