

# Index

## A

### A9 10

**abort( ) method 104, 140**

#### abstract classes

XParser 198

BaseFeed 198–200

RSSFeed 201–203

### Active Channels 193

#### ActiveObject class 150

#### ActiveX, XMLHttpRequest 4

**addHandler( ) method 408**

**Additional Information links 91–93**

**age-based promotion, priority queues 141–142**

**agePromote( ) method 142**

**\$.ajax( ) method 123–124**

#### Ajax

A9 and 10

Bitflux Blog 12

Gmail and 8

Google Maps and 9

Google Suggest and 7

history of 2

principles 6

technologies behind 6–7

Yahoo! News and 11

#### Ajax engine 5

#### AJAX library

classes 440–443

code, writing 440–446

enumerations 443–444

interfaces 444–446

namespaces 440–441

#### Ajax.NET Professional 427–428

client page 434–436

CustomerInfo class 431–434

GetCustomerInfo( ) method 431

GetMessage( ) method 428

GetUserFromId( ) method 430

JavaScript wrappers 428–429

marking methods 428

session access 430

type conversion 429–430

#### Ajax.Request object 109

asynchronous property, Prototype 111

events 111–112

Options object 109–110

requestHeaders property, Prototype 110

#### Ajax.Responders object 115–117

#### Ajax.Updater object 113–115

#### AjaxMail

action methods 565–567

AjaxMailAction.php 536–537

AjaxMailAttachment.php 540

AjaxMailbox class 513–515

AjaxMailNavigate.php 537–539

AjaxMailSend.php 539–540

attachments 532–533

callback functions 571–572

communication methods 557–559

configuration file 512–513

data loading 555–556

### **AjaxMail (continued)**

- database tables 511–512
- Douglas Crockford's JSON JavaScript Library 510
- e-mail list 520–524
- event handlers 573
- Helper functions 552–553
- initialization methods 570–571
- JSON-PHP 510
- mailbox 553–554
- navigation methods 567–569
- PHP3Lib 511
- PHPMailer 510
- read messages, marking 534–535
- rendering methods 560–565
- requirements 509
- retrieving e-mail 515–519
- retrieving specific messages 524–528
- sending messages 528–532
- user interface 541–543
  - compose view 548–550
  - folder view 543–546
  - layout 550
  - read view 546–548
- user notification 556–557
- zXml Library 510

**AjaxMailbox class 513–515**

**AjaxSiteSearch class 392–401**

**ajaxStart( ) method 124**

**AjaxStockWatcher class 376**

- automated updates 384
- class constructor 377–378
- error handling 381–383
- polling for data 378–379
- server response 379–381
- updated information 383–384
- user interface 377

**ajaxStop( ) method 124**

**animation, news ticker 214–216**

**architecture, autosuggest textbox 249**

**argument property, Yahoo! Connection Manager 102**

**array literals, JSON 237**

**Array object, priority objects and 128**

**array\_pop( ) function 375**

**ASP.NET, client tools and 438–439**

**ASP.NET AJAX Extensions**

- ASP.NET AJAX Library 440
- classes, registering 441–442
- enumerations, registering 443–444

- initializeBase( ) method 442
- interfaces, registering 444–446
- namespaces 440–441
- registerClass( ) method 442
- registerEnum( ) method 444

client library 438–155

- classes 440
- client tools 438–440

controls 446–447

- Button control 447–448
- Label control 450
- Selector control 448–449
- TextBox control 447

HTTP requests

- GET requests 451–454
- POST requests 454–455

requirements 438

setup 438

Site Search widget

- clearing results 467
- error handling 467
- events 468–470
- interface 459–460
- declaring form 460–461
- searches 462–467

UpdatePanel control

- adding content 456
- adding to page 455–456
- updates, triggering 457–458

**Atom 196–197**

- feeds, parsing 203–205

**autosuggest textbox 246**

- adding suggestions 259
- architecture 249
- autosuggest( ) method 253
- AutoSuggestControl class 250
- classes 250
- client-side component 270–271
- database table 249
- displaying suggestions 259
- drop-down list 258–259
- event handling 253–256
- fast-type support 266–267
- functionality 260–261
- hide suggestions 246
- highlightSuggestion( ) method 256
- HTML 247–248
- init( ) method 264–265

- keyboard controls 246
- keyboard support 261–264
- keyCode property 254
- multiple suggestions 256–258
- requestSuggestions( ) method 255
- server-side component 268–270
- SuggestionProvider class 267–268
- suggestions list 246
- typeahead 246
  - implementing 251–252
- typeAhead( ) method 252–253

**autosuggest( ) method 253**

## B

**BaseFeed class 198–200**

**Berners-Lee, Tim 2**

**Bitflux Blog 12**

**breakpoints, debugging (Fiddler) 347**

**browsers**

- caching, 63
- cross-browser XML 162–163
- XML 159
- XPath 175–177
- XSLT 187–188

