

# Index

## SYMBOLS AND NUMBERS

### { } (braces)

indentation standards for, 214–217  
reducing nesting levels, 217–220

### : (colon)

with class constants (: :), 28–29  
after scheme name in PHP URLs (: //), 72  
with `self` keyword (: :), 7, 27–28

### \$ (dollar sign)

not used with class constants (PHP), 29  
`$this` reference variable (PHP), 23, 50

### :: (double colon). See : (colon)

### = (equals sign) in equality versus assignment operators, 219

### 404 handler, caching and, 226

### “ (quotation marks)

for function names within `set_exception_handler()`, 5  
speed of strings and, 201–205

### ‘ (single quotes), speed of strings and, 201–205

### / (slash)

/G terminator for MySQL queries, 86  
after scheme name in PHP URLs (: //), 72

## A

ab **ApacheBench** program, 208–209

`abort()` method (AJAX), 277

### abstract classes (PHP)

`abstract` keyword for, 7  
example, 7–8  
language translation, 341–342  
naming conventions, 222  
overview, 34–35  
rules for using, 8

### abstract keyword (PHP), 7

**access control (Apache). See authentication**

### access control (MySQL)

alternatives to `GRANT` and `REVOKE`, 97–100  
`columns_priv` table for, 99, 100  
`db` table for, 98–99, 100  
`GRANT` command for, 94–96  
reloading privilege tables, 100  
`REVOKE` command for, 96–97  
server restriction, 100  
`tables_priv` table for, 99, 100  
user administration, 93–100  
`user` table for, 97–98, 99–100

### access control (PHP)

Basic Authentication, 166–171  
CAPTCHA images in registration forms for, 175–177

### **access control (PHP) (continued)**

- forms authentication, 172–174
- HTTP\_REFERER value and, 174–175
- REMOTE\_ADDR value and, 174
- server variables and, 174–175

### **access modifiers (PHP). See also specific modifiers**

- Circle class example, 24–25
- language translation, 337
- PHP5 enhancements, 6
- visibility specified by, 23–24

### **addslashes() function (PHP), 113**

### **AJAX (Asynchronous JavaScript and XML)**

- accessibility concerns, 283–285
- CPAINT library, 283
- described, 273
- DOM manipulations, 273, 281–282
- further information, 285
- Google's GMail application, 273
- handling ReadyStateChange event, 280
- handling the response, 280–282
- history, 274–277
- JSPAN library, 283
- methods, 277–278
- properties, 278–279
- SAJAX library, 282–283
- support problems, 285
- traditional web applications versus, 273, 274
- usability concerns, 283–285
- working with browser differences, 279–280
- XMLHTTP and XMLHttpRequest interfaces, 277–282

### **Alcantara, Frank (eAccelerator maintainer), 290**

### **Allman indentation style, 215–216**

### **alpha transparency, 244**

### **ALTER TABLE statement (MySQL), 87, 90–91, 92**

### **Alternative PHP Cache. See APC**

### **analyzing databases (MySQL), 100–102**

### **anonymous functions (PHP), 57–58**

### **Apache. See also URL rewriting (Apache)**

- authentication with MySQL, 129, 149–154, 166
- Basic Authentication, 154, 165–166

- content compression, 145–149
- 404 handler and caching, 226
- mod\_alias, 226
- mod\_auth\_mysql, 129, 149, 150–154
- mod\_dav, 129, 158
- mod\_deflate, 129, 146–148
- mod\_proxy, 138
- mod\_rewrite, 226
- mod\_speling, 129, 144–145
- mod\_ssl, 129, 154–155
- as online file repository, 158–163
- PDFLib configuration and, 234
- performance improvements, 211–213
- SSL with, 154–157
- URL spell checking, 144–145

### **ApacheBench program (ab), 208–209**

### **APC (Alternative PHP Cache)**

- configuring, 288–290
- history, 287
- installing, 288
- other caching engines and, 305
- overview, 287
- removing, 290
- situations suitable for, 305
- website, 288

### **array functions (PHP). See also callbacks (PHP)**

- array\_filter(), 53
- array\_map(), 49, 50–51
- array\_walk(), 49, 51–52
- callbacks provided for most, 48
- call\_user\_func\_array(), 55–56
- described, 48
- memory requirements for series of, 59–60
- preg\_grep(), 53
- preg\_replace\_callback(), 53–55
- using callbacks with, 48–50
- variety and number of, 59

### **array\_filter() function (PHP), 53**

### **ArrayIterator class (PHP), 6**

### **array\_map() function (PHP), 49, 50–51, 55–56**

**arrays (PHP). See also array functions (PHP)**

- checking size of, performance and, 229
- language translation, 331
- naming conventions, 222
- scratch variables and performance, 229–230

**array\_walk() function (PHP), 49, 51–52****ASP. See language translation****ASP.NET. See language translation****ASP-style tags, `php.ini` settings for, 106–107****assignment by value versus by reference (PHP), 29****Asynchronous JavaScript and XML. See AJAX****attempt\_auth() function (PHP), 169–170****authentication**

- with Apache and MySQL, 149–154
- with Apache and SSL, 154–157
- Apache Basic Authentication, 154, 165–166
- Auth\_HTTP PEAR package for, 187–188
- PHP-controlled, 166–177
- with WebDAV, 159

**Auth\_HTTP PEAR package, 187–188****AUTO\_INCREMENT columns (MySQL), 89****B****backups (MySQL), 103–104****Basic Authentication (Apache)**

- implementing, 165
- limitations, 154
- with MySQL, 149–154, 166
- PHP-controlled authentication and, 166, 171
- situations not suitable for, 165–166

**Basic Authentication (PHP)**

- Apache-controlled authentication and, 166, 171
- creating the authentication database, 166–167
- `phpauthbasic.php` script for, 167–171
- testing, 171

**benchmarking**

- ab ApacheBench program, 208–209
- defined, 206–207
- MyBench MySQL tool, 350
- PEAR Benchmarking class, 207–208

- results varying for different runs, 202
- speed of strings example, 201–205

**Benchmarking class (PEAR), 207–208****blog Micro CMSs**

- popularity of blogging, 322
- reasons for blogging, 322
- Serendipity, 324
- WordPress, 323

**braces ( { })**

- indentation standards for, 214–217
- reducing nesting levels, 217–220

**BSD indentation style, 215–216****Bzip2 extension (PHP), 271****C****C#. See language translation****caching (PHP)**

- APC caching engine, 287–290
- choosing a caching engine, 305
- client-side, 227
- CMS provisions for, 225–226
- eAccelerator caching engine, 290–294
- `Expires:` header and, 227
- JPCache caching engine, 296–300
- memcached caching engine, 300–304
- output pre-generation, 224–226
- PEAR packages for, 224–225
- performance advantages of, 223
- source pre-parsing, 223–224
- using 404 handler for, 226
- using different caching engines together, 304–305
- Zend Optimizer caching engine, 294–296

**calcArea() method, 21****\_\_call() magic method (PHP), 8–9, 38****callbacks (PHP)**

- for abstracting functionality, 48
- for `array_filter()` function, 53
- for `array_map` function, null callback, 50–51
- for `array_walk()` function, 52
- for `call_user_func_array()` function, 55–56

### callbacks (PHP) (continued)

`create_function()` function and, 57–58  
new implementations likely for, 59  
`preg_replace_callback()` function, 53–55  
provided for most array functions, 48  
`self` keyword with, 50  
specifying class and object methods, 49–50  
strings containing the name of a function, 49  
`$this` reference variable with, 50  
using, 48–50

### calling methods and functions

`__construct()` magic method for, 6  
language translation, 338–339, 341  
overloading with magic methods, 8–9  
procedurally (MySQLi), 13

