

# Index

## SYMBOLS

### & (ampersand)

- &&, JavaScript AND operator, 35–36, 431
- &, JavaScript bitwise AND operator, 431
- &, in XSLT, 235

### \* (asterisk)

- \*=, JavaScript multiplication assignment operator, 430
- \*, JavaScript multiplication operator, 34, 430
- \*, in MySQL, 370

### \ (backslash)

- JavaScript escape character, 433
- in JSON, 352

### ^ (caret), JavaScript bitwise XOR operator, 432

### { } (curly braces)

- in JavaScript, 37
- in JSON, 359

### \$ (dollar sign)

- in JavaScript variables, 33
- \$ ( ) method, Prototype, 417
- \$\$ ( ) method, Prototype, 417
- \$1, . . . , \$9 properties, RegExp object, 463

### . (dot notation for objects), 41, 432

### “ (double quotes)

- enclosing JSON strings, 352
- enclosing tag attributes, 139
- escaping in JavaScript, 433
- in XSLT, 235

### = (equal sign)

- =, JavaScript assignment operator, 34, 430
- ===, JavaScript equal to and same type operator, 35, 431
- ==, JavaScript equal to operator, 35, 431

### ! (exclamation point)

- !=, JavaScript not equal to operator, 35, 431
- !==, JavaScript not exactly equal to operator, 35, 431
- !, JavaScript NOT operator, 35–36, 431

### < (left angle bracket)

- <<, JavaScript left shift operator, 432
- <, JavaScript less than operator, 35, 431
- <=, JavaScript less than or equal to operator, 35, 431
- <, in XSLT, 235

### - (minus sign)

- =, JavaScript decrement assignment operator, 430
- , JavaScript decrement operator, 36–37, 430
- , JavaScript subtraction operator, 34, 430

### () (parentheses)

- in conditional statements, 37
- in event registration (incorrect), 56, 109
- in function calls, 40
- in method calls, 41

### % (percent sign)

- %=, JavaScript modulus assignment operator, 431
- %, JavaScript modulus operator, 35, 430

## + (plus sign)

---

### + (plus sign)

- +, JavaScript addition operator, 34, 430
- +, JavaScript concatenation operator, 34, 432
- +=, JavaScript increment assignment operator, 430
- ++, JavaScript increment operator, 36–37, 430
- ? (question mark), **JavaScript conditional operator**, 432
- > (right angle bracket)
  - >, JavaScript greater than operator, 35, 431
  - >=, JavaScript greater than or equal to operator, 35, 431
  - >>, JavaScript right shift operator, 432
  - >>>, JavaScript zero fill right shift operator, 432
  - >, in XSLT, 235

### ; (semicolon), in **JavaScript**, 32

### ' (single quote)

- enclosing tag attributes, 139
- escaping in JavaScript, 433
- in XSLT, 235

### / (slash)

- //, indicating XPath recursive descent, 259
- //, in JavaScript, 32, 433
- /\* ... \*/ , in JavaScript, 32, 433
- /=, JavaScript division assignment operator, 430
- /, JavaScript division operator, 34, 430
- /, in JSON, 352
- /, preceding XPath document root, 258
- /\*, preceding XPath root element, 258

### ~ (tilde) **JavaScript bitwise NOT operator**, 432

### \_ (underscore), in **JavaScript variables**, 33

### | (vertical bar)

- |, JavaScript bitwise OR operator, 431
- ||, JavaScript OR operator, 35–36, 431

## A

### **A Venture Forth's web site**, 7

<a> tag, 440, 456

abort() method, XMLHttpRequest object, 103

### **About.com web site**, 426

abs() method, Math object, 458

acos() method, Math object, 458

action property, Form object, 452

### **Active Channels**, 324

### **ActiveX controls**, 100

ActiveXObject object, 14

add() method, Select object, 466

addEventListener() method, Document object, 56, 448

**addition operator (+)**, JavaScript, 34, 430

**aggressive caching, controlling**, 110–117

**"Ajax: A New Approach to Web Applications"** (Garrett), 2, 9

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

- advantages of, 17–18
- application model for, 16–17
- applications using, 3–8
- bad examples of, 8
- definition of, 2, 9
- history of, 2
- tasks to use Ajax for, 7–8
- technologies required for, 9
- user and browser requirements for, 20
- when not to use, 18–20

### **Ajax is not cool web site**, 19

Ajax object, Prototype, 416

### **Ajax promise or hype web site**, 19

### **Ajax sucks web site**, 19

alert() method, Window object, 472

**Amazon search engine (A9.com)**, 5

**Amazon Web Services web site**, 181

### **ampersand (&)**

- &&, JavaScript AND operator, 35–36, 431
- &, JavaScript bitwise AND operator, 431
- &, in XSLT, 235

anchor() method, String object, 467

Anchor object, JavaScript, 440–441

anchors property, Document object, 448

**AND operator, bitwise (&)**, JavaScript, 431

**AND operator (&&)**, JavaScript, 35–36, 431

and operator, XPath, 261

**A9.com (Amazon search engine)**, 5

**answers to exercises**, 391–414

### **APIs (application programming interfaces)**

- compared to web services, 181, 199
- definition of, 179, 198
- list of, 198–199

### **apostrophe (')**

- enclosing tag attributes, 139
- escaping in JavaScript, 433
- in XSLT, 235

appCodeName **property**, Navigator **object**, 459  
 <applet> **tag**, 91  
**application model**, 16–17  
**application programming interfaces**. *See* APIs  
 appName **property**, Navigator **object**, 459  
 appVersion **property**, Navigator **object**, 459  
 Area **object**, JavaScript, 441  
 <area> **tag**, 441, 456  
 arguments **property**, Function **object**, 453  
 arguments.length **property**, Function **object**, 453  
**arithmetic functions**, XPath, 260–261  
**arithmetic operators**, JavaScript, 34, 430  
 Array **constructor**, JSON, 354  
**array literals**, JSON, 354  
 Array **object**, JavaScript, 441–442  
 Array **type**, JavaScript, 33–34  
**arrays**, JSON, 352  
**articles**. *See* publications  
 asin() **method**, Math **object**, 458  
 .asmx **files**, 181  
**ASP.NET**  
   advantages of, 97  
   calling applications from JavaScript, 75  
   definition of, 15–16, 73–74  
   deploying applications, 74–75  
   example using, 75–85  
**ASP.NET Ajax (Atlas)**, 423  
**assignment operator (=)**, JavaScript, 34, 430  
**assignment operators**, JavaScript, 430–431  
**asterisk (\*)**  
   \*=, JavaScript multiplication assignment operator, 430  
   \*, JavaScript multiplication operator, 34, 430  
   \*, in MySQL, 370  
**Asynchronous JavaScript and XML**. *See* Ajax  
**asynchronous model**, 17  
 atan() **method**, Math **object**, 458  
 atan2() **method**, Math **object**, 458  
**Atlas (ASP.NET Ajax)**, 423  
**Atom**. *See also* news feeds  
   elements of, 329–330  
   history of, 324  
 attachEvent() **method**, Document **object**, 56, 448