**Button control 447–448**

## C

**Cache-Control header 63**

**caching 63**

**callback object, Yahoo! Connection Manager 100–102**

**Cancel Pending Requests pattern 94–95**

**cancel( ) method 139**

**cancelling requests 139–140**

**cancelReply( ) method 568**

**CDF (Channel Definition Format) 193**

**change event 76**

**checkActiveRequests( ) method 136**

**checkMail( ) method 515**

**Chicago Crime 300**

**cleanupEmail( ) method 562**

**clearResults( ) method 401**

**code writing, AJAX library 440–446**

**Comet**

- browser support 294–296
- connection management 296–297
- DivMod 297

- HTTP streaming 274

- file modification 276–277

- Firefox 285–288

- heartbeats 278

- iframes 277–281

- Internet Explorer 282–284

- LiveConnect 288–294

- progressive rendering 274

- request delays 274–276

- Pushlets 297

- server-sent DOM events 291–292

- UI events 292–294

- server-side support 297

- Twisted 297

**comments, New Comment Notifier 86–90**

**communication patterns 65**

**communication techniques**

- hidden frame technique 21

- advantages 36–37

- disadvantages 36–37

- GET requests 22–28

- iFrames 31–33

- iFrame POST requests 33–36

- POST requests 28–31

- patterns 21

**connections, Comet 296–297**

**Content-Type header, POST request, 19**

**controls 446**

- Button 447–448

- Label 450

- Selector 448–449

- TextBox 447

- UpdatePanel 455–456

- triggering updates 457–458

**cookies, images and 53–55**

**createProcessor( ) method 184**

**createRequest( ) function 39**

**createTransport( ) method 134**

**createXHR( ) function 38**

**CSS (Cascading Style Sheets) 3**

## D

**data validation, XML 152**

**database table, autosuggest textbox 249**

**debugging**

- Fiddler 342, 348

- HTTP breakpoints 347

- installation 343

### **debugging (continued)**

interface 344–347

setup 343

Firebug 338–341

installation 338–340

limitations 342

setup 338–340

XHR logging 340

introduction 337

### **decode( ) method 244–245**

### **decoding, JSON 241**

### **deleteMessage( ) method 533**

### **design patterns 65**

### **disconnect( ) method 517**

### **displayFolder( ) method 568**

### **DivMod 297**

### **DOM (Document Object Model) 3**

IE and

node creation 157–158

node insertion 158

node removal 158

node replacement 158

MSXML

white space 152

server-sent events 291–292

UI events 292–294

XML

creating 159

IE 153–156

### **Douglas Crockford's JSON JavaScript Library 510**

### **drop-down list, autosuggest textbox 258–259**

### **DTD (Document Type Definition), XML documents and 152**

### **DWR (Direct Web Remoting) 416–419**

client page 423

converters 426

createStatement( ) method 421

CustomerInfo class 419–422

dwr.xml 422–423

executeQuery( ) method 421

excluding methods 424

getCustomerInfo( ) method 419

getInstance( ) method, 424

newInstance( ) method 420

scope attribute 425–426

scripts, creating 424–425

Spring framework 425

### **dynamic HTML 3**

### **dynamic script loading 59–62**

### **dynamically created iframes 281**

## **E**

### **e-mail (AjaxMail)**

attachments, 532–533

list 520–524

marking messages read 534–535

retrieving 515–519

retrieving specific messages 524–528

sending 528–532

### **Eich, Brendan 2**

### **emptyTrash( ) method 534**

### **encode( ) method 243–244**

### **encoding, JSON 241**

### **enumerations, AJAX library 443–444**

### **error handling**

FooReader.NET 496–497

JPSpan 412

Site Search widget 467

XML data, IE and 158–159

### **ESPN, Periodic Refresh 85**

### **event handling**

AjaxMail 573

autosuggest textbox 253–256

request responses 39

Site Search widget 468–469

### **events**

Ajax.Request object (Prototype) 111–112

change 76

Google Maps API 311–313

onchange 76

Yahoo! Maps API 327–328

### **Expires header 63**

### **explode( ) function 375**

### **expressions, jQuery 117–118**

## **F**

### **failure( ) method 100**

### **fast-type, autosuggest textbox 266–267**

### **fetch on demand 66**

### **Fiddler 342, 348**

Auth view 346

Forms view 346

Headers view 346

Hex view 346

HTTP breakpoints 347  
 HTTP proxy 342  
 incoming data 348  
 interface 344–347  
 installation 343  
 interface 344–347  
 outgoing data 348  
 Raw view 346  
 Session Inspector 345  
 setup 343  
 TextView 346  
 XML view 346

