

Index

Symbols

- `{}` (braces), 291
- `@` tag, 279
- `**` wildcard, 339

A

- abbreviations, 50
- abbr tag, 50
- abort() function, 79
- abortAll() function, 83, 338-339
- accessibility, 44
 - design
 - easily targeted controls*, 62-63
 - high-contrast design*, 59-60
 - zoomable interfaces*, 60-62
 - event listeners, 56-58
 - screen readers, 31
 - compatibility with Ajax*, 53-54
 - content replacement*, 54-55
 - form validation*, 55-56
 - semantic markup, 30-31
 - United States Rehabilitation Act
 - Section 508, 51-53
 - WAI-ARIA (Accessible Rich Internet Applications), 63-65
 - WCAG (Web Content Accessibility Guidelines), 44-45
 - Priority 1 checkpoints*, 45-46
 - Priority 2 checkpoints*, 47-50
 - Priority 3 checkpoints*, 50-51
- Accessible Rich Internet Applications (ARIA), 63-65
- acronyms, 50
- acronym tag, 50
- addEventListener() function, 70, 74, 83
- addServer() function, 169
- addslashes() function, 248-249
- Advanced PHP Debugger (APD), 179-181
- Ajax Design Patterns*, 87
- AjaxEvent class, 76
- Ajax in Action*, 330
- Ajax Patterns and Best Practices*, 330
- AjaxRequest.prototype.send()
 - function, 261-262
- AjaxRequest class, 77-78
 - abort() function, 79
 - open() function, 79-80
 - send() function, 80-81
 - stateChanged() function, 78-79
 - urlencodeObject() function, 81
- AjaxRequestManager class, 82-83

AjaxRequestManager object, 84-87
AliasPicker() function, 75
Alternative PHP Cache (APC), 166-168
alternative text, 45
animation

- predictive animation, 317-320
- progress indicators
 - advantages*, 20
 - incorporating into interface design*, 20-21
 - pop-up dialogs*, 22
- throbbers
 - definition of*, 17
 - design*, 17
 - Throbber class*, 18-20

apache_request_headers() function, 259-261
Apache webserver, 8
APC (Alternative PHP Cache), 166-168
APCCache class, 167
APD (Advanced PHP Debugger), 179-181
apd_set_pprof_trace() function, 179-181
API (application programming interface)

- documentation**
 - coding standards, 289-293
 - JSDoc, 283-288
 - phpDocumentor, 275-283
 - programming guides, 293-295
 - style guides, 295-296

applications. *See* web applications
architecture. *See* client-side architecture;
server-side architecture
askQuestion() function, 84-85
assertDataSize() function, 273-274
assertEquals() function, 134
assertions, 134-135
asynchronicity, 3
at symbol (@), 279
attachEvent() function, 71

attacks

- CSRF (cross-site request forgeries)**
 - checking Referrers*, 259-261
 - example*, 258-259
 - secondary, random tokens*, 262-265
 - submitting additional headers*, 261-262
- SQL injection**, 247-248
 - filtering*, 249-251
 - magic quotes*, 248-249
 - prepared statements*, 251-252
- XSS (cross-site scripting)**
 - escaping for markup*, 252-254
 - escaping for markup from JavaScript*, 255-256
 - escaping for URLs*, 257-258

authorization, 265-266

B

badRangeRequested() function, 178
bandwidth

- JSON, 154-155
- output compression, 156-157
- plist format, 155
- SOAP, 154
- XML, 155

BBC website, 111
blink tag, 12
book recommendations, 330
bottlenecks, recognizing, 128-131
braces ({}), 291
browsers

- consoles, 112-113
- debugging tools, 111
- Firefox, 116
 - Firebug*, 118-120
 - Venkman*, 120-122
 - Web Developer Extension*, 117-118

- Internet Explorer
 - Developer Toolbar*, 113-115
 - event handling*, 70-71
 - Microsoft Script Debugger*, 115-116
- Opera
 - Web Accessibility Toolbar*, 122-123
 - Web Developer Toolbar and Menu*, 122
- Safari, 124-126
- bruteForce() function**, 189
- C**
- cache**, 160-161
 - Alternative PHP Cache (APC), 166-168
 - Cache class, 161
 - filesystemCache class, 161-163
 - loading, 170-171
 - memcache, 168-170
 - shared memory operations extension (shmop), 164-166
- Cache class**, 161, 278-280
- cachegrind trace file format**, 182-185
- calculatePosition() function**, 320
- calculateRotation() function**, 318
- calculateSpeed() function**, 319-320
- canvas tag**, 305-307, 310
- caption tag**, 48
- Cascading Style Sheets**. *See* CSS
- CentralController class**, 223-225
- CentralController object**, 264
- characters, encoding into numeric entity equivalents**, 253-254
- checkLine() function**, 131
- classes**
 - AjaxEvent, 76
 - AjaxRequest, 77-78
 - abort() function*, 79
 - open() function*, 79-80
 - send() function*, 80-81
 - stateChanged() function*, 78-79
 - urlEncodeObject() function*, 81
- AjaxRequestManager, 82-83
- APCCache, 167
- Cache, 161, 278-280
- CentralController, 223-225
- ColorFade, 195-197
- CustomEvent, 73-75, 133
- Dispatcher, 336
- DVO, 212-219
- EventDispatcher, 78
- ExtendedSample, 72
- filesystemCache, 161-163
- Logger, 29
- Man, 68
- memcacheCache, 169, 278, 281
- Message, 28
- Messenger, 28
- PDOIterator, 192-194
- ProfilerController, 101-104
- RegistrationController, 228-231
- RenderingEngine, 232-233
- Sample, 72
- Session, 219-221
- shmopCache, 164
- TestDispatcher, 135-136
- Throbber, 18-20
- User, 221-222
- View, 237, 240
- Wormhole, 310-312, 316
- XHTMLRenderingEngine, 240-241
- client-side architecture**
 - advantages of architecture, 104-105
 - event-driven application development, 104-105