**attribute nodes, DOM node tree**  
   definition of, 49  
   extracting XML data using, 148–152  
**attributes property**, JavaScript, 148  
**audience (users), ability to use Ajax**, 20  
 availHeight **property**, Screen **object**, 465  
 availWidth **property**, Screen **object**, 465  
 avg() **function**, XPath, 261

## B

\b (**backspace character**), JavaScript, 433  
**Back button, disabled by Ajax**, 19  
 back() **method**, History **object**, 454  
**backslash (\)**  
   JavaScript escape character, 433  
   in JSON, 352  
**backspace character (\b)**, JavaScript, 433  
**Basecamp**, 4–5  
*Beginning JavaScript, Second Edition (Wiley)*, 31, 160  
 bgColor **property**, Document **object**, 448  
 big() **method**, String **object**, 467  
**bitwise AND operator (&)**, JavaScript, 431  
**bitwise NOT operator (~)**, JavaScript, 432  
**bitwise operators**, JavaScript, 431–432  
**bitwise OR operator (|)**, JavaScript, 431  
**bitwise XOR operator (^)**, JavaScript, 432  
 blink() **method**, String **object**, 467  
**Blogger**, 324  
**blogs about JavaScript**, 425–426  
 blur() **method**, Window **object**, 472  
 blur **standard()** **method**, Window **object**, 438  
**body element**, DOM family tree, 48  
 <body> **tag**, 471  
 bold() **method**, String **object**, 467  
**BOM (Browser Object Model)**, 42–43  
**bookmarks, broken by Ajax**, 19  
**books**. *See* publications  
 Boolean **object**, JavaScript, 442–443  
 Boolean **type**, JavaScript, 33  
 border **property**, Image **object**, 455  
**Bosworth, Alex (list of Ajax mistakes)**, 8  
**break statement**, JavaScript, 40, 433, 436

### browser events

- bubbling, 54–55
- capturing, 54–55
- definition of, 54
- objects for, 57
- registering, 55–60
- unregistering, 56, 57

### Browser Object Model (BOM), 42–43

### browser objects, 41, 42–43

### browsers

- BOM (Browser Object Model), 42–43
- chrome of, 162
- configuration of, ability to use Ajax based on, 20
- differences between
  - aggressive caching, controlling, 114
  - cross-browser code for, 11–12
  - debugging `responseXML` property, 72
  - event models, 54–55
  - event objects, 57
  - event registration, 56–57
  - retrieving data from `responseXML` property, 72–73
- name of, in `Navigator` object, 42, 459
- performance of, poor, 19–20
- requirements for, 20
- support for XSLT, 237
- version of
  - ability to use Ajax based on, 20
  - in `Navigator` object, 42, 459

### bubbling browser events, 54–55

### builder script, `Scriptaculous`, 418

### built-in objects, 41–42

### Button object, JavaScript, 443–444

## C

### caching, aggressive, 110–117

### callback() function, 69–70

### callback object, YUI, 420

### camel notation, for JavaScript variable names, 33

### camelize() method, Prototype, 416

### captureEvents() method, Window object, 472

### capturing browser events, 54–55

### caret (^), JavaScript bitwise XOR operator, 432

### carriage return character (\r), JavaScript, 433

### Cascading Style Sheets (CSS)

#### applying to XML data, 152–154

#### definition of, 10

#### example using, 388

### case statement, JavaScript, 435–436

### case-sensitivity, of JavaScript, 32

### catch clause, try statement, 158–159, 436

### ceil() method, Math object, 459

### ceiling() function, XPath, 261

### channel element, RSS, 325–326, 328

### charAt() method, String object, 467

### charCodeAt() method, String object, 467

### Checkbox object, JavaScript, 444

### checked property, Checkbox object, 444

### checked property, Radio object, 462

### childNodes property, JavaScript, 145

### chrome, 162

### classes, prototype objects compared to, 45

### className property, JavaScript, 154

### clearInterval() method, Window object, 472

### clearTimeout() method, Window object, 472

### click() standard method, JavaScript, 438

### client-side technology

#### compared to server-side technology, 1–2

#### for database processing, 368

#### DOM as, 11

#### libraries

##### Dojo library, 418–419

##### MochiKit library, 421

##### Prototype library, 416–417, 422

##### Scriptaculous library, 376–380, 417–418

##### Yahoo! User Interface (YUI) library, 419–421

### close() method

#### Document object, 448

#### Window object, 472

### closed property, Window object, 471

### code reuse. See patterns

### colorDepth property, Screen object, 465

### comments element, RSS, 328

### comments, in JavaScript, 32, 433

### comparison operators, JavaScript, 35, 431

### compile() method, RegExp object, 464

### complete property, Image object, 455

### concat() function, XPath, 260

- concat() **method**
    - Array object, 442
    - String object, 467
  - concatenation operator (+), JavaScript, 34, 432**
  - conditional operator (?), JavaScript, 432**
  - conditional statements, JavaScript, 37–38**
  - confirm() **method**, Window **object**, 473
  - constants, JavaScript, 429**
  - constructor **property**, Object **object**, 461
  - constructor **standard property**, Object **object**, 438
  - constructors**
    - definition of, 43–44
    - functions as, 44, 47
  - ContextAgnosticXMLHttpRequest **object**, 198
  - continue **statement**, JavaScript, 40, 434
  - controls, HTML, 64**
  - controls **script**, Scriptaculous, 418
  - cookie **property**, Document **object**, 448
  - cookies, 130–135**
  - copyChildren() **method**, Dojo, 419
  - Core DOM, 47**
  - Core JavaScript 1.5 Guide, web site, 426**
  - Cornford, Richard (article by), 428**
  - cos() **method**, Math **object**, 459
  - count() **function**, XPath, 260
  - Couvreur, Julian (FlashXMLHttpRequest object), 198**
  - CREATE DATABASE **statement**, MySQL, 368
  - CREATE TABLE **statement**, MySQL, 368
  - createAttribute() **method**, Document **object**, 448
  - createComment() **method**, Document **object**, 448
  - createDocument() **method**, Dojo, 419
  - createElement() **method**, Document **object**, 51–53, 448
  - createTextNode() **method**, Document **object**, 51–53, 448
  - Crockford, Douglas**
    - articles by, 428
    - JSON developed by, 351
    - JSONRequest object by, 198
  - CSS (Cascading Style Sheets)**
    - applying to XML data, 152–154
    - definition of, 10
    - example using, 388
  - Curl library, 194–195**
  - curly braces ({})**
    - in JavaScript, 37
    - in JSON, 359
  - current context, XPath, 258**
- ## D
- data retrieval, invisible, 18**
  - database**
    - connection to, creating, 370–371
    - creating, 368
    - definition of, 368
    - deleting records from, 375–376
    - inserting records in, 374–375
    - queries
      - creating, 371–372
      - extracting results from, 372
  - database management system (DBMS), 368**
  - datatypes**
    - JavaScript, 33–34
    - JSON, 352–353
  - Date **object**, JavaScript, 445–447
  - DBMS (database management system), 368**
  - debugging. See errors and exceptions; troubleshooting**
  - decision-making, XSLT, 234–235**
  - decrement assignment operator (--), JavaScript, 430**
  - decrement operator (-), JavaScript, 36–37, 430**
  - defaultChecked **property**
    - Checkbox object, 444
    - Radio object, 462
  - defaultSelected **property**, Option **object**, 461
  - defaultStatus **property**, Window **object**, 471
  - defaultValue **property**
    - Text object, 469
    - Textarea object, 470
  - delete **operator**, JavaScript, 432
  - del.icio.us API, 199**