**fields, validation, incremental 82–85****filetime( ) method 277****file uploads, Yahoo! Connection Manager 105–106****Firebug 338, 341**

headers 342  
 incoming data 341  
 installation 338–340  
 interface 339–340  
 limitations 342  
 outgoing data 341  
 setup 338–340  
 XHR logging 340

**Firefox, HTTP streaming and 285–288****fooReader object 482****FooReader.NET**

error handling 496–497  
 feed, caching 503–506  
 feeds pane 472, 478  
 headers 498  
 items pane 474, 479  
 proxy 498  
 reading pane 474, 480  
 remote feed 499–503  
 setup 506–507  
 testing 506–507  
 user interface 472–475  
   clearing items 489  
   feed methods 490  
   feeds 491  
   feeds list 491–494  
   fooReader object 482  
   getStyle( ) function 481  
   heading information 489  
   Helper functions 481  
   initializing 482–485  
   items pane 486–489  
   loading cue 476

loading cues, showing/hiding 486  
 loading specific feeds 490  
 panes 476–481  
 reading pane content 486  
 selecting items 490  
 TopBar 475–476

**forms**

validation, incremental 76–82  
 Yahoo! Connection Manager 104

**frames, HTML 3**

hidden 3, 21  
 advantages/disadvantages 36–37  
 GET requests 22–28  
 POST requests 28–31

iframes 4

**frameworks**

Ajax.NET Professional 427–428  
 client page 434–436  
 CustomerInfo class 431–434  
 JavaScript wrappers 428–429  
 marking methods 428  
 session access 430  
 type conversion 429–430  
 DWR 416–419  
   client page 423  
   converters 426  
   CustomerInfo class 419–422  
   dwr.xml 422–423  
   excluding methods 424  
   scope attribute 425–426  
   scripts, creating 424–425  
   Spring framework 425  
 JPSpan 407–410  
   client page 414–415  
   CustomerInfo class 413–414  
   error handling 412  
   server page 414  
   type translation 410–412  
 remote invocation frameworks 407

**FreeThreadedDOMDocument class 184****functions**

array\_pop( ) 375  
 createRequest( ) 39  
 createXHR( ) 38  
 explode( ) 375  
 getCookie( ) 55  
 getRequestBody( ) 48, 552  
 getURLForPage( ) 69  
 imagejpeg( ) 52

## functions (continued)

- init( ) 169
- loadAndRender( ) 565
- selectSingleNode( ) 174
- sendRequest( ) 46
- validateField( ) 77, 83

## G

### Garrett, Jesse James 2

#### geocoding 300

- geocoder.us 301
- services
  - GoogleMaps Geocoding Service 301
  - Yahoo! Maps Geocoding Service 301
- Travel GIS 301
- Web sites 300
- WorldKit GeoCoder 301

#### GET request example, Prototype 112–113

#### GET request, HTTP 18

#### GET requests 451–454

- hidden frames 22–28
- jQuery 118–120
- XHR 43–45
- Yahoo! Connection Manager 106–107

#### getAllResponseHeaders( ) method 42

#### getAttachment( ) method 532

#### getAttribute( ) method 166

#### \_getCachedWeather( ) method 354

#### getCookie( ) function 55

#### getEmails( ) method 516

#### getFeed( ) method 205

#### getFolderPath( ) method 520

#### getMessage( ) method 409, 524

#### getMessage( 0 method 524

#### getRequestBody( ) function 48, 552

#### getResponseHeader( ) method 42

#### getUnreadCount( ) method 521

#### getURLForPage( ) function 69

#### GetWeather( ) method 359

#### \_getWebWeather( ) method 354

#### GLargeMapControl 304

#### GMail 8

- Periodic Refresh 85

#### GMapTypeControl 304

#### Google Maps 9, 299

## Google Maps API

- controls 304–305

- GLargeMapControl 304
  - GMapTypeControl 304
  - GOverviewMapControl 304
  - GSMailMapControl 304
  - GSmallZoomControl 304
  - GScaleControl 304

- events 311–313

- addListener( ) method 311

- GMap 2 303

- addmaptypes event 312

- addoverlay event 312

- click event 312

- clearoverlays event 312

- drag event 312

- dragend event 312

- dragstart event 312

- infowindowclose event 312

- infowindowopen event 312

- load event 312

- maptypeschanged event 312

- mousemove event 313

- mouseout event 313

- mouseover event 313

- move event 313

- moveend event 313

- movestart event 313

- removemaptypes event 313

- removeoverlay event 313

- zoomend event 313

- info windows 306–311

- basic info windows 307

- configuration options 307

- getInfoWindow( ) method 310

- getPixelOffset( ) method 311

- getPoint( ) method 311

- map blowups 309–310

- openInfoWindow( ) method 307

- openInfoWindowHtml( ) method 307

- openInfoWindowTabs( ) method 308

- openInfoWindowTabsHtml( ) method 308

- selectTab( ) method 311

- tabbed info windows 307–309

- limitations 302

- map blowups 309

- map overlays 313–321
  - clearOverlays( ) method 313
  - GMarker object 314
  - Glcon object 314–315
  - icons 314–315
  - marker events 317–318
  - marker info windows 316
  - marker manager 318–319
  - markers 314
  - polylines 319–321
  - removeOverlay( ) method 313
- moving map 306
- overview 301–303
- showMapBlowup( ) method 309
- startup 302