- event handling, 68-69
 - AjaxRequestManager* object, 84-87
 - DOM, 70
 - EventDispatcher* object, 73-76
 - Internet Explorer*, 70-71
 - JavaScript* object overview, 71-73
 - Man* class example, 68-69
- XMLHttpRequest object, 76-83
- MVC (Model-View-Controller)
 - pattern, 87-88
- ProfileEvent object
 - ProfilerController* class, 101-104
 - ProfileView* object, 95-101
 - sample XHTML page, 93-94
 - UserProfile* object, 89-92
 - View* responsibilities, 92-93
- client-side code, 9
 - architecture. *See* client-side architecture
 - debugging, 108, 111
 - browser consoles*, 112-113
 - Firefox*, 116-122
 - Internet Explorer*, 113-116
 - JavaScript* profiling, 126-131
 - Opera*, 122-123
 - Safari*, 124-126
 - unit testing*, 132-144
 - validation, 108-109
 - CSS Validator*, 110-111
 - Markup Validator*, 109-110
 - semantic extractor*, 111
- coding standards, documenting, 289-293
- ColorFade class, 195-197
- commands
 - pprof2calltree, 182
 - pprofp, 179-180
- compression, output compression, 156-157
- conformance with OpenAjax, 334-337
- consoles (browsers), 112-113
- consolidating resources, 158-159
- ContentLoaded() function, 177
- contentLoaded() function, 176
- contentNotModified() function, 176-178
- contentPartiallyLoaded() function, 178
- content replacement (screen reader support), 54-55
- controllers (MVC pattern)
 - nested controllers, 222-231
 - CentralController* class, 223-225
 - loadController()* function, 227-228
 - RegistrationController* class, 228-231
 - ProfilerController* class, 101-104
- controls, accessibility, 62, 63
- craigslist.org, 6
- Crane, Dave, 330
- createAjaxRequest() function, 82
- createMessenger() function, 310
- createParallelUniverse() function, 314
- createUser() function, 181, 229
- cross-site request forgeries. *See* CSRF
- cross-site scripting. *See* XSS
- CSRF (cross-site request forgeries)
 - checking Referrers, 259-261
 - example, 258-259
 - secondary, random tokens, 262-265
 - submitting additional headers, 261-262
- CSS (Cascading Style Sheets), 9
 - accessibility, 48
 - hacks, 109
 - usability, 37, 40-41
 - Validator, 110-111
- css-discuss wiki, 331
- CustomEvent class, 73-75, 133

D**databases, 146**

- design considerations for future
 - development, 323
- indexes, 148-150
- MySQL Database Server, 8
- normalization
 - first normal form, 147*
 - second normal form, 147*
 - third normal form, 148*
- queries, 150-152

debugging client-side code, 108

- browser consoles, 112-113
- browser tools, 111
- Firefox, 116
 - Firebug, 118-120*
 - Venkman, 120-122*
 - Web Developer Extension, 117-118*
- Internet Explorer
 - Developer Toolbar, 113-115*
 - Microsoft Script Debugger, 115-116*
- JavaScript profiling, 126
 - bottlenecks, recognizing, 128-131*
 - output, 127*
- Opera
 - Web Accessibility Toolbar, 122-123*
 - Web Developer Toolbar and Menu, 122*
- Safari, Drosera, 124-126
- unit testing, 132-133
 - assertions, 134-135*
 - mock objects, 140-143*
 - test example, 137-139*
 - test setup, 135-137*
 - test suites, 143-144*

validation, 108-109

- CSS Validator, 110-111*
- Markup Validator, 109-110*
- semantic extractor, 111*