description **element, RSS, 326**

### design patterns

- definition of, 275, 276
- for drag-and-drop lists, 301–314
- for form validation, 276–284
- for handling errors, 314–320
- for mouseovers, 284–291
- for polling server, 291–301
- web site containing, 276

detachEvent() **method, Document object, 56, 449**

### Diaz, Dustin

- blog by, 425
- “Forget addEvent, use Yahoo’s Event Utility”, 421

disableExternalCapture() **method, Window object, 473**

division assignment operator (/=), **JavaScript, 430**

division operator (/), **JavaScript, 34, 430**

do **loop, JavaScript, 434**

Document **object, JavaScript, 42–43, 48, 447–449**

**Document Object Model. See DOM**

document **property, Window object, 471**

document root, **XPath, 258**

**Document Type Definition (DTD), 139**

documentElement **property, JavaScript, 145, 146**

**Dojo library, 418–419**

dojo.io.bind() **method, Dojo, 418–419**

**dollar sign (\$)**

- in JavaScript variables, 33
- \$() **method, Prototype, 417**
- \$\$() **method, Prototype, 417**
- \$1, . . . , \$9 **properties, RegExp object, 463**

**DOM (Document Object Model)**

- Core DOM, 47
- definition of, 10–11, 47–48
- as family tree, 48
- HTML DOM, 47
- inspectors for
  - Firefox DOM inspector, 168–170
  - IE DOM inspector, 170–171
  - Mouseover DOM Inspector (MODI), 171–172
- as node tree, 48–49
- nodes, adding, 51–54

objects in, accessing, 49–50

parts of, 47

XML DOM, 47

**DOM Scripting (Keith), 425**

domain **property, Document object, 448**

**dot notation for objects, 41, 432**

**double quotes (“)**

- enclosing JSON strings, 352
- enclosing tag attributes, 139
- escaping in JavaScript, 433
- in XSLT, 235

**double slash (//), in JavaScript, 32**

do-while **loop, JavaScript, 39**

**drag and drop, 18**

**drag-and-drop lists**

- definition of, 301
- example using, 302–314
- pattern for, 301–302

dragdrop **script, Scriptaculous, 418**

**DTD (Document Type Definition), 139**

**dynamic menu system, example of, 21–29**

**dynamic script loading**

- advantages and disadvantages of, 130
- definition of, 128
- example using, 128–130

## E

**E constant, Math object, 458**

**eBay API, 199**

**eBay web site, 181**

**ECMA (European Computer Manufacturer’s Association), 31**

**ECMAScript, 31. See also JavaScript**

**ECMA-262 standard, 41–42**

**ECMA-327 standard, 429**

**Edwards, Dean (blog by), 421**

effects **script, Scriptaculous, 418**

**element nodes, DOM node tree, 49**

Element **object, Prototype, 417**

Element.cleanWhitespace() **method, Prototype, 417**

elements **property, Form object, 452**

embeds **property, Document object, 448**

enableExternalCapture() **method, Window object, 473**

enclosure **element, RSS, 328**

- `encodeURIComponent()` **function, JavaScript, 439**
- encoding **property, Form object, 452**
- endpoint, for web service, 181**
- `entry` **element, Atom, 330**
- equal sign (=)**
  - `=`, JavaScript assignment operator, 34, 430
  - `===`, JavaScript equal to and same type operator, 35, 431
  - `==`, JavaScript equal to operator, 35, 431
- `error` **event, 160–162**
- `Error` **object, JavaScript, 159–160**
- errors and exceptions. See also troubleshooting**
  - definition of, 157
  - exception handling, 158–160
  - Firebug for, 166–168
  - Microsoft Script Debugger for, 164–166
  - Mozilla JavaScript console for, 162–164
  - `onerror` event handler, 160–162
  - pattern for, 314–320
- escaping special characters**
  - in JavaScript, 433
  - in JSON, 352–353
  - in XSLT, 235
- European Computer Manufacturer’s Association (ECMA), 31**
- `eval()` **function, JavaScript, 351, 359, 439**
- `EvalError` **type, JavaScript, 160**
- event handlers, JavaScript, 438–439**
- event models, cross-browser differences in, 54–55**
- `Event` **object, JavaScript, 449–451**
- events (browser)**
  - bubbling, 54–55
  - capturing, 54–55
  - definition of, 54
  - objects for, 57
  - registering, 55–60
  - unregistering, 56, 57
- examples**
  - aggressive caching, controlling, 110–117
  - ASP.NET, 75–85
  - dynamic display, 103–109
  - dynamic menu system, 21–29
  - dynamic script loading, 128–130
  - elements, adding to page, 51–53
  - event registration, 57–60
  - extracting text data from news feeds, 337–339
  - extracting XML data
    - from attribute nodes, 149–152
    - from news feeds, 331–337
  - Flickr API, 212–221, 222–225
  - Google Maps API, 202–209
  - hidden frames, 118–124
  - hidden inline frames, 125–128
  - Java servlets, 91–96
  - JSON (JavaScript Object Notation), 360–364
  - mashups, 222–225
  - news feeds
    - extracting text data from, 337–339
    - extracting XML data from, 331–337
    - feed reader for, 339–348
  - objects, creating, 45–47
  - `onerror` event handler, 161–162
  - patterns
    - for drag-and-drop lists, 301–314
    - for form validation, 277–284
    - for handling errors, 314–320
    - for mouseovers, 284–291
    - for polling server, 291–301
  - PHP, 86–91
  - POST method, 115–117
  - shopping cart
    - with ASP.NET, 75–85
    - interactive, 267–273
    - with Java servlets, 91–96
    - with PHP, 86–91
    - with XSLT, 247–256
  - sortable list
    - CSS styles for, 388
    - database for, creating, 368–371
    - database for, editing, 374–376
    - database for, querying, 371–374
    - description of, 367
    - droppable elements, creating, 377–378
    - files used in, 388
    - requests for list updates, 383–387
    - sortable elements, creating, 378–380
    - user interaction with, 380–383
  - web services, consuming
    - with a proxy, 188–195
    - process for, 181–183
    - with script tag hack, 195–197
  - XML data, requesting, 140–144
  - `XMLHttpRequest` object, 103–109, 360–364

### examples (continued)

- XPath, querying XML document, 262–266
- XSLT transform, performing
  - with Firefox, 242–244
  - with IE (Internet Explorer), 237–242

### exclamation point (!)

- !=, JavaScript not equal to operator, 35, 431
- !==, JavaScript not exactly equal to operator, 35, 431
- !, JavaScript NOT operator, 35–36, 431