**Google Maps Geocoding Service 301****Google Suggest 7, 246**

- Submission Throttling and 74

**GOverviewMapControl 304****GSMallMapControl 304****GSmallZoomControl 304****GScaleControl 304****H****handleResponse( ) method 379****headers**

- Cache-Control 63
- Expires 63
- FooReader.NET 498

**height property, images 55****Helper functions, AjaxMail 552–553****hidden frames 21**

- advantages/disadvantages 36–37
- GET requests 22–28
- POST requests 28–31

**hidden iframes 31–33**

- POST requests 33–36

**history of Ajax 2****Housing Maps 300****HTML**

- autosuggest textbox 247–248
- dynamic 3
- frames 3
  - hidden 3, 21
- iframes, hidden 31–33
- news feed and 221–222

**HTML (HyperText Markup Language) 2****HTTP**

- GET request 18
- POST request 18
- requests 18–21
  - GET 451–454
  - POST 454–455
  - synchronous 43
  - XHR objects 39–43
- responses, 20–21
- URL encoding 19

**HTTP streaming 274**

- file modification 276–277
- Firefox and 285–288
- iframes 277–280
  - dynamically created 281
- Internet Explorer and 282–285
- LiveConnect and 288–291
- requests, delays 274–276

**hypertext 2****I****iCurPage variable 69****IE (Internet Explorer)**

- DOM, nodes 157–158
- HTTP streaming and 282–285
- MSXML and 149
- XML, error handling 158–159
- XML data
  - loading 151–152
  - retrieving 156
- XML DOM 153–156
- XPath 172
- XSLT 182–187

**iframes, HTML 4**

- dynamically created 281
- hidden 31–33
  - POST requests 33–36
- HTTP streaming 277–280

**imagejpeg( ) function 52****images**

- advantages/disadvantages 58
- cookies and 53–55
- creating dynamically 50–51
- document.cookie property 53
- dispose( ) method 52

## images (continued)

---

### images (continued)

- downloading 50
- error event 50
- getCookie( ) function 55
- height property 55
- imagedestroy( ) function 52
- imagejpeg( ) function 52
- <img/> element 50
- load event 50
- outputting, PHP and 51–53
- redirecting to, PHP and 51
- SendRequest( ) method 57
- size 55–58
- src attribute 50
- width property 55

### incremental field validation 82–85

### incremental form validation 76–82

### iNextPageToLoad variable 69

### init( ) function 169

### init( ) method 264–265

### iPageCount variable 69

### isCallInProgress( ) method 104

### iWaiteBeforeLoad variable 69

## J

### JavaScript 2

#### jQuery 117

- \$.ajax( ) method 123–124
- complete property 123
- data property 123
- dataType property 123
- error( ) property 123
- success property 123
- type property 123
- url property 123
- \$.post( ) method 120
- ajaxStart( ) method 124
- ajaxStop( ) method 124
- expressions 117–118
- GET requests 118–120
- \$.getIfModified( ) method 119
- \$.getJSON( ) method 119
- \$.getScript( ) method 119
- load( ) method 122–123
- POST requests 120–122

### JSON (JavaScript Object Notation)

- array literals 237
- decode( ) method 244–245

- decoding 241
- encode( ) method 243–244
- encoding 241
- introduction 237
- mixing literals 239–240
- object literals 238–239
- overview 237
- server-side tools
  - JSON-PHP 243
  - libraries 245
- syntax 240–241
- XML comparison 242–243

### JSON-PHP 243, 510

- decode( ) method 244–245
- encode( ) method 243–244

### JSPan 407–410

- addHandler( ) method 408
- client page 414–415
- CustomerInfo class 413–414
- displayClient( ) method 409
- error handling 412
- ErrorHandler.php 412
- example 412–415
- getCustomerInfo( ) method 413, 414
- getMessage( ) method 409
- server( ) method 409
- server page 414
- setMessage( ) method 409
- type translation 410–412
- Types.php 411
- Unserializer.php 411