delete() function, 215**deleteCache() function, 165, 168-169****design**

- accessibility
 - easily targeted controls, 62-63*
 - high-contrast design, 59-60*
 - zoomable interfaces, 60-62*
 - event-driven application
 - development, 104-105
 - factory pattern, 237, 240-241
 - future development, 323-324
 - multiple interfaces, 208-211
 - MVC (Model-View-Controller)
 - pattern, 87-88
 - DBO class, 212-215, 218-219*
 - nested controllers, 222-231*
 - ProfileEvent object, 95*
 - ProfilerController class, 101-104*
 - ProfileView object, 95-101*
 - rendering engines, 232-234*
 - sample XHTML page, 93-94*
 - Session class, 219-221*
 - templates, 234-237*
 - User class, 221-222*
 - UserProfile object, 89-92*
 - View responsibilities, 92-93*
 - throbbers, 17
 - usability, importance of, 322-323
- detachEvent() function, 71**
- Developer Toolbar (IE), 113-115**
- development**
- documentation. *See* documentation
 - future development, 324

games, 298

double-buffering with canvas tag, 305-307, 310

event-source HTML5 element, 315-317

predictive animation, 317-320

server-side logic, 302-304

streaming response, 310-315

validation, 300-301

dialog box progress indicators, 22

disabled users. *See* accessibility

Dispatcher class, 336

dispatchEvent() function, 70, 74

display() function, 225, 230, 233

docBuilder (phpDocumentor), 276-277

documentation

benefits of, 272

eliminating points of failure, 274-275

jogging your own memory, 272-274

lessening learning curve, 274

coding standards, 289-293

JSDoc, 283-288

phpDocumentor, 275-283

programming guides, 293-295

style guides, 295-296

Dojo ShrinkSafe, 159

DOM event handling, 70

DOM Level 2 Document Object Model

Events Technical Report, 70

double-buffering with canvas tag,
305-307, 310

drawShip() function, 318

Drosera, 124-126

DVO class, 212-219

E

ECMA-262 standard (footnote), 3

Ecma International (footnote), 3

ECMAScript, 3

Eichorn, Joshua, 330

eliminate() function, 91

eliminateAjaxRequest() function, 82

eliminated() function, 91

encodeURIComponent() function, 257

encoding characters into numeric entity
equivalents, 253-254

error messages, 46

escape() function, 241

escapeHTML() function, 255

escapeIdentifier() function, 218

escapeIdentifiers() function, 218

escapeMarkup() function, 254

escapeTable() function, 218

escaping (XSS attacks)

for markup, 252-254

for markup from JavaScript, 255-256

for URLs, 257-258

event-driven application

development, 104-105

event-source HTML5 element, 315-317

EventDispatcher() function, 284-287

EventDispatcher object, 73-78

event handling, 68-69

DOM, 70

event listeners, 56-58

Internet Explorer, 70-71

JavaScript objects, 71-73

AjaxRequestManager, 84-87

EventDispatcher, 73-76

XMLHttpRequest, 76-83

Man class example, 68-69
OpenAjax, 338-340
expectations of users, 16-17
EXPLAIN statement, 150-152
ExtendedSample class, 72
eXtensible Markup Language. *See* XML

F

factory pattern, 237, 240-241
fail() function, 134
fakeThread() function, 190
Fangs, 53
Fawcett, Joe, 330
feedback, importance of, 17
Fiddler, 119
filesystemCache class, 161-163
filtering, 249-251
filterRequest() function, 229
Firebug, 12-13, 118-120
Firefox, 116
 Firebug, 118-120
 Venkman, 120-122
 Web Developer Extension, 117-118
Fire Vox, 54
first normal form, 147
forms
 accessibility, 48-50
 screen reader support, 55-56
Freedom Scientific Jaws, 53
functions and methods
 abort(), 79
 abortAll(), 83, 338-339
 addEventListener(), 70, 74, 83
 addServer(), 169
 addslashes(), 248-249
 AjaxRequest.prototype.send(), 261-262

AliasPicker(), 75
apache_request_headers(), 259-261
apd_set_pprof_trace(), 179-181
askQuestion(), 84-85
assertDataSize(), 273-274
assertEquals(), 134
assertion functions (JUnit), 134
attachEvent(), 71
badRangeRequested(), 178
bruteForce(), 189
calculatePosition(), 320
calculateRotation(), 318
calculateSpeed(), 319-320
checkLine(), 131
ContentLoaded(), 177
contentLoaded(), 176
contentNotModified(), 176-178
contentPartiallyLoaded(), 178
createAjaxRequest(), 82
createMessenger(), 310
createParallelUniverse(), 314
createUser(), 181, 229
delete(), 215
deleteCache(), 165, 168-169
detachEvent(), 71
dispatchEvent(), 70, 74
display(), 225, 230, 233
drawShip(), 318
eliminate(), 91
eliminateAjaxRequest(), 82
eliminated(), 91
encodeURIComponent(), 257
escape(), 241
escapeHTML(), 255
escapeIdentifier(), 218
escapeIdentifiers(), 218