`exec()` **method**, `RegExp` **object**, 464

### exercises, solutions to, 391–413

`exp()` **method**, `Math` **object**, 459

### expressions, XPath, 257–259

### eXtensible HyperText Markup Language (XHTML), 9–10

### Extensible Stylesheet Language Transformation (XSLT). See also XPath

- browsers supporting, 237
- decision-making structure for, 234–235
- definition of, 12–14, 227, 228–229
- elements of, 230–237
- including files in stylesheet, 231
- looping structure for, 235
- output format for transformation, 230–231
- reasons not to use, 229–230
- reasons to use, 229
- shopping cart example using, 247–256
- sorting, 236
- special characters in, escaping, 235
- style sheets (templates) for
  - creating, 232–233
  - example using, 21–29
  - invoking by match, 232
  - invoking by name, 233
  - parameters for, 233–234
- transform, performing
  - with Firefox, 242–244
  - with IE (Internet Explorer), 237–242
  - on server, with .NET Framework, 245–247
- variables, 236

**external data. See news feeds**

## F

- `\f` (form feed character), JavaScript, 433
- `false` **operator**, XPath, 261

### family tree for DOM, 48

### “fat-client” ideology, 1

`feed` **element**, Atom, 329–330

### feeds

- Atom
  - elements of, 329–330
  - history of, 324
  - extracting text data from, 337–339
  - extracting XML data from, 331–337
  - feed reader, creating, 323, 339–348
  - history of, 324–325, 327
  - RSS (RDF Site Summary)
    - history of, 324, 327
    - RSS 0.9x, 325–327
    - RSS 1.0, 328–329
    - RSS 2.0, 327–328
    - versions of, 324

`FileUpload` **object**, JavaScript, 451

**finally** **clause**, **try** **statement**, 158, 437

### Firebug, 166–168, 172–173

**Firefox. See browsers**

### Firefox DOM inspector, 168–170

`firstChild` **property**, JavaScript, 145, 146

`fixed()` **method**, `String` **object**, 467

`FlashXMLHttpRequest` **object**, 198

### Flickr API

- definition of, 210
- displaying photos from, 222–225
- example using, 212–221, 222–225
- geotagging photos in, 212–213
- key required for, 212
- tag clouds with, 211
- web site for, 198

### flickr web site, 3–4

`floor()` **function**, XPath, 261

`floor()` **method**, `Math` **object**, 459

`focus()` **method**

- `Anchor` **object**, 441
- `Window` **object**, 473

### Focus on JavaScript web site, 426

`focus()` **standard method**, JavaScript, 438

`fontcolor()` **method**, `String` **object**, 467

`fontSize()` **method**, `String` **object**, 467

**for** **loop**, JavaScript, 38, 434

“Forget `addEventListener`, use Yahoo’s Event Utility” (Diaz), 421

`Form` **collection**, ASP.NET, 68

**form feed character (\f), JavaScript, 433**Form **object**, JavaScript, 451–452form **property**

- Button object, 443
- Checkbox object, 444
- FileUpload object, 451
- Hidden object, 454
- Password object, 462
- Radio object, 462
- Reset object, 465
- Select object, 466
- Submit object, 469
- Text object, 469
- Textarea object, 470

<form> **tag, 451**form\_remote\_tag() **function**, Ruby on Rails, 422**forms**

- submission models for, 64–66
- validation of
  - definition of, 276
  - pattern for, 277–284

**forms model of submission, 64–65**forms **property**, Document **object**, 448forward() **method**

- History object, 454
- Window object, 473

**frames**

- hidden frames
  - advantages and disadvantages of, 124
  - definition of, 118
  - example using, 118–124
- hidden inline frames
  - advantages and disadvantages of, 128
  - definition of, 124–125
  - example using, 125–128

frames **property**, Window **object**, 471<frameset> **tag, 471****frameworks**

- Atlas, 423
- definition of, 415
- Ruby on Rails, 422

fromCharCode() **method**, String **object**, 468function **keyword**, JavaScript, 40**function literals, 43**Function **object**, JavaScript, 452–453function **statement**, JavaScript, 434–435Function **type**, JavaScript, 33–34**functions. See also specific functions**

- JavaScript, 40, 439–440
- XPath, 259–261

**G****Garrett, Jesse James (“Ajax: A New Approach to Web Applications”), 2, 9**Geocode **converter**, Google Maps API, 201GET **collection**, PHP, 68GET **method**

- advantages and disadvantages of, 117
- submitting data using, 67
- submitting web service request using, 184

getAllResponseHeaders() **method**, XMLHttpRequest **object**, 103getAncestorsByTag() **method**, Dojo, 419getDate() **method**, Date **object**, 445getDay() **method**, Date **object**, 445getElementById() **method**, Document **object**, 11, 49–50, 449getElementsByClassName() **method**  
Prototype, 417  
YUI, 420getElementsByName() **method**, Document **object**, 449getElementsByTagName() **method**, Document **object**, 11, 50, 147–148, 449getFullYear() **method**, Date **object**, 445getHours() **method**, Date **object**, 445getMilliseconds() **method**, Date **object**, 445getMinutes() **method**, Date **object**, 446getMonth() **method**, Date **object**, 446getNamedItem() **method**, JavaScript, 148getResponseHeader() **method**, XMLHttpRequest **object**, 103getSeconds() **method**, Date **object**, 446getTime() **method**, Date **object**, 446getTimezoneOffset() **method**, Date **object**, 446getUTCDate() **method**, Date **object**, 446getUTCDay() **method**, Date **object**, 446getUTCFullYear() **method**, Date **object**, 446getUTCHours() **method**, Date **object**, 446getUTCMilliseconds() **method**, Date **object**, 446