`call_user_func()` **function, 56–57**

`call_user_func_array()` **function, 55–56**

**CAPTCHA (Completely Automatic Public Turing Test to Tell Computers and Humans Apart), 175–177**

`caption()` **method (WebImage class), 251–252**

**CASCADE option (MySQL), 92**

`case` **statements, language translation for, 332–333**

**CERT website, 182**

`check_login()` **function (PHP), 170–171**

**Circle class example**

abstract `Shape` class and, 34–35  
`__call()` magic method with, 38  
`clone` statement with, 29–30  
constructor, 25–26  
defining, 20  
destructor, 26–27  
`__get()` and `__set()` magic methods with, 39  
inheritance from `Shape` class, 31–33  
for memcached caching engine, 303–304  
methods, 21  
`parent::` references, 34  
properties, 21  
`__sleep()` and `__wakeup()` magic methods with, 40–43

`static` keyword with, 27–28  
`__toString()` magic method with, 44  
visibility specified in, 24–25

**class constants (PHP), 28–29**

**classes (MySQLi), 13–16**

**classes (PHP). See also specific classes**

abstract, 7–8, 34–35, 222, 341–342

basics, 20–23

class constants, 28–29

described, 20

`final` keyword with, 6

instantiating, 22–23

language translation, 337–342

naming conventions, 222

**CLI, installing PEAR using, 186**

**Client URL Library Functions (CURL) extension (PHP), 271**

**client-side caching, 227**

`clone` **statement (PHP), 29–30**

**closing streams (PHP), 63**

**CMSMatrix website, 328**

**CMSReview website, 327**

**CMSs (Content Management Systems)**

all-inclusive Web CMSs, 309–322

blog packages, 322–324

caching provided by, 225–226

described, 307

DocuWiki, 326

Drupal, 321

enterprise CMSs, 307–308

ExponentCMS, 309–312

Mambo, 321

MediaWiki, 325–326

Micro CMSs, 322–328

open source Web CMS packages, 308

PHP-Nuke, 321

phpWebsite, 315–318

Postnuke, 322

resources, 327–328

Serendipity, 324

TikiWiki, 318–321

Web CMSs, described, 308

wiki packages, 324–326

- WordPress, 323
- XOOPS, 312–315
- CMSWire website, 328**
- code efficiency. See performance**
- coding standards (PHP)**
  - braces and indentation, 214–217
  - comments, 216, 217, 220
  - enforcing with parse errors, 222–223
  - naming conventions, 220–222
  - overview, 213–214
  - PEAR, 214
  - reducing nesting levels, 217–220
- colon (:)**
  - with class constants (: :), 28–29
  - after scheme name in PHP URLs (: //), 72
  - with self keyword (: :), 7, 27–28
- columns\_priv table (MySQL), 99, 100**
- comments**
  - Big Flashy Box style, 216, 217
  - coding standards for, 220
  - Javadoc-style, 220
  - language translation, 329
  - tab characters for indentation and, 216, 217
- Completely Automatic Public Turing Test to Tell Computers and Humans Apart (CAPTCHA), 175–177**
- compressing content. See content compression (Apache)**
- concatenation, benchmarking speed of, 201–205**
- conditional statements**
  - language translation, 331–336
  - performance and, 230
- config-help command (PEAR manager), 186**
- ConfigManager.php class**
  - code listing, 117
  - configfile property, 118
  - configtype property, 118
  - constructor method, 118
  - declaration, 117
  - errors property, 118
  - processConfig method, 118–119
- ConfigProcessor interface (PHP), 119**
  - config-set command (PEAR manager), 186**
  - config-show command (PEAR manager), 185–186**
  - configuration checker (PHP)**
    - ConfigManager.php class, 117–119
    - ConfigProcessor interface, 119
    - reqs.xml file, 116
    - test.php code, 128
    - XMLConfigProcessor.php class, 119–128
  - configuration variables (PEAR), 185**
  - configuring PHP. See php.ini configuration file; runtime configuration (PHP)**
  - const keyword (PHP), 29**
  - constants**
    - class constants (PHP), 28–29
    - language translation, 330
    - naming conventions, 222
  - \_\_construct() magic method (PHP)**
    - Circle class example, 25–26
    - ConfigManager.php class, 118
    - described, 37
    - overview, 6, 25–26
    - PHP4 constructors versus, 25
    - XMLConfigProcessor.php class, 122–123
  - constructors (PHP)**
    - Circle class example, 25–26
    - \_\_construct() magic method for, 6, 25–26
    - language translation, 339
    - overview, 25–26
    - PHP4 constructors, 25
  - content compression (Apache)**
    - configuration directives, 147–148
    - described, 145–146
    - effectiveness of mod\_deflate, 148
    - httpd.conf file for, 146–147
    - need for, 145
    - for PHP scripts, 148–149
    - using mod\_deflate, 146–148
  - Content Management Systems. See CMSs**
  - converting tables**
    - InnoDB to MyISAM, 93
    - MyISAM to InnoDB, 90

### coordinate systems

- Ming extension (PHP), 264
- PDFLib extension (PHP), 237
- Coppermine CMS, 327**
- copying objects with `clone` statement (PHP), 29–30**
- Cowgill, Daniel (APC creator), 287**
- CPAINT (Cross-Platform Asynchronous Interface Toolkit), 283**
- CREATE DATABASE statement (MySQL), 75**
- CREATE TABLE statement (MySQL)**
  - for Apache authentication example, 149–150
  - enabling full-text searching, 86–87
  - overview, 75
  - for PHP authentication example, 166–167
- `create_function()` function, 57–58**
- cross joins (MySQL), 81, 82**
- Cross-Platform Asynchronous Interface Toolkit (CPAINT), 283**
- cross-site scripting (XSS), 180–181**
- CSS stylesheets**
  - AJAX use of, 273
  - PEAR packages for, 192–193
- CURL (Client URL Library Functions) extension (PHP), 271**
- CVE website, 182**
- CVS version control system, 350–351**

## D

- Data Objects, PDO PECL package for integrating with, 196–197**
- database analysis (MySQL), 100–102**
- database maintenance (MySQL), 103–104**
- database-related packages (PEAR)**
  - `Auth_HTTP`, 187–188
  - `DB`, 190–192
  - `HTML_Table`, 194–195
  - `PEAR::Auth`, 187, 188
  - `PEAR::DB`, 187
  - website for, 191
- databases (MySQL). See MySQL**
- Date PEAR package, 189–190**
- DB PEAR package, 190–192**
- db table (MySQL), 98–99, 100**
- debugging scripts, Xdebug PECL package for, 197**
- `DeflateCompressionLevel` directive (`mod_deflate` of Apache), 147**
- `DeflateFilterNote` directive (`mod_deflate` of Apache), 148**
- `DeflateWindowSize` directive (`mod_deflate` of Apache), 148**
- DELETE statement (MySQL), 77, 92**
- deleting database information (MySQL), 77, 92**
- `__destruct()` magic method (PHP), 6, 26–27, 37**
- destructors (PHP)**
  - `Circle` class example, 26–27
  - `__destruct()` magic method for, 6, 26–27
  - language translation, 339–340
  - overview, 26–27
- directives (PHP). See `php.ini` configuration file**
- directories**
  - automatically redirecting URL to, 132
  - creating for SSL website, 157
  - pattern matching reads with `glob()`, 60–61
  - for WebDAV file repository, 158
- `DirectoryIterator` class (PHP), 5**
- disk access, performance and, 211**
- `display()` method (`WebImage` class), 244**
- DocuWiki Micro CMS, 326**
- dollar sign (\$)**
  - not used with class constants (PHP), 29
  - `$this` reference variable (PHP), 23, 50
- DOM (Document Object Model)**
  - AJAX manipulation of, 273, 281–282
  - DOM extension (PHP), 16, 271
- double colon (::). See colon (:)**
- `do...while` loops, language translation for, 334**
- `drawHR()` function (PDFLib), 236**
- Drupal CMS, 321**

**DSO (dynamic shared objects)**

installing APC as, 287, 288  
 installing eAccelerator as, 291  
 installing memcached as, 301  
 PDFLib configuration and, 234

**E****eAccelerator caching engine**

configuring, 291–294  
 installing, 291  
 other caching engines and, 305  
 overview, 290–291  
 removing, 294  
 website, 291