escapeMarkup(), 254
escapeTable(), 218
EventDispatcher(), 284-287
fail(), 134
fakeThread(), 190
filterRequest(), 229
generateToken(), 263
generateValidationToken(), 226, 263
get(), 213, 220
getCache(), 165-169
getDatabaseHandle(), 225
getHeader(), 224
getMethodFromRequest(), 239
getQueue(), 28
getRenderingEngine(), 239
getUserInfo(), 230
handleRequest(), 223, 228
hashWithSalt(), 210
highlightElementAfterReplacement(), 55
hitTest(), 302
howManyRoadsMustAManWalkDown(),
 137-139
htmlentities(), 253
include(), 198-199
init(), 100-102
insert(), 216
itClicked(), 73
json_encode(), 255
jsUnitTestSuite(), 144
load(), 89, 216, 221
loadController(), 224, 227, 263
loadDatabase(), 225
loadJavaScript(), 198
loadRenderingEngine(), 239
loadUser(), 223
loadValues(), 90
logGlobalAjaxEvents, 339
mb_encode_numericentity(), 253
md5(), 209
meetsFieldConstraints(), 214
MessageOutput(), 24-25
messageSent(), 312
Messenger(), 25-26
modelDeleted(), 102
modelLoaded(), 102
modelSaved(), 102
mysqli_real_escape_string(), 250
NameEnteredEvent(), 75
nextStep(), 99
Night(), 336
notifyOfElementChanges(), 55-56
ob_gzhandler(), 157
ob_start(), 156
open(), 79-80
parseJSON (), 268
passTheBuck(), 224
preg_replace(), 255
prepare(), 311
prepareElementForReplacement(), 55, 64
presentAnswer(), 84-85
profile(), 129
profileEnd(), 129
rawurlencode(), 257
regenerate(), 220
registerLibrary(), 337
removeEventListener(), 70, 74, 83, 139
replaceAndFocusElement(), 55
Rule(), 305-307
save(), 90, 215
saved(), 91
select(), 34-36, 217
send(), 80-81
sendHeaders(), 234, 240
sendSnapshot(), 312

set(), 209-210, 213, 220
 setAssoc(), 214
 setCache(), 164, 167-169, 280
 setContext(), 232
 setElementText(), 54, 256
 setTemplate(), 233
 setTemplatesDirName(), 233
 setUp(), 137
 setVariable(), 233
 sha1(), 209
 stateChanged(), 78-79, 172-174
 submit(), 99
 suite(), 144
 syncFromUI(), 97
 syncToUI(), 98
 tearDown(), 137
 testCustomEvent(), 133
 testEventDispatcher(), 138-139
 Throbber(), 18
 Universe(), 294
 update(), 215
 updateContent(), 175-177
 updateUniverse(), 314, 317
 urlEncodeObject(), 81, 128-130, 257-258
 useClass(), 190
 validateHeader(), 226
 validatePost(), 226
 validateToken(), 227
 viewDeleted(), 103
 viewDisplayed(), 103
 viewSaved(), 103
 wakeUp(), 68
 Watcher(), 75
 WhatDoYouGetWhenYouMultiplySix-
 ByNine(), 135-136, 291-292
**future development, considering in
 design, 323-324**

G

game development, 298
 double-buffering with canvas tag,
 305-307, 310
 event-source HTML5 element, 315-317
 predictive animation, 317-320
 server-side logic, 302-304
 streaming response, 310-315
 validation, 300-301
generateToken() function, 263
generateValidationToken()
 function, 226, 263
get() function, 213, 220
getCache() function, 165-169, 280
getDatabaseHandle() function, 225
getHeader() function, 224
getMethodFromRequest() function, 239
getQueue() function, 28
getRenderingEngine() function, 239
getUserInfo() function, 230
Gross, Christian, 330

H

hacks (CSS), 109
handleRequest() function, 223, 228
hashWithSalt() function, 210
headers (HTTP/1.1)
 If-Modified-Since, 174-176
 Range, 176-178
 submitting additional headers, 261-262
high-contrast design, 59-60
highlightElementAfterReplacement()
 function, 55
hitTest() function, 302
howManyRoadsMustAManWalkDown()
 function, 137-139