## L

### Label control 450

#### libraries

- jQuery 117
  - \$.ajax( ) method 123–124
  - \$.post( ) method 120
  - ajaxStart( ) method 124
  - ajaxStop( ) method 124
  - expressions 117–118
  - GET requests 118–120
  - load( ) method 122–123
  - POST requests 120–122
  - MSXML Active X library 149
- Prototype 109
  - Ajax object 109
  - Ajax.Request object 109–112

- Ajax.Responders 115–117
- Ajax.Updater 113–115
- Yahoo! Connection Manager 99
  - argument property 102
  - callback object 100–102
  - file uploads 105–106
  - form interaction 104
  - GET example 106–107
  - POST example 107–108
  - requests 100, 104
  - scope property 102–103
  - timeout property 103
- YUI 99–100

**literals**

- array literals 237
- mixing 239–240
- object literals 238–239

**LiveConnect, HTTP streaming and 288–291****LiveSearch 388****load( ) method 122–123, 151****loadAndRender( ) function 565****loadInfo( ) method 555****loadMessage( ) method 556****loadXML( ) method 151****logoff( ) method 517****M****Magnetic Ajax demo, Periodic Refresh 86****Mahemoff, Michael 6****mailbox, AjaxMail 553–554****map overlays**

- Google Maps API 313–321
  - icons 314–315
  - marker events 317–318
  - marker info windows 316
  - marker manager 318–319
  - markers 314
  - polylines 319–321
- Yahoo! Maps API 328
  - auto-expand windows 330
  - custom 332–333
  - custom images 330
  - GeoRSS support 333–334
  - labels 329
  - marker smart windows 331–332
  - markers 328
  - polylines 332

**maps**

- geocoding 300
  - geocoder.us 301
  - services 301
  - Travel GIS 301
  - Web sites 300
  - WorldKit GeoCoder 301
  - Google Maps API 301–304

**MapQuest 299****mashups 300**

- Chicago Crime 300
- Housing Maps 300

**\$message variable 79****methods**

- \$.ajax( ) 123–124
- \$.post( ) 120
- abort( ) 104, 140
- addHandler( ) 408
- agePromote( ) 142
- ajaxStart( ) 124
- ajaxStop( ) 124
- autosuggest( ) 253
- cancel( ) 139
- cancelReply( ) 568
- checkActiveRequests( ) 136
- checkMail( ) 515
- cleanupEmail( ) 562
- clearResults( ) 401
- createProcessor( ) 184
- createTransport( ) 134
- decode( ) 244–245
- deleteMessage( ) 533
- disconnect( ) 517
- displayFolder( ) 568
- emptyTrash( ) 534
- encode( ) 243–244
- failure( ) 100
- filemtime( ) 277
- getAttachment( ) 532
- getAttribute( ) 166
- getEmails( ) 516
- getFeed( ) 205
- getFolderPage( ) 520
- getMessage( ) 409
- getUnreadCount( ) 521
- GetWeather( ) 359
- handleResponse( ) 379
- init( ) 264–265

### methods (continued)

- isCallInProgress( ) 104
- load( ) 122–123, 151
- loadInfo( ) 555
- loadMessage( ) 556
- loadXML( ) 151
- logout( ) 517
- navigate( ) 567
- onFailure( ) 110
- onSuccess( ) 110
- readMessage( ) 568
- refreshFolder( ) 571
- remove( ) 140
- renderFolder( ) 560
- renderMessage( ) 560
- request( ) 557, 571
- restoreMessage( ) 533
- sendConfirmation( ) 572
- sendMail( ) 528, 571
- sendNext( ) 136
- sendRequest( ) 57
- setForm( ) 105
- setMessage( ) 409
- sleep( ) 274
- success( ) 100
- transform( ) 184
- typeAhead( ) 252–253
- \_getCachedWeather( ) 354
- \_getWebWeather 354

### mixing literals, JSON 239–240

### MSXML 37

- DOM, whitespace 152
- IE and 149

### MSXML Active X library 149

### MSXML DOM document 150

### Multi-Stage Download 143

### Multi-Stage Download pattern 90–91

## N

### namespaces

- AJAX library 440–441
- prefix 173
- resolvers 177–178
- URI 173
- XML 173–175
- xmlns keyword 174
- xparser 197–198

### navigate( ) method 567

### New Comment Notifier 86–90

### news ticker

- adding content 218–220
- adding feeds 216
- animation 214–216
- client-side 211–212
- Content-Control header 211
- Content-Type header 211
- CSS, 221
- dispose( ) method 216
- feeds, adding 216
- HTML structure 221–222
- isset( ) function 211
- news ticker widget 222–223
- NewsTicker class 212–214
- NewsTickerFeed class 217
- poll( ) method 217–218
- polling for new information 217–218
- removing 216–217
- removing data 220–221
- server-side 210–211
- stopTick( ) method 213
- stopping polling 218
- tickerContainer 213
- widget, using 222–223