**echo statement (PHP)**

print statement versus, 228–229  
 \_\_toString() magic method with, 44

**emailException class, 3–4****enterprise CMSs, 307–308. See also CMSs (Content Management Systems)****equals sign (=) in equality versus assignment operators, 219****error messages, php.ini settings for, 106****Exception class (PHP), 2–4****exceptions (PHP)**

described, 1  
 Exception class for, 2–4  
 language translation, 344–345  
 PHP5 changes and enhancements, 1–5  
 set\_error\_handler() function versus, 1–2  
 try/catch/throw blocks for,  
 2, 234, 344–345

**Expires: header, caching and, 227****ExponentCMS**

additional modules, 310  
 customizing default settings, 310–311  
 default features, 309  
 “inactive” features, 309  
 installing, 309–310  
 strengths and weaknesses, 311  
 websites, 309, 310

**extends keyword (PHP)**

emailException example, 3–4  
 inheritance and, 31–33  
 interfaces and, 36–37  
 overview, 3, 31–33

**eXtensible Object Oriented Portal System.****See XOOPS****extension\_loaded() function (PHP)**

overview, 111  
 in XMLConfigProcessor.php class,  
 124, 125

**external files, language translation for including, 343–344****F****fat-client web applications. See AJAX (Asynchronous JavaScript and XML)****file access logger example**

code listing, 64–66  
 registering stream wrapper, 66  
 stream\_close() function, 67  
 stream\_open() function, 67  
 stream\_read() function, 67  
 stream\_write() function, 67  
 using logged streams, 66–67

**file repository using Apache. See WebDAV (Web-based Distributed Authoring and Versioning)****Fileinfo PECL package, 196****final keyword (PHP), 6, 33****Flash files (Shockwave). See Ming extension (PHP)****FLUSH PRIVILEGES command (MySQL), 100****for loops**

array functions not requiring, 48  
 language translation, 335

**foreach loops**

array functions not requiring, 48  
 array\_walk() function for avoiding,  
 49, 51–52  
 language translation, 335

### foreign key constraints (MySQL)

- advantages, 89
- creating with `ALTER_TABLE` statement, 90–91, 92
- naming, 90
- `NULL` values and, 89
- options for specifying behaviors, 92
- `UPDATE` statements and, 91–93

### forms (HTML), PEAR package for, 193–194

#### forms (PHP)

- for login authentication, 172–174
- `php.ini` settings for, 105, 107

### 404 handler, caching and, 226

### frames (Ming), 264

### FTP, installing PEAR using, 186

### full-text searching (MySQL)

- altering existing table to support, 87
- creating new table enabling, 86–87
- limitations, 88
- minimum full-text word length specification, 88
- not supported by InnoDB, 89
- `SELECT` statement for, 87–88

### functions (MySQLi), calling procedurally, 13

#### functions (PHP). *See also array functions (PHP); specific functions*

- anonymous, 57–58
- `ArrayIterator` class, 6
- callbacks and, 49
- checking extensions and their properties, 111–112
- checking installed version, 108
- checking magic quotes settings, 113
- `Exception` class, 3
- `Iterator` class, 5
- language translation, 336–337
- naming conventions, 220, 221–222
- new in PHP5, 91–92
- retrieving configuration settings at runtime, 109–110
- variety and number of, 47

### FusionNews CMS, 327

## G

### /G terminator for MySQL queries, 86

#### GD library extension (PHP)

- adding captions, 250–254
- adding logos, 254–257
- alpha transparency and, 244
- creating the image base, 241–245
- described, 241
- `ImageCaption` class, 250–254
- `ImageLogo` class, 254–257
- `ImageResize` class, 245–248
- `ImageRotate` class, 249–250
- index transparency and, 244
- manual for, 263
- resizing images, 245–248
- rotating images, 248–250
- testing, 257–261
- `WebImage` base class for, 241–245, 246–247, 249, 251–252
- `WebImage` class, 255–256

#### \_\_get() magic method (PHP)

- `Circle` class example, 39
- example, 8
- overview, 38–39

#### get\_cfg\_var() function (PHP), 109

#### get\_defined\_functions() function, 57

#### get\_extension\_funcs() function (PHP), 112

#### get\_loaded\_extensions() function (PHP), 111–112, 116

#### get\_magic\_quotes\_gpc() function (PHP), 113

#### get\_magic\_quotes\_runtime() function (PHP), 113

#### getMessage() function (PHP), 2, 3

#### getResponseHeader() method (AJAX), 277

#### GIS photographic database query example

- approaches for solutions, 67–68
- code listing, 68–70
- nzmng protocol, 70
- PNG files for images, 70, 71–72

`stream_open()` function, 70–71

`stream_read()` function, 71

`stream_tell()` function, 71

website for the database, 67

`glob()` **function (PHP), 60–61**

**GMail application (Google), 273**

**GNU indentation style, 215**

**GRANT command (MySQL)**

for Apache access to credentials, 149

creating column-level permissions, 95

creating permissions for a specific table, 95

for extremely limited access, 96

granting all privileges, 96

granting limited table-wide permissions, 96

IDENTIFIED BY phrase, 96

ON clause, 95

for PHP authentication example, 166

privileges, 94–95

syntax, 94

TO clause, 95

## H

**hard drives, performance and, 211**

**hardware improvements for performance, 210–211**

**hidden frames technique to avoid page refreshes, 275–276**

**hidden IFRAME technique to avoid page refreshes, 276–277**

**.htaccess file**

for `mod_auth_mysql`, 151–152

performance and, 213

for turning off `register_globals` directive, 179

for WebDAV authentication, 159

**HTML\_CSS PEAR package, 192–193**

**HTML\_Page PEAR package, 192–193**

**HTML\_QuickForm PEAR package, 193–194**

**HTML\_Table PEAR package, 194–195**

**HTTP GET values, 345**

**HTTP POST values, 345**

**httpd.conf file**

enabling SSL capabilities, 157

HostnameLookups option, 212

KeepAlive option, 212

KeepAliveTimeout option, 212

MaxClients option, 211

MaxKeepAliveRequests option, 212

MaxRequestsPerChild option, 211

MinSpareServers option, 212

for `mod_auth_mysql`, 151–152

for `mod_dav`, 158

for `mod_deflate` (Apache), 146–147

for `mod_ssl`, 155

performance and, 211–212

StartServers option, 212

for WebDAV authentication, 159

## I

**IFRAME element, hidden, 276–277**

**image source trick to avoid page refreshes, 275**

`imagealphablending()` **function (ImageResize class), 248**

`ImageCaption` **class for GD library**  
code listing, 250–251

`imageline()` function, 253–254

`imagegetttftext()` function, 253

`process()` method, 252–253

`imagecopy()` **function (ImageLogo class), 257**

`imagecopymerge()` **function (GD library), 257**

`imagecopyresampled()` **function (ImageResize class), 248**

`imagecopyresized()` **function (ImageResize class), 248**

`imageline()` **function (ImageCaption class), 253–254**

`ImageLogo` **class for GD library**  
code listing, 254–255

`imagecopy()` function, 257

`process()` method, 256–257

### ImageResize **class for GD library**

- code listing, 245–246
- imagealphablending() function, 248
- imagecopyresampled() function, 248
- imagecopyresized() function, 248
- imagesavealpha() function, 248
- process() method, 247–248

**ImageRotate class for GD library, 249–250**

**imagesavealpha() function**

- ImageResize class, 248
- WebImage class, 244

**imagettftext() function (ImageCaption class), 253**

implements **keyword (PHP), 36, 37**

**indentation standards for coding, 214–217**

**index transparency, 244**

**info command (PEAR manager), 185**

**inheritance (PHP)**

- abstract classes and, 7–8, 34–35
- described, 31
- extends keyword and, 31–33
- final keyword and, 6, 33
- language translation, 340–341
- parent:: references, 33–34
- redefining properties and methods in subclasses, 32–33
- static keyword and, 7

**ini\_get() function (PHP), 109**

**ini\_get\_all() function (PHP)**

- access values for directives, 115
- overview, 109, 110
- in XMLConfigProcessor.php class, 126, 127