HTML tags, 12

HTML5 event-source element, 315-317

htmlentities() function, 253

HTTP

security problems, 245-246

HTTP/1.1 specification

If-Modified-Since header, 174-176

Range header, 176-178

stateChanged() function, 171-174

HTTPS, 244

clear text HTTP security

problems, 245-246

performance loss, 247

I

IE. *See* Internet Explorer

If-Modified-Since header, 174-176

images

alternative text, 45

progress indicators

advantages, 20

incorporating into user interface

design, 20-21

pop-up dialogs, 22

throbbers

definition of, 17

design, 17

Throbber class, 18-20

include() function, 198-199

indenting code, 292

indexes, 148-150

indicators

importance of, 17

inline messages, 22-23

client-side message queue, 25-26

client-side output management, 23-25

example, 23

server-side message queue, 27-30

server-side output management, 26-27

progress indicators

advantages, 20

incorporating into user interface

design, 20-21

pop-up dialogs, 22

throbbers

definition of, 17

design, 17

Throbber class, 18-20

init() function, 100-102

inline messages, 22-23

client-side message queue, 25-26

client-side output management, 23-25

example, 23

server-side message queue, 27-30

server-side output management, 26-27

insert() function, 216

interfaces

design considerations for future

development, 323

late loading, 198-199

modularity, 195-198

multiple interfaces, designing

applications for, 208-211

zoomable interfaces, 60-62

internal developer documentation, 288

coding standards, 289-293

programming guides, 293-295

style guides, 295-296

Internet Explorer

Developer Toolbar, 113-115

event handling, 70-71

Microsoft Script Debugger, 115-116

itClicked() function, 73

J**JavaScript, 3-4**

- classes. *See* classes
- escaping for markup from, 255-256
- event handling, 71-73
 - AjaxRequestManager* object, 84-87
 - EventDispatcher* object, 73-76
 - XMLHttpRequest* object, 76-83
- JsUnit, 132-133
 - assertions, 134-135
 - mock objects, 140-143
 - test example, 137-139
 - test setup, 135-137
 - test suites, 143-144
- profiling, 126
 - bottlenecks, recognizing, 128-131
 - output, 127
- resource consolidation, 158-159
- Unicode support, 267
- usability, 37, 40-41

JavaScript Object Notation. *See* JSON**Jaws, 53****JSDoc, 283-288****JSON (JavaScript Object Notation), 5**

- bandwidth usage, 154-155
- security, 267-269

json_encode() function, 255**jsUnitTestSuite() function, 144****K-L****KCachegrind, 182-185****Koch, Peter-Paul, 331****late loading, 198-199****latency**

- request queuing, 160
- resource consolidation, 158-159

learning curve, lessening with**documentation, 274****listeners, 56****load() function, 89, 216, 221****loadController() function, 224, 227, 263****loadDatabase() function, 225****loading****cache classes, 170-171****functions***load()*, 89, 216, 221*loadController()*, 224, 227, 263*loadDatabase()*, 225*loadJavaScript()*, 198*loadRenderingEngine()*, 239*loadUser()*, 223*loadValues()*, 90**late loading, 198-199****loadJavaScript() function, 198****loadRenderingEngine() function, 239****loadUser() function, 223****loadValues() function, 90****Logger class, 29****logGlobalAjaxEvents function, 339****luminosity contrast ratios, 59****Lynx, 53****M****magic quotes, 248-249****Man class, 68****Markup Validator, 109-110****marquee tag, 12****MathML, 47****mb_encode_numericentity() function, 253****McPeak, Jeremy, 330****md5() function, 209****meetsFieldConstraints() function, 214****memcache, 168-170**

memcacheCache class, 169, 278, 281

memory

cache, 160-161

Alternative PHP Cache (APC), 166-168

Cache class, 161

class loading, 170-171

filesystemCache class, 161-163

memcache, 168-170

shared memory operations extension (shmop), 164-166

scalability of web applications, 191-194

Message class, 28

MessageOutput() function, 24-25

messages

error messages:accessibility, 46

inline messages, 22-23

client-side message queue, 25-26

client-side output management, 23-25

example, 23

server-side message queue, 27-30

server-side output management, 26-27

messageSent() function, 312

Messenger() function, 25-26

Messenger class, 28

methods. *See* functions and methods

Microsoft Script Debugger (IE), 115-116

mock objects, 140-143

Model-View-Controller pattern. *See*

MVC pattern

modelDeleted() function, 102

modelLoaded() function, 102

models (MVC pattern)

DVO class, 212-215, 218-219

ProfileEvent object, 95

ProfileView object, 95-101

Session class, 219-221

User class, 221-222

UserProfile object, 89-92

modelSaved() function, 102

modularity, 195-198

Mozilla

Developer Center, 331

Spidermonkey, 3

multiplayer games

event-source HTML5 element, 315-317

predictive animation, 317-320

streaming response, 310-315

MVC (Model-View-Controller)

pattern, 87-88

DBO class, 212-215, 218-219

nested controllers, 222-231

CentralController class, 223-225

loadController() function, 227-228

RegistrationController class, 228-231

ProfileEvent object, 95

ProfilerController class, 101-104

ProfileView object, 95-101

rendering engines, 232-234

sample XHTML page, 93-94

Session class, 219-221

templates, 234-237

User class, 221-222

UserProfile object, 89-92

View responsibilities, 92-93

MySQL

Database Server, 8

EXPLAIN statement, 150-152

mysqli_real_escape_string()

function, 250

N

NameEnteredEvent() function, 75

namespaces, registering with

- OpenAjax**, 337-338

nested controllers, 222-231

- CentralController class, 223-225
- loadController() function, 227-228
- RegistrationController class, 228-231

nextStep() function, 99

Night() function, 336

normalization

- first normal form, 147
- second normal form, 147
- third normal form, 148

notifyOfElementChanges() function, 55-56

numbers, encoding characters into

- numeric entity equivalents, 253-254

O

ob_gzhandler() function, 157

ob_start() function, 156

objects

- AjaxRequestManager, 84-87
- CentralController, 264
- EventDispatcher, 73-76
- event handling, 71-73
 - AjaxRequestManager*, 84-87
 - EventDispatcher*, 73-76
 - XMLHttpRequest*, 76-83
- mock objects, 140-143
- PDO (PHP Data Objects), 251-252
- ProfileEvent, 95
- ProfileView, 95-101
- UserProfile object, 89-92