## getUTCMinutes() method, Date object

---

`getUTCMinutes()` **method**, Date **object**, 446

`getUTCMonth()` **method**, Date **object**, 446

`getUTCSeconds()` **method**, Date **object**, 446

global **property**, RegExp **object**, 464

### Gmail

definition of, 3

XMLHttpRequest object used by, 15

`go()` **method**, History **object**, 454

### Google Maps API

definition of, 198, 199

example using, 202–209

Geocode converter for, 201

key required for, 200

Map object, 200–201

requirements for, 200

XMLHttpRequest factory method for, 201–202

### Google Maps web site, 6–9

### Google Suggest, 6

**greater than operator (>)**, JavaScript, 35, 431

**greater than or equal to operator (>=)**,

JavaScript, 35, 431

## H

### hash **property**

Link object, 456

Location object, 457

head **element**, DOM **family tree**, 48

head **section**, HTML/XHTML **page**, 40

### height **property**

Image object, 455

Screen object, 466

### Heilmann, Christian

article by, 428

blog by, 425

tutorial by, 427

### hidden frames

advantages and disadvantages of, 124

definition of, 118

example using, 118–124

### hidden inline frames

advantages and disadvantages of, 128

definition of, 124–125

example using, 125–128

Hidden **object**, JavaScript, 453–454

### history

of Ajax, 2

of JavaScript, 31

of .NET Framework, 74

of news feeds, 324–325, 327

of XMLHttpRequest object, 100

History **object**, JavaScript, 42, 454

“History of RSS”, online article, 325

history **property**, Window **object**, 471

Holland, Chris (ContextAgnosticXMLHttpRequest **object**), 198

home() **method**, Window **object**, 473

### host objects, 41

#### host **property**

Link object, 456

Location object, 457

#### hostname **property**

Link object, 456

Location object, 457

#### href **property**

Link object, 456

Location object, 457

hspace **property**, Image **object**, 455

### HTML controls, 64

### HTML DOM, 47

html **element**, DOM **family tree**, 48

HTML Goodies, JavaScript **tutorials by**, 427

HTML Source, JavaScript **tutorials by**, 427

### HTTP headers, troubleshooting

with ieHTTPHeaders Explorer Bar, 175–176

with Live HTTP Headers, 173–175

Http **object**, 100

### hyphen (-)

–, JavaScript decrement assignment operator, 430

–, JavaScript decrement operator, 36–37, 430

–, JavaScript subtraction operator, 34, 430

## I

id **element**, Atom, 329, 330

IE Developer Toolbar, 170–171

IE DOM inspector, 170–171

IE (Internet Explorer). *See* browsers

ieHTTPHeaders Explorer Bar, 175–176

IETF (Internet Engineering Task Force), 324

**if statement, JavaScript, 37, 435**  
**ignoreCase property, RegExp object, 464**  
**IIS (Internet Information Server), 73–74**  
**image element, RSS, 326, 328**  
**Image object, JavaScript**  
 definition of, 455–456  
 error event for, 162  
**images property, Document object, 448**  
**images, server requests sent using, 130–135**  
**<img> tag, 455**  
**increment assignment operator (+=), JavaScript, 430**  
**increment operator (++), JavaScript, 36–37, 430**  
**index property, RegExp object, 463**  
**indexOf() method, String object, 468**  
**Infinity constant, JavaScript, 429**  
**initEvent() method, Event object, 451**  
**inline event registration, 55**  
**inline frames, hidden**  
 advantages and disadvantages of, 128  
 definition of, 124–125  
 example using, 125–128  
**innerHTML property, 53–54**  
**input property, RegExp object, 464**  
**INSERT INTO statement, MySQL, 370**  
**inspectors, DOM**  
 Firefox DOM inspector, 168–170  
 IE DOM inspector, 170–171  
 Mouseover DOM Inspector (MODI), 171–172  
**Internet Engineering Task Force (IETF), 324**  
**Internet Explorer (IE). See browsers**  
**Internet Information Server (IIS), 73–74**  
**invisible data retrieval, 18**  
**isFinite() function, JavaScript, 440**  
**isNaN() function, JavaScript, 440**  
**italics() method, String object, 468**  
**item element, RSS, 326–329**

## J

### Java

advantages of, 97  
 definition of, 91  
 example using, 91–96

### Java Virtual Machine (JVM), 91

**javaEnabled() method, Navigator object, 460**

### JavaScript

articles about, 427–428  
 blogs about, 425–426  
 books about, 31  
 case-sensitivity of, 32  
 comments in, 32, 433  
 constants, 429  
 cross-browser code for, 11–12  
 datatypes in, 33–34  
 definition of, 11–12  
 escape characters, 433  
 event handlers, 438–439  
 exception handling, 158–160  
 functions, 40, 439–440  
 history of, 31  
 line returns in, 32  
 methods, 438  
 node properties, 145–147  
 operators, 34–37, 430–432  
 properties, 438  
 semicolon (;) in, 32  
 statements, 37–40, 433–437  
 syntax errors in, 157  
 syntax for, 32  
 tutorials about, 427  
 variables in, 32–33  
 web sites about, 426–427  
**JavaScript console, Mozilla, 162–164**  
**JavaScript Kit web site, 426**  
**JavaScript model of submission, 65–66**  
**JavaScript Object Notation. See JSON**  
**Johns, Morris (tutorial by), 427**  
**join() method, Array object, 442**  
**JScript User's Guide, web site, 426**  
**JSON (JavaScript Object Notation)**  
 array literals, 354  
 data transmission using, 355–357  
 datatypes, 352–353  
 definition of, 351  
 example using, 360–364  
 extracting data from JSON response, 360  
 libraries for, 351  
 object literals, 353–354  
 parser for, 354–355  
 parsing response of JSON data, 359–360  
 requests for JSON data, 357–359

## JSON (JavaScript Object Notation)(continued)

special characters, escaping, 352–353

syntax for, 352

using with PHP, 364–365

“JSON: The Fat-Free Alternative to XML”, online article, 357

JSON-PHP library, 364

JSONRequest object, 198

J2EE SDK, 91

JVM (Java Virtual Machine), 91

## K

Keith, Jeremy (blog by), 425

Koch, Peter-Paul (blog by), 425

## L

label **statement**, JavaScript, 435

language **element**, RSS, 326

lastChild **property**, JavaScript, 145, 146

Last.FM web site, 181

lastIndex **property**, RegExp **object**, 464

lastIndexOf() **method**, String **object**, 468

lastMatch **property**, RegExp **object**, 464

lastModified **property**, Document **object**, 448

lastParen **property**, RegExp **object**, 464

left angle bracket (<)

<<, JavaScript left shift operator, 432

<, JavaScript less than operator, 35, 431

<=, JavaScript less than or equal to operator, 35, 431

<, in XSLT, 235

leftContext **property**, RegExp **object**, 464

length **property**

Array object, 441

Form object, 452

Function object, 453

History object, 454

Select object, 466

String object, 467

Window object, 471

Lerdorf, Rasmus (developer of PHP), 85

less than operator (<), JavaScript, 35, 431

less than or equal to operator (<=), JavaScript, 35, 431

Levitt, Jason (Curl library), 194

libraries

definition of, 415

Dojo library, 418–419

MochiKit library, 421

Prototype library, 416–417

Scriptaculous library, 417–418

Yahoo! User Interface (YUI) library, 419–421

line returns, in JavaScript, 32

link **element**, RSS, 325, 326

link() **method**, String **object**, 468

Link **object**, JavaScript, 456–457

links **property**, Document **object**, 448

link\_to\_remote() **function**, Ruby on Rails, 422

lists

drag-and-drop, 301–314

sortable list example

CSS styles for, 388

database for, creating, 368–371

database for, editing, 374–376

database for, querying, 371–374

description of, 367

droppable elements, creating, 377–378

files used in, 388

requests for list updates, 383–387

sortable elements, creating, 378–380

user interaction with, 380–383

Live HTTP Headers, 173–175

LN2 **constant**, Math **object**, 458

LN10 **constant**, Math **object**, 458

Location **object**, JavaScript, 42, 457–458

location **property**, Window **object**, 471

log() **method**, Math **object**, 459

LOG2E **constant**, Math **object**, 458

LOG10E **constant**, Math **object**, 458

logical functions, XPath, 261

logical operators, JavaScript, 35–36, 431

loops

JavaScript, 38–40

XSLT, 235

## M

Mahemoff, Michael (patterns by), 276

Map **object**, Google Maps API, 200–201

**maps, Google**

- Google Maps API
  - definition of, 198, 199
  - example using, 202–209
  - Geocode converter for, 201
  - key required for, 200
  - Map object, 200–201
  - requirements for, 200
  - XMLHttpRequest factory method for, 201–202
- Google Maps web site, 6–9