**ini\_set() function (PHP), 114–115, 127–128**

**initializing properties (PHP), 21**

**inner joins (MySQL)**

- aliases for tables, 78–79
- chaining in order of evaluation, 79
- comma-delimited list of tables for, 79–81
- defined, 78
- limitations, 81

- outer joins combined with, 82
- syntax, 78
- table prefixes for, 78–79
- USING statement with, 79
- WHERE clause with, 79, 80, 81

**Innbase Oy company, 88**

**InnoDB tables (MySQL)**

- advantages, 88–89
- converting a MyISAM table to, 90
- converting to MyISAM tables, 93
- creating a table from scratch, 89
- development of InnoDB storage engine, 88
- disadvantages, 89
- foreign key constraints, 88–89, 90–93

**INSERT statement (MySQL), 75–76**

**installing**

- APC caching engine, 288
- DocuWiki Micro CMS, 326
- eAccelerator caching engine, 291
- ExponentCMS, 309–310
- JPCache caching engine, 296–297
- MediaWiki Micro CMS, 325
- memcached caching engine, 300–301
- Ming extension (PHP), 263
- mod\_auth\_mysql (Apache), 150–151
- PEAR packages, 186
- phpWebsite CMS, 315–317
- Serendipity blog Micro CMS, 324
- TikiWiki CMS, 318–320
- WordPress blog Micro CMS, 323
- XOOPS CMS, 312–313
- Zend Optimizer caching engine, 294–295

**instantiating**

- classes, 22–23
- SWFMovie object (Ming), 263

**interface keyword (PHP), 36**

**interfaces (PHP)**

- defining, 36
- described, 35
- extends keyword with, 36–37
- implementing, 36, 37
- language translation, 342–343

multiple for one class, 37

overview, 35–37

PHP5 changes, 5

uses for, 35

## **Internet Assigned Numbers Authority, 72**

### **Internet resources**

AJAX information, 285

Apache mod\_ssl information, 157

APC caching engine, 288

Auth\_HTTP PEAR package information, 188

CMS resources, 327–328

CPAINT, 283

CVS version control system, 351

DocuWiki Micro CMS, 326

Drupal CMS, 321

eAccelerator caching engine, 291

ExponentCMS, 309, 310

Fileinfo PECL package, 196

GD section of PHP manual, 263

Google's GMail application, 273

Internet Assigned Numbers Authority, 72

JavaWebStart site, 351

JPCache caching engine, 296, 300

JPSpan, 283

Land Information New Zealand site, 67

Mambo CMS, 321

MediaWiki Micro CMS, 325

memcached caching engine, 300, 301

Micro CMSs, 323, 324, 325, 326, 327

Ming information, 263

MyBench tool, 350

MySQL Administrator tool, 348

MySQL Query Browser tool, 349

MySQL Super Smack tool, 350

PDF Open Source libraries, 233

PDFLib extension (PHP), 233, 234

PDFLib Lite extension (PHP), 234

PDO PECL package, 196, 197

PEAR configuration variables, 185

PEAR database packages, 191

PEAR manager commands, 184

PEAR packages, 195

PEAR::DB package, 187

PEPr site, 184

PHP Wiki site, 196

PHP-Nuke CMS, 321

phpWebsite CMS, 315

Postnuke CMS, 322

SAJAX information, 282

security-related sites, 182

Serendipity blog Micro CMS, 324

SourceForge site, 263, 353

SQLite information, 17

Subversion version control system, 351

Tidy library, 17

TikiWiki CMS, 318

Umbrello UML tool, 353

WordPress blog Micro CMS, 323

WROX site, 76

Xdebug PECL package, 197

XOOPS, 312

Zend Optimizer, 295

## **interpolation, benchmarking speed of, 201–205**

**Iterator class (PHP), 5**

**iterators (PHP), 5–6**

## **J**

**Java. See language translation**

**Javadoc-style comments, 220**

**JavaScript. See also AJAX (Asynchronous JavaScript and XML)**

hidden frames technique, 275–276

image source trick, 275

**JavaWebStart website, 351**

**JPCache caching engine**

configuring, 297–299

installing, 296–297

other caching engines with, 305

overview, 296

removing, 300

situations suitable for, 305

website, 296, 300

**JPSpan AJAX library, 283**

**JSP. See language translation**

### K

**K & R (Kernighan and Ritchie) indentation style, 215, 216, 217**

### L

**Land Information New Zealand website, 67**  
**language translation**

- abstract class declaration, 341–342
- array creation, 331
- array element referencing, 331
- base class definition, 338
- calling method in same class, 338–339
- calling parent method, 341
- class constructor, 339
- class destructor, 339–340
- class inheritance, 340–341
- class property declaration, 338
- class using interface, 343
- class visibility keywords, 337
- comments, 329
- conditional evaluation, 331–332
- constant declaration, 330
- exception handling, 344
- function/method declaration, 336
- grouping classes with namespaces, 337
- HTTP GET values, 345
- HTTP POST values, 345
- including external files, 343–344
- interface declaration, 342–343
- loops, advance to next round, 336
- loops, `do...while`, 334
- loops, `for`, 334
- loops, `foreach`, 335
- loops, termination, 335
- loops, `while`, 333
- multiple-choice selection, 332–333
- object creation, 340
- returning function values, 336–337
- server variables, 345–346
- static class method usage, 340

- stopping subclassing, 342
- throwing exceptions, 344–345
- variable declaration, 330
- variable typing, 330

**left outer joins (MySQL), 81, 84**

**load() function (WebImage class), 243–244**

**log files**

- `emailException` class for monitoring logs, 3–4

- file access logger example, 63–67
- for URL rewriting (Apache), 143–144

**logo() method (WebImage class), 255–256**

**loops**

- array functions not requiring, 48
- `array_walk()` function for avoiding, 49, 51–52
- for loops, 48, 335
- foreach loops, 48, 49, 51–52, 335
- language translation, 333–336

### M

**Mac OS, WebDAV setup for, 163**

**Macromedia Flash versus Ming, 263**

**magic methods (PHP)**

- for accessing methods and properties not explicitly defined, 37
- `__call()`, 8–9, 38
- `__construct()`, 6, 25–26, 37
- `__destruct()`, 6, 26–27, 37
- `__get()`, 8, 38–39
- for overloading, 8–9
- overview, 37
- `__set()`, 8, 38–39
- `__sleep()`, 39, 40–43
- `__toString()`, 43–44
- `__wakeup()`, 40–43

**magic quotes (PHP)**

- checking `php.ini` settings, 113
- disabling, 106, 113
- overview, 112–113
- `php.ini` setting for, 106, 113

**maintaining databases (MySQL), 103–104****Mambo CMS, 321****MediaWiki Micro CMS, 325–326****memcached caching engine**

- daemon, 300
- installing, 300–301
- Memcache object methods, 301–303
- other caching engines with, 305
- overview, 300
- PECL module for, 301
- removing, 304
- using, 301–304
- websites, 300, 301

**memory (RAM)**

- performance and, 200, 210–211
- series of array functions and, 59–60

**methods (MySQLi)**

- calling procedurally, 13
- mysqli class, 13–14
- mysqli\_result class, 16
- mysqli\_stmt class, 15

**methods of AJAX interfaces, 277–278****methods (PHP). See also functions (PHP);****magic methods (PHP); specific methods and functions**

- abstract, 8
- defining, 21
- described, 20, 21
- final keyword with, 6
- language translation, 336, 338–339, 340, 341
- naming conventions, 221–222
- redefining in subclasses, 32–33
- static keyword with, 7, 27–28
- stream wrapper methods, 62–63

**Micro CMSs**

- blog packages, 322–324
- DocuWiki, 326
- MediaWiki, 325–326
- other packages, 327
- Serendipity, 324
- wiki packages, 324–326
- WordPress, 323

**Ming extension (PHP)**

- Content-Type header for, 268
- coordinate system, 264
- creating a Flash display, 265–269
- frames, 264
- installing, 263
- Macromedia Flash versus, 263
- objects, 263, 264–265
- outputting directly to browser, 268–269
- overview, 263–264
- simple example, 263–264
- slide show example, 266–269
- websites for further information, 263