- XMLHttpRequest, 76-83
 - abort()* function, 79
 - AjaxEvent* class, 76
 - AjaxRequest* class, 77-78
 - AjaxRequestManager* class, 82-83
 - EventDispatcher* class, 78
 - mock objects*, 140-143
 - open()* function, 79-80
 - send()* function, 80-81
 - stateChanged()* function, 78-79
 - urlEncodeObject()* function, 81

open() function, 79-80

Open Ajax

- conformance, 334-337
- event management, 338-340
- namespace registration, 337-338
- OpenAjax Alliance, 7, 334
- OpenAjax Hub, 334

OpenAjax Alliance, 7, 334

OpenAjax Hub, 334

Opera

- console, 112
- Web Accessibility Toolbar, 122-123
- Web Developer Toolbar and Menu, 122

optimizing performance. *See*
performance optimization

output compression, 156-157

P

@param tag, 280

parseJSON () function, 268

parsing semantic markup, 34-37

Pascarello, Eric, 330

passTheBuck() function, 224

\$password_hash variable, 248

passwords, storing, 248

- PDO (PHP Data Objects), 251-252
- PDOIterator class, 192-194
- performance optimization, 146
 - bandwidth
 - JSON, 154-155
 - output compression, 156-157
 - plist format, 155
 - SOAP, 154
 - XML, 155
 - cache, 160-161
 - Alternative PHP Cache (APC), 166-168
 - Cache class, 161
 - class loading, 170-171
 - filesystemCache class, 161-163
 - memcache, 168-170
 - shared memory operations extension (*shmop*), 164-166
 - databases, 146
 - indexes, 148-150
 - normalization, 147-148
 - queries, 150-152
 - HTTP/1.1 specification
 - If-Modified-Since header, 174-176
 - Range header, 176-178
 - stateChanged() function, 171-174
 - HTTPS, 247
 - latency, 158-159
 - PHP profiling
 - Advanced PHP Debugger (*APD*), 179-181
 - Xdebug, 182-185
- Perl, 283
- PHP, 8
 - and bytes of strings (footnote), 8
 - functions. *See* functions
 - PDO (PHP Data Objects), 251-252
 - profiling
 - Advanced PHP Debugger (*APD*), 179-181
 - Xdebug, 182-185
 - PHP Data Objects (PDO), 251-252
 - phpdoc script (phpDocumentor), 276-277
 - phpDocumentor, 275-283
 - plist format, 155
 - Poe, Edgar Allan, 12
 - pprof2calltree utility, 182
 - pprofp command, 179-180
 - predictive animation, 317-320
 - preg_replace() function, 255
 - prepare() function, 311
 - prepared statements, 251-252
 - prepareElementForReplacement() function, 55, 64
 - presentAnswer() function, 84-85
 - Priority 1 checkpoints (WCAG), 45-46
 - Priority 2 checkpoints (WCAG), 47-50
 - Priority 3 checkpoints (WCAG), 50-51
 - processor usage, 188-191
 - Professional Ajax*, 330
 - profile() function, 129
 - profileEnd() function, 129
 - ProfileEvent object, 95
 - ProfilerController class, 101-104
 - ProfileView object, 95-101
 - profiling
 - JavaScript, 126
 - bottlenecks, recognizing, 128-131
 - output, 127
 - PHP
 - Advanced PHP Debugger (*APD*), 179-181
 - Xdebug, 182-185
 - programming guides, 293-295

progress indicators
 advantages, 20
 incorporating into user interface
 design, 20-21
 pop-up dialogs, 22

Q

QA Toolbox, 109
q tag, 48
queries, 150-152
queues, request queuing, 160
Quirks Mode, 331
quotes, magic, 248

R

rainbow tables, 248
Range header, 176-178
raw text, 4
rawurlencode() function, 257
readability of semantic markup, 33-34
readers, screen
 compatibility with Ajax, 53-54
 content replacement, 54-55
 form validation, 55-56
real-time multiplayer games
 event-source HTML5 element, 315-317
 predictive animation, 317-320
 streaming response, 310-315
Referers, checking, 259-261
regenerate() function, 220
regenerating session ID tokens on login, 246
registering
 namespaces with OpenAjax, 337-338
 users, 234-236
registerLibrary() method, 337
RegistrationController class, 228-231
registration interface, 14-15

removeEventListener() function, 70,
 74, 83, 139
RenderingEngine class, 232-233
rendering engines, 232-234
replaceAndFocusElement() function, 55
**replacing content (screen reader
 support)**, 54-55
requests, queueing, 160
@require tag, 286
resource consolidation, 158-159
resources
 books, 330
 websites, 330-331
responses, streaming, 310-315
@return tag, 280
Rule() function, 305-307