**mashups**

- benefits of using with Ajax, 210
- definition of, 179–180, 209
- example using, 222–225

`match()` **method**, *String* **object**, 468

*Math* **object**, *JavaScript*, 458–459

`max()` **function**, *XPath*, 261

`max()` **method**, *Math* **object**, 459

`MAX_VALUE` **constant**, *Number* **object**, 460

**menu system, dynamic**, 21–29

**meta data, in RDF**, 328

**method property**, *Form* **object**, 452

**methods. See also specific methods and objects**

- definition of, 41

- JavaScript*, 438

**Microsoft Script Debugger**, 164–166

`min()` **function**, *XPath*, 261

`min()` **method**, *Math* **object**, 459

`MIN_VALUE` **constant**, *Number* **object**, 460

**minus sign (–)**

- =, *JavaScript* decrement assignment operator, 430

- , *JavaScript* decrement operator, 36–37, 430

- , *JavaScript* subtraction operator, 34, 430

**MochiKit library**, 421

**MODI (Mouseover DOM Inspector)**, 171–172

**modulus assignment operator (%=), *JavaScript***, 430

**modulus operator (%)**, *JavaScript*, 35, 430

**Mouseover DOM Inspector (MODI)**, 171–172

**mouseovers, transparent pop-ups for**, 284–291

`moveBy()` **method**, *Window* **object**, 473

`moveTo()` **method**, *Window* **object**, 473

**Mozilla *JavaScript* console**, 162–164

**Mozilla *JavaScript* Language Resource web site**, 426

**Mozilla *JavaScript* web site**, 426

**Mozilla-based browsers. See browsers**

**multiline property**, *RegExp* **object**, 464

**multiplication assignment operator (\*=), *JavaScript***, 430

**multiplication operator (\*)**, *JavaScript*, 34, 430

**My Netscape portal**, 324

**MySQL**

- database connection, creating, 370–371

- database, creating, 368

- definition of, 368

- queries, creating, 371–372

- table

- adding data to, 370

- creating, 368–369

**MySQL daemon**, 368

**MySQL Monitor**, 368

`mysql_connect()` **function**, *MySQL*, 371

`mysql_select_db()` **function**, *MySQL*, 371

**“The Myth of RSS Compatibility”**, online article, 325

**N**

**\n (new line character)**, *JavaScript*, 433

**name property**

- Anchor* **object**, 441

- Button* **object**, 443

- Checkbox* **object**, 444

- FileUpload* **object**, 451

- Form* **object**, 452

- Hidden* **object**, 454

- Image* **object**, 455

- Password* **object**, 462

- Radio* **object**, 463

- Reset* **object**, 465

- Select* **object**, 466

- Submit* **object**, 469

- Text* **object**, 469

- Textarea* **object**, 470

- Window* **object**, 471

**namespace modules, RSS**, 327

**NaN constant**

- JavaScript*, 429

- Number* **object**, 460

**Navigator object**, *JavaScript*, 42, 459–460

**NCZOnline (Zakas)**, 426

## **.NET Framework**

history of, 74  
required for ASP.NET, 73  
XSLT transform performed using, 245–247

**Netscape. See browsers**

**new** keyword

JavaScript, 41, 44  
JSON, 353, 354

**new line character (\n), JavaScript, 433**

**new operator, JavaScript, 432**

**news feeds**

Atom  
elements of, 329–330  
history of, 324  
extracting text data from, 337–339  
extracting XML data from, 331–337  
feed reader, creating, 323, 339–348  
history of, 324–325, 327  
RSS (RDF Site Summary)  
history of, 324, 327  
RSS 0.9x, 325–327  
RSS 1.0, 328–329  
RSS 2.0, 327–328  
versions of, 324

**nextSibling property, JavaScript, 145, 146**

**node tree for DOM**

adding nodes to, 51–54  
definition of, 48–49

**nodeName property, JavaScript, 145**

**nodes**

adding to DOM, 11  
extracting XML data using, 145–147

**nodeType property, JavaScript, 145–146**

**nodeValue property, JavaScript, 145, 149**

**not equal to operator (!=), JavaScript, 35, 431**

**not equal to or not same type operator (!==),  
JavaScript, 35, 431**

**NOT operator, bitwise (~), JavaScript, 432**

**NOT operator (!), JavaScript, 35–36, 431**

**not operator, XPath, 261**

**null keyword, JavaScript, 45**

**Null type, JavaScript, 33**

**Number() function, JavaScript, 440**

**number() function, XPath, 259–260**

**Number object, JavaScript, 460**

**Number type, JavaScript, 33**

**Nyman, Robert (article by), 427**

## **O**

**Object constructor**

JavaScript, 43  
JSON, 353

**object literals**

JavaScript, 43  
JSON, 353–354

**Object object, JavaScript, 460–461**

**<object> tag, 91**

**Object type, JavaScript, 33–34**

**Object.extend() function, Prototype, 416**

**objects. See also specific objects**

browser objects, 42–43  
built-in objects, 41–42  
creating  
with constructors, 41, 43–44, 47  
example of, 45–47  
with object literals, 43, 353–354  
definition of, 41  
destroying, 45  
host objects, 41  
JSON, 352  
methods of, 41  
properties of, 41  
prototype objects, 44–45  
types of, 41  
user-defined objects, 43–44

**observe\_field() function, Ruby on Rails, 422**

**onabort event handler, JavaScript, 438**

**onblur event handler, JavaScript, 438**

**onchange event handler, JavaScript, 438**

**onclick event handler, JavaScript, 438**

**ondblclick event handler, JavaScript, 438**

**onerror event handler, JavaScript, 160–162, 438**

**onfocus event handler, JavaScript, 438**

**onkeydown event handler, JavaScript, 438**

**onkeypress event handler, JavaScript, 439**

**onkeyup event handler, JavaScript, 439**

**onload event handler, JavaScript, 439**

**onmousedown event handler, JavaScript, 439**

**onmousemove event handler, JavaScript, 439**

**onmouseout event handler, JavaScript, 439**

**onmouseover event handler, JavaScript, 439**

**onmouseup event handler, JavaScript, 439**

**onreadystatechange property, XMLHttpRequest  
Request object, 101, 102**

onreset **event handler, JavaScript, 439**  
 onresize **event handler, JavaScript, 439**  
 onselect **event handler, JavaScript, 439**  
 onsubmit **event handler, JavaScript, 439**  
 onunload **event handler, JavaScript, 439**  
 open() **method**  
   Document object, 449  
   Window object, 473  
   XMLHttpRequest object  
     definition of, 103  
     deploying ASP.NET application, 75  
     deploying PHP application, 86  
     deploying servlet application, 92  
     submitting data, 67  
 opener **property, Window object, 471**  
**Opera. See browsers**  
**operators, JavaScript, 34–37, 430–432**  
 Option **object, JavaScript, 461**  
 <option> **tag, 461**  
 options **property, Select object, 466**  
**OR operator, bitwise (|), JavaScript, 431**  
**OR operator (||), JavaScript, 35–36, 431**  
 or **operator, XPath, 261**  
 outerheight **property, Window object, 472**  
 outerwidth **property, Window object, 472**  
**Outlook Web Access 2000, 100**