### nodes, DOM

- creating 157–158
- inserting 158
- removing 158
- replacing 158

## O

### object literals, JSON 238–239

### onchange event 76

### onFailure( ) method 110

### onSuccess( ) method 110

### output, images (PHP) 51–53

## P

### pages, preloading (Predictive Fetch) 66–74

### patterns 21

- Cancel Pending Requests 94–95
- communication patterns 65
- design patterns 65
- Multi-Stage Download 90–91, 142, 143

Periodic Refresh 85–86, 142, 143

poll( ) method 143

prefetch( ) method 143

Predictive Fetch 66, 142

page preloading 66–74

priorities 142–145

Submission Throttling 142, 143

submit( ) method 143

submitPart( ) method 143

Try Again 96–97

### **Periodic Refresh 143**

New Comment Notifier and 86

### **Periodic Refresh pattern 85–86**

#### **PHP, images**

outputting 51–53

redirecting to 51

#### **PHPMailer 510**

#### **POP3Lib 511**

#### **\$.post( ) method, jQuery 120**

#### **POST request example, Prototype 113**

#### **POST requests 454–455**

hidden frames 28–31

hidden iframes 33–36

HTTP, 18

jQuery 120–122

XHR 45–49

Yahoo! Connection Manager 107–108

### **Predictive Fetch 143**

#### **Predictive Fetch pattern 66**

page preloading 66–74

#### **priorities, patterns 142–145**

#### **priority queues**

age-based promotion 141–142

Array object and 128

get( ) method 129

item( ) method 129

peek( ) method 129

put( ) method 129

prioritize( ) method 129

size( ) method 129

sort( ) method 128

#### **PriorityQueue object 128, 133**

#### **progressive rendering 274**

#### **properties**

height 55

readyState 151

width 55

#### **Prototype 109**

Ajax object 109

Ajax.Request object 109

asynchronous property 111

events 111–112

Options object 109–110

requestHeaders property 110

Ajax.Responders object 115–117

Ajax.Updater object 113–115

GET Request example 112–113

onComplete( ) event handler 112

onException( ) event handler 111

onFailure( ) method 110

onInteractive( ) event handler 111

onLoaded( ) event handler 111

onLoading( ) event handler 111

onSuccess( ) method 110

POST Request example 113

#### **proxies, FooReader.NET 498**

#### **Pushlets 297**

## **Q**

#### **queuing requests 133–134**

## **R**

#### **RDF (Resource Description Framework) 193**

readMessage( ) method 568

readyState property 151

redirecting to images, PHP and 51

reflection 407

refreshFolder( ) method 571

remote feed, FooReader.NET 499–503

remote invocation frameworks 407

remove( ) method 140

renderFolder( ) method 560

renderMessage( ) method 560

**Request Description objects, RequestManager object 132–133**

request( ) method 557, 571

#### **RequestManager object 131–132, 145–147**

addLowPriority( ) function 147

addPrefetch( ) function 147

addSubmit( ) function 147

addSubmitPart( ) function 147

age-based promotion 141–142

cancelling requests 139–140

outputFailureResult( ) function 146

## RequestManager object (continued)

---

### **RequestManager object (continued)**

- outputNotModifiedResult( ) function 146
- outputResult( ) function 146
- outputSuccessResult( ) function 146
- queuing requests 133–134
- Request Description objects 132–133
- sending requests 134–139
  - checkActiveRequests( ) method 134
  - initiating 135–136
  - monitoring 136–139

### **requests**

- cancelling 139–140
- queuing 133–134
- sending 134–139
  - initiating 135–136
  - monitoring 136–139
- Yahoo! Connection Manager 104

### **restoreMessage( ) method 533**

### **RSS**

- Web searches and 223
  - client-side 225
  - close( ) method 224
  - createElement( ) method 224
  - customizing widget 232–234
  - drawResultBox( ) method 224
  - populateResults( ) method 230
  - results box 231
  - results interface 227–229
  - results position 226
  - search interface 231
  - server-side 224–225