S

Safari
 console, 113
 Drosera, 124-126
salts, 248
Sample class, 72
save() function, 90, 215
saved() function, 91
scalability
 late loading, 198-199
 memory usage, 191-194
 modularity, 195-198
 processor usage, 188-191
schemas (database), 146
 indexes, 148-150
 normalization
first normal form, 147
second normal form, 147
third normal form, 148
 queries, 150-152

- screen magnifiers, 61
- screen readers, 31
 - compatibility with Ajax, 53-54
 - content replacement, 54-55
 - form validation, 55-56
- scripting
 - client-side scripting, 9
 - phpdoc (phpDocumentor), 276-277
 - server-side scripting, 8
- second normal form, 147
- Section 508 (United States Rehabilitation Act), 51-53
- security, 244
 - CSRF (cross-site request forgeries)
 - checking Referers*, 259-261
 - example*, 258-259
 - secondary, random tokens*, 262-265
 - submitting additional headers*, 261-262
 - games, 299
 - server-side logic*, 302-304
 - validation*, 300-301
 - HTTPS, 244
 - clear text HTTP security problems*, 245-246
 - performance loss*, 247
 - JSON, 267-269
 - servers, 266-269
 - SQL injection, 247-248
 - filtering, 249-251
 - magic quotes, 248-249
 - prepared statements, 251-252
 - user authorization, 265-266
 - XSS (cross-site scripting)
 - escaping for markup*, 252-254
 - escaping for markup from JavaScript*, 255-256
 - escaping for URLs*, 257-258
- @see tag, 287
- select() function, 34-36, 217
- semantic extractor, 111
- semantic markup, advantages of
 - accessibility, 30-31, 47
 - ease of use, 32
 - parsing, 34-37
 - readability and maintenance, 33-34
- send() function, 80-81
- sendHeaders() function, 234, 240
- sendSnapshot() function, 312
- server-side architecture, 208
 - factory pattern, 237, 240-241
 - multiple interfaces, designing applications for, 208-211
 - MVC (Model-View-Controller) pattern, 212
 - DBO class*, 212-215, 218-219
 - nested controllers*, 222-231
 - rendering engines*, 232-234
 - Session class*, 219-221
 - templates*, 234-237
 - User class*, 221-222
- server-side logic, 302-304
- server-side scripting, 8
- servers
 - Apache webserver, 8
 - MySQL Database Server, 8
 - security, 266-269
- Session class, 219-221
- session ID tokens, regenerating on login, 246
- set() function, 209-210, 213, 220
- setAssoc() function, 214
- setCache() function, 164, 167-169, 280
- setContext() function, 232
- setElementText() function, 54, 256
- setTemplate() function, 233
- setTemplatesDirName() function, 233

setUp() function, 137
setVariable() function, 233
sha1() function, 209
shmop (shared memory operations extension), 164-166
shmopCache class, 164
ShrinkSafe, 159
SOAP, 154
Spidermonkey, 3
SQL injection, 247-248

- filtering, 249-251
- magic quotes, 248-249
- prepared statements, 251-252

stateChanged() function, 78-79, 172-174
statements

- EXPLAIN, 150-152
- prepared, 251-252

storing passwords, 248
streaming response, 310-315
style guides, 295-296
style sheets. *See* **CSS (Cascading Style Sheets)**
submit() function, 99
submitting additional headers, 261, 262
suite() function, 144
syncFromUI() function, 97
syncToUI() function, 98

T

tables

- accessibility, 48
- rainbow tables, 248

table tag, 48
tags

- abbr, 50
- acronym, 50
- blink, 12

canvas, 305-307, 310
 caption, 48
 event-source, 315-317
 marquee, 12
 @param, 280
 q, 48
 @require, 286
 @return, 280
 @see, 287
 table, 48
tearDown() function, 137
templates, 234-237

- factory pattern, 237, 240-241
- user registration page template, 234-236

testCustomEvent() function, 133
TestDispatcher class, 135, 136
testEventDispatcher() function, 138-139
testing, unit, 132-133

- assertions, 134-135
- mock objects, 140-143
- test example, 137-139
- test setup, 135-137
- test suites, 143-144

test suites (unit testing), 143-144
text, alternative, 45
third normal form, 148
Throbber() function, 18
Throbber class, 18-20
throbbbers

- definition of, 17
- design, 17
- Throbber class, 18-20

tokens

- secondary, random tokens, 262-265
- session ID tokens, regenerating on login, 246