## P

**p element, DOM family tree, 48**  
**page (service) streaming, 291–301**  
 pageXOffset **property, Window object, 472**  
 pageYOffset **property, Window object, 472**  
**parameters, in JavaScript functions, 40**  
 Params **collection, ASP.NET, 68**  
 parent **property, Window object, 472**  
**parentheses (()), JavaScript**  
   in conditional statements, 37  
   in event registration (incorrect), 56, 109  
   in function calls, 40  
   in method calls, 41  
 parse() **method, Date object, 446**  
 parseFloat() **function, JavaScript, 440**  
 parseInt() **function, JavaScript, 440**  
 parseJSON() **function, JSON, 354–355, 359–360**  
**partial page update, 17**  
 Password **object, JavaScript, 462**  
 pathname **property**  
   Link object, 456  
   Location object, 457  
**patterns**  
   definition of, 275, 276  
   for drag-and-drop lists, 301–314  
   for form validation, 276–284  
   for handling errors, 314–320  
   for mouseovers, 284–291  
   for polling server, 292–301  
   web site containing, 276  
**percent sign (%)**  
   %=, JavaScript modulus assignment operator, 431  
   %, JavaScript modulus operator, 35, 430  
**performance**  
   of Ajax applications, 19–20  
   of JSON, 365  
   of web applications, 17  
   of web services, 180  
   of XPath, 256  
   of XSLT, 229, 230, 245  
**period (.), dot notation for objects, 41, 432**  
 personalbar **property, Window object, 472**  
**photo-organizing web site. See flickr web site**  
**PHP**  
   advantages of, 97  
   calling applications from JavaScript, 86  
   definition of, 15–16, 85  
   installing, 85  
   using with JSON, 364–365  
**PI constant, Math object, 458**  
 pixelDepth **property, Screen object, 466**  
 platform **property, Navigator object, 459**  
**plus sign (+)**  
   +, JavaScript addition operator, 34, 430  
   +, JavaScript concatenation operator, 34, 432  
   +=, JavaScript increment assignment operator, 430  
   ++, JavaScript increment operator, 36–37, 430  
**polling model, 17**  
**polling server, 291–301**  
 pop() **method, Array object, 442**  
**pop-ups on mouseovers, 284–291**

### port **property**

Link object, 456

Location object, 457

position() **function**, XPath, 260

POSITIVE\_INFINITY **constant**, Number **object**, 460

POST **collection**, PHP, 68

POST **method**

advantages and disadvantages of, 117

aggressive caching controlled using, 115–117

submitting data using, 67

submitting web service request using, 184

pow() **method**, Math **object**, 459

preventDefault() **method**, Event **object**, 451

previousSibling **property**, JavaScript, 145

primitive datatypes, JavaScript, 33

print() **method**, Window **object**, 474

**problems**. See **troubleshooting**

*Professional JavaScript for Web Developers* (Wiley), 31, 160

**project management and tracking with Basecamp**, 4–5

prompt() **method**, Window **object**, 474

**properties**. See also *specific objects and properties*

definition of, 41

JavaScript, 438

protocol **property**

Link object, 456

Location object, 457

**Prototype**, 376–380

**Prototype library**, 416–417, 422

**prototype objects**, 44–45

prototype **property**, Object **object**, 461

prototype **standard property**, JavaScript, 44–45, 438

**proxy, consuming web services using**, 188–195

**public web services**, 181

**publications**

“Ajax: A New Approach to Web Applications” (Garrett), 2, 9

*Beginning JavaScript*, Second Edition (Wiley), 31, 160

dynamic script loading, online article about, 128

“Forget addEvent, use Yahoo’s Event Utility” (Diaz), 421

“History of RSS”, online article, 325

about JavaScript, 427–428

“JSON: The Fat-Free Alternative to XML”, online article, 357

“The Myth of RSS Compatibility”, online article, 325

*Professional JavaScript for Web Developers* (Wiley), 31, 160

“Using PHP 5’s SimpleXML”, online article, 348

*Web Standards Programmer’s Reference: HTML, CSS, JavaScript, Perl, Python, and PHP* (Schafer), 429

push() **method**, Array **object**, 442

## Q

**queries, in MySQL**, 371–372

QueryString **collection**, ASP.NET, 67

**question mark (?)**, JavaScript **conditional operator**, 432

*Quirks Blog* (Koch), 425

**quotation marks, double (“)**

enclosing JSON strings, 352

enclosing tag attributes, 139

escaping in JavaScript, 433

in XSLT, 235

**quotation marks, single (‘)**

enclosing tag attributes, 139

escaping in JavaScript, 433

in XSLT, 235

## R

\r (**carriage return character**), JavaScript, 433

Radio **object**, JavaScript, 462–463

random() **method**, Math **object**, 459

RangeError **type**, JavaScript, 160

**RDBMS (relational database management system)**, 368

rdf **element**, RSS, 329

**RDF (Resource Description Framework)**, 324

**RDF Site Summary (RSS)**

history of, 324, 327

RSS 0.9x, 325–327

RSS 1.0, 328–329

RSS 2.0, 327–328

versions of, 324

- readyState **property**, XMLHttpRequest **object**, **102**
- readystatechange **event**, **101**
- Really Simple Syndication**. *See* **RSS (RDF Site Summary)**
- recursive descent**, XPath, **259**
- reference datatypes**, JavaScript, **33–34**
- ReferenceError **type**, JavaScript, **160**
- referrer **property**, Document **object**, **448**
- RegExp **object**, JavaScript, **463–464**
- registering browser events**, **55–60**
- relational database management system (RDBMS)**, **368**
- releaseEvents() **method**, Window **object**, **474**
- reload() **method**, Location **object**, **458**
- remainder operator (%)**, JavaScript, **35**
- remove() **method**, Select **object**, **466**
- removeEventListener() **method**, Document **object**, **57, 449**
- replace() **function**, XPath, **260**
- replace() **method**  
     Location **object**, **458**  
     String **object**, **468**
- Representational State Transfer (REST)**, **184–185**
- REQUEST **collection**, PHP, **68**
- requests for JSON data**, **357–359**
- requests to server**. *See* XMLHttpRequest **object**
- reset() **method**, Form **object**, **452**
- Reset **object**, JavaScript, **465**
- resizeBy() **method**, Window **object**, **474**
- resizeTo() **method**, Window **object**, **474**
- Resource Description Framework (RDF)**, **324**
- resources**. *See also* **publications; web site resources**
- frameworks
    - Atlas (ASP.NET Ajax), **423**
    - Ruby on Rails, **422**
  - libraries
    - Dojo library, **418–419**
    - MochiKit library, **421**
    - Prototype library, **416–417**
    - Scriptaculous library, **417–418**
    - Yahoo! User Interface (YUI) library, **419–421**
- Response **object (HTTP)**, Write() **method**, **68–69**
- response time**. *See* **performance**
- responseBody **property**, XMLHttpRequest **object**, **103**
- responseText **property**, XMLHttpRequest **object**  
 definition of, **70–71, 103**  
 retrieving data, **323, 337–339, 347**  
 retrieving JSON data, **351**
- responseXML **property**, XMLHttpRequest **object**  
 debugging, **71–73**  
 definition of, **71, 103**  
 retrieving data, **323, 336**
- REST (Representational State Transfer)**, **184–185**
- return characters**, in JavaScript, **32**
- return statement**, JavaScript, **40, 434–435**
- reverse() **method**, Array **object**, **442**
- Rich Site Summary**. *See* **RSS (RDF Site Summary)**
- right angle bracket (>)**  
 >, JavaScript greater than operator, **35, 431**  
 >=, JavaScript greater than or equal to operator, **35, 431**  
 >>, JavaScript right shift operator, **432**  
 >>>, JavaScript zero fill right shift operator, **432**  
 >, in XSLT, **235**
- rightContext **property**, RegExp **object**, **464**
- root element**, XPath, **258**
- round() **function**, XPath, **261**
- round() **method**, Math **object**, **459**
- routeEvent() **method**, Window **object**, **474**
- RSS**. *See also* **news feeds**
- rss **element**, RSS, **325**
- RSS (RDF Site Summary)**  
 history of, **324, 327**  
 RSS 0.9x, **325–327**  
 RSS 1.0, **328–329**  
 RSS 2.0, **327–328**  
 versions of, **324**
- Ruby on Rails**, **422**
- runtime errors**. *See* **errors and exceptions**