### **RSS (RDF Site Summary) 193**

### **RSS (Really Simple Syndication) 194**

### **RSS (Rich Site Summary) 194**

### **RSS 0.91 194–195**

### **RSS 1.0 195**

### **RSS 2.0 196**

### **RSS feeds, parsing 201–203**

### **RSSFeed class 201–203**

## **S**

### **scope property, Yahoo! Connection Manager 102–103**

### **scripts**

- advantages/disadvantages 62
- document.write( ) method 59

- dynamic script loading 59–62

- makeRequest( ) function 60

- <script/> element 59

### **searches, Web, RSS and 223–234**

### **SearchResults class 391**

### **Selector control 448–449**

### **selectSingleNode( ) function 174**

### **sendConfirmation( ) method 572**

### **sending requests 134–139**

- initiating 135–136

- monitoring 136–139

### **sendMail( ) method 528, 571**

### **sendNext( ) method 136**

### **sendRequest( ) function 46**

### **sendRequest( ) method 57**

### **serialization 156**

### **server-sent DOM events 291–292**

- UI events 292–294

### **services, geocoding 301**

### **setForm( ) method 105**

### **setMessage( ) method 409**

### **site search widget 388**

- Add( ) method 395

- adding to page 405–406

- building search page 397

- clearResults( ) method 401

- client-side component 398

- AjaxSiteSearch class 398–401

- clearing results 401

- customizing site 403–405

- information processing 402–403

- user interface 398

- XHR request 402

- ExecuteReader( ) method 393

- handleResponse( ) method 402

- Read( ) method 394

- Search( ) method 397

- server-side component 389

- AjaxSiteSearch class 392–395

- database connection string 390

- database information 389–390

- returned data structure 391

- SearchResults class 391

- user interface 459–460

- clearing results 467

- database connection string 460

- error handling 467

- event handling 468–469
- form declaration 460–462
- performing search 462–467

**sleep( ) method 274**

**Spring framework, DWR 425**

**SQL injection attack 26**

**Stock Quote Proxy 372**

- data organization 372–373
- retrieving quotes 373–375

**stock watcher widget 371**

- AjaxStockWatcher class 376
- automated updates 384
- class constructor 377–378
- error handling 381–383
- polling for data 378–379
- server response 379–381
- updated information 383–384
- user interface 377

- custom quotes 385–386

- data cells, adding 380

- error handling 381–383

- file\_get\_contents( ) function 374

- get\_stock\_quotes( ) function 373

- handleResponse( ) method 379

- implode( ) function 374

- insertCell( ) method 380

- poll( ) method 378

- server response 379

- Stock Quote Proxy 372

- data organization 372–373

- retrieving quotes 373–375

- using 387–388

- Yahoo! Finance 371

**Submission Throttling 74, 143**

**success( ) method 100**

**SuggestionProvider class, autosuggest textbox 267–268**

**synchronous requests 43**

## T

**TextBox control 447**

**timeout property, Yahoo! Connection Manager 103**

**transform( ) method 184**

**Try Again pattern 96–97**

**Twisted 297**

**typeahead, autosuggest textbox 251–252**

**typeAhead( ) method 252–253**

## U

**UpdatePanel control 455–456**

- triggering updates 457–458

**URIs (Uniform Resource Identifiers) 2**

**URL encoding 19**

**user interface**

- AjaxMail 541–543

- compose view 548, 550

- folder view 543–546

- layout 550

- read view 546–548

- FooReader.NET 472–475

- clearing items 489

- feed methods 490

- feeds 491

- feeds list 491–494

- fooReader object 482

- heading information 489

- Helper functions 481

- initializing 482–485

- items pane 486–489

- loading cue 476

- loading cues, showing/hiding 486

- loading specific feeds 490

- panes 476–481

- reading pane content 486

- selecting items 490

- TopBar 475–476

- Site Search widget 459–460

- clearing results 467

- database connection string 460

- error handling 467

- event handling 468–469

- form declaration 460–462

- performing search 462–467

**users, predicting actions 66**

## V

**\$valid variable 79**

**validateField( ) function 77, 83**

**validation**

- fields, incremental 82–85

- forms, incremental 76–82

**variables**

- \$message 79

- \$valid 79

### variables (continued)

- iCurPage 69
- iNextPageToLoad 69
- iPageCount 69
- iWaiteBeforeLoad 69

## W

### weather widget 351

- adding to Web page 370–371
- as application 366–367, 370
- client-side component 361
  - getWeather( ) method 361–362
- customizing 362–366
- \_getCachedWeather( ) method 359
- \_getWebWeather( ) method 359
- server-side component 352
  - cached data, reading 354
  - error handling 356–359
  - gathering data from Web 354–356
  - Settings class 352–353
  - versions 359–360
  - WeatherInfo class 353, 360
- Weather.com SDK 351–352

### WeatherInfo class 353

### Web 2.0 14

### Web searches

- RSS and 223
  - client-side 225
  - customizing widget 232–234
  - results box 231
  - results display 230–231
  - results interface 227–229
  - results position 226
  - search interface 231
  - server-side 224–225

### WHATWG (Web Hypertext Application Technology Working Group) 291

### white space, MSXML DOM 152

### widgets

- site search 388, 468–469
  - adding to page 405–406
  - building search page 397
  - client-side component 398–403
  - customizing site 403–405
  - server-side component 389–395
  - user interface 459–467
- stock watcher 371