U

Understanding Ajax, 8, 330

Unicode, JavaScript support for, 267

United States Rehabilitation Act

Section 508, 51-53

unit testing, 132-133

assertions, 134-135

mock objects, 140-143

test example, 137-139

test setup, 135-137

test suites, 143-144

Universe() function, 294

Universe Conflict, 298

double-buffering with canvas tag,
305-307, 310

event-source HTML5 element, 315-317

frame rates, 307

predictive animation, 317-320

server-side logic, 302-304

streaming response, 310-315

validation, 300-301

update() function, 215

updateContent() function, 175-177

updateUniverse() function, 314, 317

URL-encoded format, 4

encodeURIComponent() function, 81,

128-130, 257-258

URLs, escaping for, 257-258

usability, 12

CSS (Cascading Style Sheets), 37, 40-41

indicators

importance of, 17

inline messages, 22-30

progress indicators, 20-22

throbbers, 17-20

JavaScript, 37, 40-41

semantic markup, advantages of

accessibility, 30-31

ease of use, 32

parsing, 34-37

readability and maintenance, 33-34

user expectations, 16-17

user registration interface, 14-15

when to use Ajax, 12-15

usability, importance of, 322-323

useClass() function, 190

User class, 221-222

UserProfile object, 89-92

users

authorization, 265-266

expectations, 16-17

importance to design, 322-323

registering: user registration page
template, 234-236

user registration interface, 14-15

users with disabilities. *See* accessibility

V

Valgrind suite, 182

validateHeader() function, 226

validatePost() function, 226

validateToken() function, 227

validation, 108-109

CSS Validator, 110-111

games, 300-301

Markup Validator, 109-110

screen reader support, 55-56

semantic extractor, 111

variables, \$password_hash, 248

Venkman, 120-122

View class, 237, 240

viewDeleted() function, 103

`viewDisplayed()` function, 103

views (MVC pattern)

rendering engines, 232-234

responsibilities, 92-93

sample XHTML page, 93-94

templates, 234-237

`viewSaved()` function, 103

W

W3C (World Wide Web Consortium), 330

ARIA (Accessible Rich Internet Applications), 63-65

QA Toolbox, 109

WCAG (Web Content Accessibility Guidelines)

Priority 1 checkpoints, 45-46

Priority 2 checkpoints, 47-50

Priority 3 checkpoints, 50-51

`wakeUp()` function, 68

`Watcher()` function, 75

WCAG (Web Content Accessibility Guidelines), 44-45

Priority 1 checkpoints, 45-46

Priority 2 checkpoints, 47-50

Priority 3 checkpoints, 50-51

Web Accessibility Toolbar (Opera), 122-123

web applications

accessibility

easily targeted controls, 62-63

event listeners, 56-58

high-contrast design, 59-60

screen readers, 53-56

United States Rehabilitation Act

Section 508, 51-53

WAI-ARIA (Accessible Rich Internet Applications), 63-65

WCAG (Web Content Accessibility Guidelines), 44-51

zoomable interfaces, 60-62

designing for multiple interfaces, 208-211

documentation

benefits of, 272-275

coding standards, 289-293

JSDoc, 283-288

phpDocumentor, 275-283

programming guides, 293-295

style guides, 295-296

game development, 298

double-buffering with canvas

tag, 305-307, 310

event-source HTML5 element, 315-317

predictive animation, 317-320

server-side logic, 302-304

streaming response, 310-315

validation, 300-301

JSDoc, 283-288

phpDocumentor, 275-283

scalability

late loading, 198-199

memory usage, 191-194

modularity, 195-198

processor usage, 188-191

Web Content Accessibility Guidelines.

See WCAG

Web Developer Extension (Firefox), 117-118

Web Developer Toolbar and Menu (Opera), 122

webservers, Apache, 8

websites. *See also* web applications

craigslist.org, 6

recommended websites, 330-331

`whatDoYouGetWhenYouMultiply-`

`SixByNine()` function, 135-136, 291-292

WHATWG

- event-source HTML5 element, 315-317
- website, 304

wildcards, **, 339

WinCacheGrind, 183-185

Worldwide Web Consortium. *See* W3C

Wormhole class, 310-312, 316

X-Y-Z

Xdebug, 182-185

XHTML, 9

- managing with Views (MVC), 93-94
- semantic markup, advantages of, 30
 - accessibility, 30-31*
 - ease of use, 32*
 - parsing, 34-37*
 - readability and maintenance, 33-34*

tags

- abbr, 50*
- acronym, 50*
- canvas, 305-310*
- q, 48*
- table, 48*

XHTMLRenderingEngine class, 240-241

XML (eXtensible Markup Language), 4-5, 155

XMLHttpRequest object, 76-83

- abort() function, 79
- AjaxEvent class, 76
- AjaxRequest class, 77-78
- AjaxRequestManager class, 82-83
- EventDispatcher class, 78
- mock objects, 140-143
- open() function, 79-80
- send() function, 80-81
- stateChanged() function, 78-79
- urlEncodeObject() function, 81

XSS (cross-site scripting)

- escaping for markup, 252-254
- escaping for markup from
 - JavaScript, 255-256
- escaping for URLs, 257-258

Zakas, Nicholas C., 330

zoomable interfaces, 60-62