**mod\_alias (Apache), 226****mod\_auth\_mysql (Apache)**

- advantages, 166
- AuthMySQLAuthoritative directive, 154
- AuthMySQLDB directive, 151, 152
- AuthMySQLEnable directive, 151, 152
- AuthMySQLGroupCondition directive, 154
- AuthMySQLGroupField directive, 154
- AuthMySQLGroupTable directive, 154
- AuthMySQLHost directive, 152
- AuthMySQLNameField directive, 153
- AuthMySQLPassword directive, 151, 152
- AuthMySQLPasswordField directive, 153
- AuthMySQLPort directive, 152
- AuthMySQLPwEncryption directive, 151, 153
- AuthMySQLSaltField directive, 153
- AuthMySQLSocket directive, 152
- AuthMySQLUser directive, 151, 152
- AuthMySQLUserCondition directive, 153
- AuthMySQLUserTable directive, 153
- AuthName directive, 151
- AuthType directive, 151
- database setup for, 149–150
- described, 129
- .htaccess file for, 151–152
- httpd.conf file for, 151–152
- installing, 150–151
- minimum directives for security, 151

**mod\_dav (Apache), 129, 158**

### **mod\_deflate (Apache)**

- building and enabling, 146
- content compression using, 146–148
- DeflateCompressionLevel directive, 147
- DeflateFilterNote directive, 148
- DeflateWindowSize directive, 148
- described, 129
- effectiveness of, 148
- httpd.conf file for, 146–147

### **mod\_rewrite (Apache). See also URL rewriting (Apache)**

- capabilities of, 129
- decoupling site document tree from server filesystem, 226
- described, 129
- enabling, 130–131, 133
- RewriteBase directive, 143
- RewriteCond directive, 139–143
- RewriteLog directive, 143–144
- RewriteLogLevel directive, 144
- RewriteRule directive, 131–138

### **mod\_spelling (Apache), 129, 144–145**

### **mod\_ssl (Apache), 129, 154–155. See also SSL (Secure Sockets Layer) with Apache Moodle CMS, 327**

### **multiplication operators, performance and, 200–201**

### **MyBench MySQL tool, 350**

**my.cnf configuration file (MySQL), 88, 100**

### **MyISAM tables (MySQL)**

- converting InnoDB to, 93
- converting to InnoDB, 90
- as default storage engine, 88

### **MySQL. See also SELECT statement (MySQL)**

- access control, 93–100
- aliases for tables, 78–79
- ALTER TABLE statement, 87, 90–91, 92
- alternative tools, 347–350
- analyzing databases, 100–102
- Apache authentication with, 129, 149–154, 166
- basics, 73–77
- columns\_priv table, 99, 100
- creating backups, 103
- creating databases, 74–75

- creating tables, 75, 86–87
- cross joins, 81
- db table, 98–99, 100
- deleting database information, 77, 92
- example database tables, 73–74
- foreign key constraints, 88–89, 90–93
- full-text searching, 86–88
- /G terminator for queries, 86
- GRANT command, 94–96
- inner joins, 78–81, 82
- InnoDB tables, 88–93
- inserting data in tables, 75–76
- maintaining databases, 103–104
- monitor, 74
- MyBench tool, 350
- my.cnf configuration file, 88
- MyISAM tables, 88, 90, 93
- MySQL Administrator tool, 347–348
- MySQL Query Browser tool, 349–350
- MySQL Super Smack tool, 350
- outer joins, 81–82
- querying multiple tables, 77–86
- reloading privilege tables, 100
- restoring from backups, 103–104
- retrieving database information, 76–77
- REVOKE command, 96–97
- server restriction, 100
- setting the active database, 75
- SHOW COLUMNS command, 101
- SHOW CREATE TABLE command, 101–102
- SHOW DATABASES command, 102
- SHOW GRANTS command, 102
- starting the command-line client, 74
- tables\_priv table, 99, 100
- unions, 82–86
- updating database information, 77, 91–92
- USE statement, 75
- user administration, 93–100
- user table, 97–98, 99–100
- USING statement, 79
- WHERE clause, 79, 80, 81

### **MySQL Administrator tool, 347–348**

### **MySQL Query Browser tool, 349–350**

### **MySQL Super Smack tool, 350**

`mysqli` **class**, 13–15  
**MySQLi (MySQL improved)**, 12–16  
`mysqli_pconnect()` **function, performance and**, 231  
`mysqli_result` **class**, 16  
`mysqli_stmt` **class**, 15

## N

**namespaces**  
 collision prevented by `create_function()` function, 57–58  
 language translation, 337  
**naming conventions**, 220–222  
**nesting, reducing levels of**, 217–220  
**network connection, performance and**, 210  
**new keyword (PHP)**, 22–23  
**NO ACTION option (MySQL)**, 92  
**NULL values**  
 callback for `array_map()` function (PHP), 50–51  
 InnoDB foreign key constraints and (MySQL), 89  
 for missing columns in unions (MySQL), 84–86  
**NULs in `create_function()` function names**, 58

## O

**object-oriented programming. See OOP; PHP5 OOP**  
**objects (PHP)**  
 creating by instantiating classes, 20, 22–23  
 described, 20  
 language translation, 340  
 real-world compared to OOP, 20  
`$this` reference variable for, 23  
`ob_start('ob_gzhandler')` **function**, 231  
`onreadystatechange` **property (AJAX)**, 278, 279  
**OOP (object-oriented programming). See also PHP5 OOP**  
 changes in PHP5, 1–9  
 procedural programming versus, 19–20  
`open()` **method (AJAX)**, 277

**open source movement**, 183  
**opening**  
 MySQL command-line client, 74  
 streams (PHP), 62, 63  
**OpenSourceCMS website**, 327  
**osCommerce CMS**, 327  
**outer joins (MySQL)**  
 generic form, 81  
 inner joins combined with, 82  
 left versus right, 81  
 unions combined with, 84  
**output buffering, performance and**, 227–228, 231  
`overload()` **function versus magic methods (PHP)**, 8  
`overrideMimeType()` **method (AJAX)**, 278

## P

**Pachev, Sasha (MySQL Super Smack creator)**, 350  
**parent keyword (PHP)**, 33–34  
**passing variables, PHP5 changes for**, 1  
**PDF (Portable Document Format)**  
 Open Source libraries, 233  
 PDFLib extension for, 233–241  
 PDFlib functions for, 233  
**PDF resume generator application**, 238–241  
`pdf_close()` **function (PDFLib)**, 234  
`pdf_closepath()` **function (PDFLib)**, 236  
`pdf_fill()` **function (PDFLib)**, 236  
`pdf_findfont()` **function (PDFLib)**, 236  
`pdf_get_buffer()` **function (PDFLib)**, 237  
`pdf_get_value()` **function (PDFLib)**, 236, 237  
**PDFlib extension (PHP)**  
 completing the PDF file, 237  
 configuration, 234  
 coordinate system, 237  
`drawHR()` function, 236  
 font functions, 236–237  
 getting started, 234  
 helper functions, 235–237  
 path functions, 236

### **PDFLib extension (PHP) (continued)**

- PDF resume generator example, 238–241
- `pdf_close()` function, 234
- `pdf_get_value()` function, 236
- PDFlib functions versus, 233
- PDFLIB-manual.pdf for, 234
- `pdf_new()` function, 234
- `pdf_set_info()` function, 235
- `pdf_show_xy()` function, 237
- performance advantages of, 233
- required elements, 235
- specifying document information, 235
- website, 233, 234

### **PDFlib functions (PHP)**

- helper functions, 235–237
- moved to PECL, 233
- PDFLib extension versus, 233

### **PDFLib Lite extension (PHP), 234**

- PDFLIB-manual.pdf, **234**
- `pdf_lineto()` **function (PDFLib), 236**
- `pdf_moveto()` **function (PDFLib), 236**
- `pdf_new()` **function (PDFLib), 234**
- `pdf_setfont()` **function (PDFLib), 236**
- `pdf_set_info()` **function (PDFLib), 235**
- `pdf_show_xy()` **function (PDFLib), 237**