- AjaxStockWatcher class 376–384
- custom quotes 385–386
- Stock Quote Proxy 372–375
  - using 387–388
- Yahoo! Finance 371
- weather widget 351
  - adding to Web page 370–371
  - as application 366–367, 370
  - client-side component 361–362
  - customizing 362–366
    - server-side component 352–360
  - Weather.com SDK 351–352

### width property, images 55

## X

### XHR (XMLHttpRequest) 37

- advantages/disadvantages 49–50
- browser caching 41
- GET requests 43–45
- getAllResponseHeaders 42
- getResponseHeader( ) method 42
- objects
  - createXHR( ) function 38
  - creating 37–39
- onreadystatechange event handler 40
- POST requests 45–49
- responseText property 41
- responseXML property 41
- send( ) method 40
- XMLHttpRequest class 39
- zXml library, 39

### XHR logging, Firebug 340

### XHR objects, HTTP requests 39–43

### XHTML Live Chat, Periodic Refresh 85

### XML

- browsers 159
  - cross-browser XML 162–163
    - loading data 160–191
    - retrieving data 161–162
- createDocument( ) method 159
- cross-browser 162–163
- data
  - loading to browser 160–161
  - loading in IE 151–152
  - retrieving in other browsers 161
- data retrieval, IE 156
- data validation 152

documents, DTDs 152

## DOM

- attributes property 153
- childNodes property 153, 154
- creating 159
- firstChild property 153, 154
- IE 153–156
- lastChild property 153
- nextSibling property 153
- nodeName property 153
- nodeType property 153
- nodeValue property 153
- object, creating 150–151
- ownerDocument property 153
- ParentNode property 153
- PreviousSibling property 153
- properties 153
- text property 153
- Xml property 153

## IE and

- error handling 158–159
- parseError object 159

JSON comparison 242–243

load( ) method 151

loadXML( ) method 152

loading data 164–165

loading data to IE 151–152

namespace resolvers 177–178

namespaces 173–175

## nodes

- appendChild( ) method 157
- creating 157
- createElement( ) method
- data retrieval 198
- inserting 158
- removing 158
- replacing 158

parsing lists 165–168

## XMLHttpRequest 4

### XParser 197

- abstract classes 198
  - BaseFeed 198–200
  - RSSFeed 201–203
- data retrieval 198
- parsing
  - Atom 203
  - RSS feeds 201–203
- xparser namespace 197–198

## XPath

- browsers 175–177
- cross-browser 178–179
- evaluate( ) method 176
- IE 172
- introduction 170–172
- namespaces
  - prefixes 173
  - resolver 177–178
  - URI 173
  - xmlns keyword 174

## XPathEvaluator objects 176

### XSLT (XSL Transformations)

- addObject( ) method 185
- browsers 187–188
- cross-browser 189
- directives 180
- IE 182–187
- ImportStylesheet( ) method 187
- introduction 180–182
- transformNode( ) method 183
- transformToDocument( ) method 188
- transformToFragment( ) method 188
- XMLSerializer class 188
- XSLTemplate class 183
- XSLTProcessor class 187

## XSLTemplate class 184

## Y

### Yahoo! Connection Manager 99

- argument property 102
- callback object 100–102
- file uploads 105–106
- form interaction 104
- GET example 106–107
- POST example 107–108
- requests 100, 104
- scope property 102–103
- timeout property 103

### Yahoo! Finance 371

### Yahoo! Maps API

- address lookup 334
- controls
  - addPanControl( ) method 324
  - addTypeControl( ) method 324
  - addZoomLong( ) method 324
  - addZoomScale( ) method 324
  - addZoomShort( ) method 324

## Yahoo! Maps API (continued)

---

### Yahoo! Maps API (continued)

- events 327–328
- map overlays 328
  - auto-expand windows 330
  - custom 332–333
  - custom images 330
  - GeoRSS support 333–334
  - labels 329
  - marker smart windows 331–332
  - markers 328
  - polylines 332
- moving maps 325
  - panToLatLon( ) method 325
  - panToXY( ) method 325
- overview 322–323
- smart windows 326
  - showSmartWindow( ) method 326
  - getCenterLatLon( ) method 326
- startup 321–322

### Yahoo! Maps Geocoding Service 301

### Yahoo! News 11

### YUI library (Yahoo! User Interface) 99

- argument property 100 102
- asyncRequest( ) method 104

- callback object 100–101
- failure method 100
- files, uploads 105
- forms 104–105
- GET request, example 106–107
- getAll ResponseHeaders( ) method 101
- getResponseHeader( ) method 101
- POST request, example 107–108
- requests 100 104
- responseText property 100
- responseXML property 101
- scope property 102–103
- setup 100
- status property 101
- statusText property 101
- tld property 101
- timeout property 103

## Z

zXml library 510