### **PDO PECL package, 196–197**

### **PEAR (PHP Extension and Repository)**

- APC caching engine provided with, 287
- Auth\_HTTP package, 187–188
- Benchmarking class, 207–208
- checking for pre-installed packages, 186
- coding standards, 214
- configuration variables, 185
- Date package, 189–190
- DB package, 190–192
- described, 183–184
- HTML\_CSS package, 192–193
- HTML\_Page package, 192–193
- HTML\_QuickForm package, 193–194
- HTML\_Table package, 194–195
- installing packages, 186
- Javadoc-style comments recommended by, 220

- K & R indentation style specified by, 216
- keeping updated with patches, 182
- libraries for phpWebsite CMS, 316
- packages available, 187, 195–196
- PEAR manager, 184–186
- PEAR::Auth, 187, 188
- PEAR::Cache and PEAR::Cache\_Lite packages, 224–225
- PEAR::DB package, 187
- PEPr (PEAR Proposal System), 184
- using installed packages, 187

### **PEAR Proposal System (PEPr), 184**

#### **PEAR::Auth package, 187, 188**

#### **PEAR::Cache and PEAR::Cache\_Lite packages, 224–225**

#### **PEAR::DB package, 187**

### **PECL (PHP Extension Community Library)**

- APC caching engine provided with, 287, 288
- described, 184, 196
- Fileinfo package, 196
- keeping updated with patches, 182
- memcached caching engine module, 301
- PDFlib functions in, 233
- PDO package, 196–197
- Xdebug package, 197

### **PEPr (PEAR Proposal System), 184**

### **performance**

- ab ApacheBench program, 208–209
- Apache web server improvements, 211–213
- benchmarking, defined, 206–207
- benchmarking, speed of strings example, 201–205
- caching and, 223–227
- checking array size and, 229
- conditional statements and, 230
- echo versus print and, 228–229
- hardware improvements, 210–211
- memory and, 200
- multiplication operators and, 200–201
- NT4/Access report generator anecdote, 200
- `ob_start('ob_gzhandler')` and, 231
- output buffering and, 227–228, 231

overview, 231–232  
 PDFlib functions versus PDFlib extension and, 233  
 PEAR Benchmarking class, 207–208  
 persistent database connections and, 231  
 PHP coding standards, 213–223  
 planning programs for, 199–201  
 profiling, defined, 207  
 regular expression syntax and, 230  
 releasing resources and, 231  
 scratch variables and, 229–230  
 string functions and, 230  
 strings and, 201–205  
 top process monitor utility, 208–210  
 unintuitive results of performance tuning, 206  
 unset() function and, 230

**persistent database connections, performance and, 231**

**PHP. See also language translation; PHP5 changes and enhancements; PHP5 OOP**

array functions and callbacks, 48–60  
 authentication, 166–177  
 automated version and feature checking, 116–128  
 changing configuration dynamically, 113–115  
 checking extensions and their properties, 111–112  
 coding standards, 213–223  
 glob() function, 60–61  
 listing installed version, 108  
 obtaining current runtime settings, 108–113  
 output compression for scripts, 148–149  
 retrieving configuration settings at runtime, 109–110  
 streams, 61–72  
 variety and number of functions in, 47

**PHP configuration. See php.ini configuration file; runtime configuration (PHP)**

**PHP Extension and Repository. See PEAR PHP Extension Community Library. See PECL**

**PHP extensions. See also PEAR (PHP Extension and Repository); PECL (PHP Extension Community Library); specific extensions**

CURL, 271  
 DOM, 16, 271  
 extension\_loaded() function, 111, 124, 125  
 functions for getting information on, 111–112  
 GD library, 241–263  
 Ming, 263–269  
 PDFlib, 233–241  
 PDFlib Lite, 234  
 SimpleXML, 16, 269–270  
 SOAP, 17, 271  
 Tidy library, 17  
 XML extensions new in PHP5, 16–17  
 XSL, 16  
 ZIP extensions, 271

**PHP Wiki website, 196**

**phpauthbasic.php script**

attempt\_auth() function, 169–170  
 check\_login() function, 170–171  
 code listing, 167–168  
 constants defined for database connection, 169  
 including in standard PHP page, 171  
 testing, 171

**phpauthforms.php script, 172–174**

**phpBB CMS, 327**

**PHP-controlled authentication**

Apache-controlled authentication and, 166, 171  
 creating the authentication database, 166–167  
 drawbacks of Basic Authentication, 171  
 forms authentication, 172–174  
 other access restrictions, 174–177  
 phpauthbasic.php script for Basic Authentication, 167–171  
 phpauthforms.php script for forms authentication, 172–174  
 testing Basic Authentication, 171

## PHP5 changes and enhancements

- abstract keyword, 7–8
- access modifiers, 6
- constructors and destructors, 6
- exceptions, 1–5
- final keyword, 6
- interfaces, 5
- iterators, 5–6
- magic methods for overloading, 8–9
- MySQLi extension, 12–16
- new functions, 10–12
- OOP changes, 1–9
- overview, 1, 17
- passing variables, 1
- php.ini configuration file changes, 12, 107–108
- SQLite improvements, 17
- static keyword, 7
- Tidy library extension, 17
- XML support, 16–17

## PHP5 OOP

- abstract keyword, 7–8
- access modifiers, 6, 23–25
- assignment versus cloning, 29–30
- changes and enhancements, 1–9
- class basics, 20–23
- class constants, 28–29
- constructors, 6, 25–26
- defining classes, 20
- destructors, 6, 26–27
- exceptions, 1–5
- final keyword, 6
- inheritance, 31–35
- instantiating classes, 22–23
- interfaces, 5, 31, 35–37
- iterators, 5–6
- magic methods, 6, 8–9, 37–44
- methods, 21
- overview, 45
- properties, 20–21
- static keyword, 7, 27–28
- visibility, 23–25

## php.ini configuration file

- access values for directives, 115
- allow\_call\_time\_pass\_reference directive, 106, 115
- for APC caching engine, 288–290
- asp\_tags directive, 106
- changing directives with ini\_set(), 114–115
- display\_errors directive, 106
- for eAccelerator caching engine, 291–294
- error\_log directive, 106
- error\_reporting directive, 106
- for JPCache caching engine, 297–299
- log\_errors directive, 106
- magic\_quotes\_gpc directive, 106, 113
- magic\_quotes\_runtime directive, 113
- mail.force\_extra\_parameters directive, 107
- MySQLi settings, 12
- output compression settings, 149
- for PHP authentication example, 171
- PHP5 improvements, 12, 107–108
- recommended configuration directives, 105–107
- register\_globals directive, 105, 116
- register\_long\_arrays directive, 107
- retrieving settings at runtime, 109–110
- safe\_mode directive, 171
- session.hash\_bits\_per\_character directive, 108
- session.hash\_function directive, 108
- short\_open\_tag directive, 106–107
- variables\_order directive, 106
- for Zend Optimizer caching engine, 295
- zend.zel\_compatibility\_mode directive, 108
- zlib.output\_compression directive, 107, 149

**phpmychat CMS, 327**

**PHP-Nuke CMS, 321**

**phpversion() function, 108, 126**

**phpWebsite CMS**

- customizing default settings, 317
- default features, 316–317
- installing, 315–317
- PEAR libraries for, 316
- strengths and weaknesses, 317–318
- website, 315

**pi() function (PHP), 29****Portable Document Format. See PDF****Postnuke CMS, 322****preg\_grep() function, 53****preg\_replace\_callback() function, 53–55****PrestoPoll CMS, 327****print statement (PHP)**

- echo statement versus, 228–229
- \_\_toString() magic method with, 44

**private keyword (PHP), 6, 23****privkey.pem file (SSL), 156****process() method**

- ImageCaption class, 252–253
- ImageLogo class, 256–257
- ImageResize class, 247–248
- ImageRotate class, 249–250

**processConfig method (PHP), 118–119****processors, performance and, 210****profiling, defined, 207****properties (MySQLi)**

- mysqli class, 14–15
- mysqli\_result class, 16
- mysqli\_stmt class, 15

**properties (PHP)**

- Circle class example, 21
- defining, 20–21
- described, 20
- initializing, 21
- language translation, 338
- naming conventions, 221, 222
- real versus synthetic, 221
- redefining in subclasses, 32–33
- static keyword with, 27
- \$this reference variable for, 23

**protected keyword (PHP), 6, 24**

**public keyword (PHP), 6, 23, 24**

**Q****querying multiple tables (MySQL)**

- inner joins, 78–81
- outer joins, 81–82
- unions, 82–86

**quotation marks (“)**

- for function names within
  - set\_exception\_handler(), 5
- speed of strings and, 201–205

**R****RAM (memory)**

- performance and, 200, 210–211
- series of array functions and, 59–60

**rar extension (PHP), 271****Raymond, Eric S. (Jargon File author), 214****readdir() function (PHP), 61****reading**

- directories (PHP), 60–61
- streams (PHP), 62

**readyState property (AJAX), 278, 279****recreate\_function() function, 58****RecursiveIterator class (PHP), 6****register\_globals directive**

- coding to reduce likelihood of exploits, 179, 182
- exploits, 178–179
- .htaccess file for turning off, 179
- HTTP\_REFERER value and, 174
- recommended configuration, 105, 116

**regular expressions**

- performance and, 230
- with RewriteRule directive (mod\_rewrite of Apache), 133–134

**reqs.xml file for configuration checker (PHP), 116****resize() function (WebImage class), 246–247, 259****responseText property (AJAX), 278****responseXML property (AJAX), 278****RESTRICT option (MySQL), 92****REVOKE command (MySQL), 96–97**

**RewriteBase directive (mod\_rewrite of Apache), 143**

**RewriteCond directive (mod\_rewrite of Apache)**

- basic format, 139
- chaining multiple lines together, 141
- comparison and system-check conditionals, 141–142
- comparison operators, 142
- HTTP Header variables, 139
- nocase or NC flag, 142
- ornext or OR flag, 143
- Request variables, 139–140
- Server variables, 140
- Special variables, 140–141
- Time variables, 140
- usage, 139–142
- using variables, 141

**RewriteLog directive (mod\_rewrite of Apache), 143–144**

**RewriteLogLevel directive (mod\_rewrite of Apache), 144**

**RewriteRule directive (mod\_rewrite of Apache)**

- automatically redirecting URL to new directory, 132
- basic format, 131
- chain or C flag, 136
- cookie or CO flag, 137
- enabling mod\_rewrite, 130–131, 133
- env or E flag, 137
- forbidden or F flag, 135
- gone or G flag, 136
- for human-input-friendly URLs, 132–133
- last or L flag, 136
- next or N flag, 136
- nocase or NC flag, 135
- noescape or NE flag, 138
- nosubreq or NS flag, 138
- passthrough or PT flag, 138
- proxy or P flag, 138
- qsappend or QSA flag, 137
- redirect or R flag, 135

- regular expressions with, 133–134
- for search-engine-friendly URLs, 132–133
- skip or S flag, 137
- syntax for flags, 135
- tokens for, 131–132
- type or T flag, 137
- as type-checking layer for PHP scripts, 134–135
- usage, 131–135

**right outer joins (MySQL), 81**

**rotate() function (WebImage class), 249**

**runtime configuration (PHP). See also configuration checker (PHP)**

- automated version and feature checking, 116–128
- changing directives with `ini_set()`, 114–115
- changing dynamically, 113–115
- disabling magic quotes, 113
- obtaining current runtime settings, 108–113
- removing the time limit, 113–114

## S

**SAJAX (Simple AJAX Toolkit), 282–283**

**save() method (WebImage class), 244–245**

**scandir() function (PHP), 61**

**Schlossnagle, George (APC creator), 287**

**scratch variables, performance and, 229–230**

**scripts (PHP)**

- compression for (Apache), 148–149
- mod\_rewrite as type-checking layer for (Apache), 134–135
- Xdebug PECL package for debugging, 197

**Secunia website, 182**

**Secure Sockets Layer with Apache. See SSL with Apache**

**security. See authentication; site security**

**SELECT statement (MySQL)**

- cross joins with, 81
- for full-text searching, 87–88
- /G terminator for vertical display, 86
- inner joins with, 78–81, 82

- outer joins with, 81–82
- simple form, 76–77
- unions with, 82–86
- self keyword (PHP), 7, 27–28, 50**
- send() method (AJAX), 278**
- Sendmail program, 3–4, 107**
- Serendipity blog Micro CMS, 324**
- server variables (PHP)**
  - access control and, 174–175
  - language translation, 345–346
  - RewriteCond directive (mod\_rewrite of Apache), 140
- sessions, php.ini settings for, 108**
- SET DEFAULT option (MySQL), 92**
- \_\_set() magic method (PHP), 8, 38–39**
- SET NULL option (MySQL), 92**
- set\_error\_handler() function (PHP), 1–5**
- set\_exception\_handler() function (PHP), 4–5**
- set\_magic\_quotes\_runtime() function (PHP), 113**
- setRequestHeader() method (AJAX), 278**
- set\_time\_limit() function (PHP), 114**
- Shape class example**
  - as abstract class, 34–35
  - Circle class inheritance from, 31–33
- Shockwave Flash files. See Ming extension (PHP)**
- SHOW COLUMNS command (MySQL), 101**
- SHOW CREATE TABLE command (MySQL), 101–102**
- SHOW DATABASES command (MySQL), 102**
- SHOW GRANTS command (MySQL), 102**
- Simple AJAX Toolkit (SAJAX), 282–283**
- SimpleXML extension (PHP)**
  - overview, 16, 269–270
  - xpath() method, 125, 127
- single quotes ('), speed of strings and, 201–205**
- site security. See also authentication**
  - access control with Apache, 165–166
  - access control with PHP, 166–177
  - code/peer review for, 182
  - cross-site scripting (XSS), 180–181
  - keeping updated with patches, 181–182
  - register\_globals exploits, 178–179, 182
  - social engineering and, 182
  - SQL injection attacks, 179–180, 182
  - website attacks, 178–181
  - websites for further information, 182
- slash (/)**
  - /G terminator for MySQL queries, 86
  - after scheme name in PHP URLs (://), 72
- \_\_sleep() magic method (PHP), 39, 40–43**
- SOAP extension (PHP), 17, 271**
- SourceForge website, 263, 353**
- spell checking URLs (Apache), 144–145**
- SPL (Standard Public Library), 5**
- SQL injection attacks, 179–180, 182**
- SQLite, 17**
- SSL (Secure Sockets Layer) with Apache**
  - checking if SSL is enabled in Apache, 154
  - creating a directory for the SSL website, 157
  - creating a self-signed certificate and key, 155–156
  - dynamically enabling SSL in Apache, 155
  - enabling SSL capabilities in httpd.conf, 157
  - further information, 157
  - information file for testing, 157
  - mod\_ssl for, 129, 154–155
  - privkey.pem file, 156
  - removing the passphrase, 156
  - statically enabling SSL in Apache, 154
  - testcert.csr file, 156
  - testing, 157
  - VirtualHost section for port 443, 157
- ssl.conf file, 157**
- Standard Public Library (SPL), 5**
- static keyword (PHP), 7, 27–28**
- status property (AJAX), 278, 279**
- stream\_close() function (PHP), 63, 67**
- stream\_eof() function (PHP), 63**
- stream\_open() function (PHP)**
  - file access logger example, 67
  - GIS photographic database query example, 70–71
  - overview, 62, 63

## stream\_read() function (PHP)

---

### `stream_read()` function (PHP)

- file access logger example, 67
- GIS photographic database query example, 71
- overview, 62

### streams (PHP)

- building a stream wrapper, 62–63
- defined, 61–62
- file access logger example, 63–67
- GIS database query example, 67–72
- only byte streams allowed in PHP, 62
- overview, 61–62
- URL system for naming stream resources, 72
- wrapper methods, 62–63

### `stream_tell()` function (PHP), 71

### `stream_wrapper_register()` function (PHP), 62

### `stream_write()` function (PHP), 63, 67

### strings (PHP)

- benchmarking speed of, 201–205
- performance and string functions, 230
- `__toString()` magic method, 43, 44
- `stripslashes()` function (PHP), 113

### subclassing

- language translation for stopping, 342
- redefining properties and methods in subclasses, 32–33

### Subversion version control system, 350–351

### SWFAction object (Ming), 264

### SWFBitmap object (Ming), 264, 267

### SWFButton object (Ming), 264, 268

### SWFDisplayItem object (Ming), 265

### SWFFill object (Ming), 265

### SWFFont object (Ming), 265

### SWFGradient object (Ming), 265

### SWFMorph object (Ming), 265

### SWFMovie object (Ming), 263, 264, 267

### SWFShape object (Ming), 265, 267, 268

### SWFSprite object (Ming), 264, 265, 266–267

### SWFText object (Ming), 265

### SWFTextField object (Ming), 265

`switch` statements, language translation for, 332–333

synthetic property naming conventions, 221

## T

tab characters for indentation, avoiding, 216–217

`tables_priv` table (MySQL), 99, 100

`testcert.csr` file (SSL), 156

`$this` reference variable (PHP), 23, 50

Tidy library extension (PHP), 17

Tigris.org community, 351

### TikiWiki CMS

- customizing default settings, 320–321
- default features, 318–319
- “inactive” features, 319–320
- installing, 318–320
- strengths and weaknesses, 321
- website, 318

`top` process monitor utility, 208–210

`__toString()` magic method (PHP), 43, 44

`try/catch/throw` blocks (PHP)

- language translation, 344–345
- for `pdf_new()` function, 234
- syntax, 2

## U

Umbrello UML tool, 352–353

### UML diagrams

- ArgoUML tool, 351–352
- overview, 351
- Umbrello tool, 352–353

### unions (MySQL)

- column lists of different lengths and, 83
- cross joins versus, 82
- described, 82
- differently named table columns and, 83
- /G terminator for queries, 86
- joins combined with, 84
- NULL values for missing columns, 84–86

`unset()` **function, performance and**, 230

**UPDATE statement (MySQL)**, 77, 91–92

### **URL rewriting (Apache)**

- automatically redirecting URL to new directory, 132
  - avoiding automatic escaping, 138
  - combining `mod_rewrite` with other modules, 138
  - controlling verbosity of log records, 144
  - enabling `mod_rewrite`, 130–131, 133
  - forcing requests through the proxy module, 138
  - for human-input-friendly URLs, 132–133
  - making patterns case-insensitive, 135
  - overriding MIME types, 137
  - preserving query-strings, 137
  - `RewriteBase` directive, 143
  - `RewriteCond` directive, 139–143
  - `RewriteLog` directive, 143–144
  - `RewriteLogLevel` directive, 144
  - `RewriteRule` directive, 131–138
  - for search-engine-friendly URLs, 132–133
  - sending HTTP 403 Forbidden response, 135
  - sending HTTP 410 Gone response, 136
  - setting cookies, 137
  - setting environment variables, 137
  - skipping rules, 137, 138
  - specifying base or prefix path, 143
  - specifying log file for processing, 143–144
  - starting the process at beginning of rule list, 136
  - stopping processing after successful match, 136
  - taking internal subrequests into account, 138
  - testing strings against patterns or conditions, 139–143
  - as type-checking layer for PHP scripts, 134–135
  - using blocks of code, 136
  - using formal redirect with HTTP status code, 135
- URLs. See also URL rewriting (Apache)**
- guidelines for naming stream resources, 72
  - PHP conventions for, 72
  - spell checking (Apache), 144–145

**USE statement (MySQL)**, 75, 149

### **user administration (MySQL)**

- alternatives to `GRANT` and `REVOKE`, 97–100
  - `columns_priv` table for, 99, 100
  - `db` table for, 98–99, 100
  - `GRANT` command for, 94–96
  - overview, 93–94
  - reloading privilege tables, 100
  - `REVOKE` command for, 96–97
  - `tables_priv` table for, 99, 100
  - `user` table for, 97–98, 99–100
- user table (MySQL)**, 97–98, 99–100
- USING statement with inner joins (MySQL)**, 79

## **V**

`var` **keyword (PHP4)**, 24

### **variables (PHP)**

- assignment by value versus by reference, 29
- cloning, 29–30
- language translation, 330
- naming conventions, 220, 222
- passing, PHP5 changes for, 1
- `php.ini` settings for, 105, 106

**VBScript. See language translation**

**version control systems**, 350–351

**visibility keywords. See access modifiers (PHP)**

## **W**

`__wakeup()` **magic method (PHP)**, 40–43

**Web CMSs. See also CMSs (Content Management Systems)**

- all-inclusive Web CMSs, 309–322
- described, 308
- Drupal, 321
- ExponentCMS, 309–312
- Mambo, 321
- Micro CMSs, 322–328
- open source packages, 308
- PHP-Nuke, 321
- phpWebsite, 315–318
- Postnuke, 322

## Web CMSs (continued)

TikiWiki, 318–321

XOOPS, 312–315

## WebCalendar CMS, 327

## WebDAV (Web-based Distributed Authoring and Versioning)

authentication, 159

building mod\_dav as shared object, 158

building mod\_dav statically, 158

checking if mod\_dav is enabled, 158

described, 158

directory for file repository, 158

editors supporting, 159

httpd.conf file entries for mod\_dav, 158

Mac OS setup, 163

Windows 2000/XP setup, 159–162

## WebImage class for GD library

capabilities of, 241

caption() method, 251–252

code listing, 241–243

display() method, 244

imagesavealpha() function, 244

load() function, 243–244

logo() method, 255–256

public properties, 243

resize() function, 246–247, 259

rotate() function, 249

save() method, 244–245

## while loops, language translation for, 333

## Whitesmith's indentation style, 215

## wiki Micro CMSs

DocuWiki, 326

MediaWiki, 325–326

origin of term, 324

overview, 323–324

## Windows 2000/XP, WebDAV setup for, 159–162

## WordPress blog Micro CMS, 323

## writing streams (PHP), 63

## WROX website, 76

# X

## Xdebug PECL package, 197

## XHTML, AJAX use of, 273

## XML. See also AJAX (Asynchronous JavaScript and XML)

configuration checker (PHP), 116–128

further information, 285

PHP5 support for, 16–17

SimpleXML extension (PHP), 16, 125, 127, 269–270

## XMLConfigProcessor.php class

checkConfig() function, 126–128

checkConfigAccess() method, 124

checkDeps() method, 124

checkExtensions() method, 125

checkPHP() method, 125–126

code listing, 119–122

constructor, 122–123

extension\_loaded() function in, 124, 125

ini\_get\_all() function in, 126, 127

ini\_set() function in, 127–128

\$message property, 122

phpversion() function in, 126

process() method, 123

SimpleXML xpath() method in, 125, 127

version\_compare() function, 126

\$xml property, 122

\$xmlfile property, 122

## XMLHTTP and XMLHttpRequest interfaces (AJAX)

further information, 285

handling ReadyStateChange event, 280

handling the response, 280–282

history, 277

methods, 277–278

properties, 278–279

working with browser differences, 279–280

## XOOPS (eXtensible Object Oriented Portal System)

customizing default settings, 314–315

---

- default features, 313
- installing, 312–313
- other modules available, 313
- strengths and weaknesses, 315
- websites, 312

`xpath()` **method (SimpleXML), 125, 127**

**XSL extension (PHP), 16**

**XSS (cross-site scripting), 180–181**

## Z

**Zawodny, Jeremy (MySQL guru), 350**

**ZenCart CMS, 327**

**Zend Engine 1, `php.ini` setting for, 108**

**Zend Optimizer caching engine**

- configuring, 295
- installing, 294–295
- other caching engines and, 305
- overview, 294
- removing, 296
- situations suitable for, 305
- website, 295

**ZIP extensions (PHP), 271**

**Zlib library (PHP), 107**