## S

- same origin issues**, XMLHttpRequest **object**, **110, 187–188**
- Schafer, Steven M. (Web Standards Programmer's Reference: HTML, CSS, JavaScript, Perl, Python, and PHP)**, **429**

Screen **object, JavaScript, 42, 465–466**

screenX/screenY **property, Event object, 450**

**Script Debugger, Microsoft, 164–166**

**script tag hack, 195–197**

**Scriptaculous JavaScript library, 376–380, 417–418**

**Scripting News, 324**

scrollbars **property, Window object, 472**

scrollTo() **method, Window object, 474**

**search engine**

A9.com (Amazon search engine), 5

Google Suggest, 6

indexes of, blocked by Ajax, 19

search() **method, String object, 468**

**search property**

Link object, 456

Location object, 457

**security, for XMLHttpRequest object, 110, 187–188**

select() **method**

Text object, 470

Textarea object, 471

Select **object, JavaScript, 466–467**

select() **standard method, JavaScript, 438**

SELECT **statement, MySQL, 370, 371**

selected **property, Option object, 461**

selectedIndex **property, Select object, 466**

self **property, Window object, 472**

**semicolon (;), in JavaScript, 32**

send() **method, XMLHttpRequest object, 103**

**server**

polling, 291–301

requests received by, handling of, 67–68

response from

retrieving, 69–73

writing, 68–69

submitting data to, 67

**server-side technology**

ASP.NET

advantages of, 97

calling applications from JavaScript, 75

definition of, 15–16, 73–74

deploying applications, 74–75

example using, 75–85

choosing, 96–97

compared to client-side technology, 1–2

for database processing, 368

forms used in, 64

frameworks

Atlas (ASPRNET Ajax), 423

Ruby on Rails, 422

HTML controls used in, 64

Java

advantages of, 97

definition of, 91

example using, 91–96

options for, 63–64, 73

PHP

advantages of, 97

calling applications from JavaScript, 86

definition of, 15–16, 85

installing, 85

using with JSON, 364–365

submission models for, 64–66

**service (page) streaming, 291–301**

**servlets**

definition of, 91

example using, 91–96

requirements for, 91–92

setDate() **method, Date object, 446**

setForm() **method, YUI, 420**

setFullYear() **method, Date object, 446**

setHours() **method, Date object, 446**

setInterval() **method, Window object, 474**

setMilliseconds() **method, Date object, 446**

setMinutes() **method, Date object, 446**

setMonth() **method, Date object, 446**

setRequestHeader() **method, XMLHttpRequest**

**Request object, 103**

setSeconds() **method, Date object, 446**

setTime() **method, Date object, 447**

setTimeout() **method, Window object, 474**

setUTCDate() **method, Date object, 447**

setUTCFullYear() **method, Date object, 447**

setUTCHours() **method, Date object, 447**

setUTCMilliseconds() **method, Date object, 447**

setUTCMinutes() **method, Date object, 447**

setUTCMonth() **method, Date object, 447**

setUTCSeconds() **method, Date object, 447**

shift() **method, Array object, 442**

**shopping cart example**

- with ASP.NET, 75–85
- interactive, 267–273
- with Java servlets, 91–96
- with PHP, 86–91
- with XSLT, 247–256

**Simon Willison's Weblog (Willison), 426****Simple Object Access Protocol (SOAP), 184, 185–186****simpleXML, 348****sin() method, Math object, 459****single quote (')**

- enclosing tag attributes, 139
- escaping in JavaScript, 433
- in XSLT, 235

**slash (/)**

- //, indicating XPath recursive descent, 259
- //, in JavaScript, 32, 433
- /\* ... \*/ , in JavaScript, 32, 433
- /=, JavaScript division assignment operator, 430
- /, JavaScript division operator, 34, 430
- /, in JSON, 352
- /, preceding XPath document root, 258
- /\*, preceding XPath root element, 258

**slice() method**

- Array object, 442
- String object, 468

**slider script, Scriptaculous, 418****small() method, String object, 468****SOAP (Simple Object Access Protocol), 184, 185–186****solutions to exercises, 391–413****sort() method, Array object, 442****sortable list example**

- CSS styles for, 388
- database for
  - creating, 368–371
  - editing, 374–376
  - querying, 371–374
- description of, 367
- droppable elements, creating, 377–378
- files used in, 388
- requests for list updates, 383–387
- sortable elements, creating, 378–380
- user interaction with, 380–383

**sorting, in XSLT, 236****source property, RegExp object, 464****special characters**

- escaping in JavaScript, 433
- escaping in JSON, 352–353
- escaping in XSLT, 235

**speed. See performance****splice() method, Array object, 442****split() method, String object, 468****sqrt() method, Math object, 459****SQRT1\_2 constant, Math object, 458****SQRT2 constant, Math object, 458****src property, Image object, 455****statements, JavaScript, 37–40****status property**

- Window object, 472

XMLHttpRequest object, 103

**statusbar property, Window object, 472****statusText property, XMLHttpRequest object, 103****stock ticker, polling, 291–301****stop() method, Window object, 474****stopPropagation() method, Event object, 451****strike() method, String object, 468****string formatting, XPath, 260****String() function, JavaScript, 440****String object, JavaScript, 467–468****string operators, JavaScript, 432****String type, JavaScript, 33, 34****style property, JavaScript, 153–154****style sheets. See CSS (Cascading Style Sheets); XSLT (Extensible Stylesheet Language Transformation)****sub() method, String object, 468****submission models, 64–66****submit() method, Form object, 452****Submit object, JavaScript, 469****substr() method, String object, 468****substring() function, XPath, 260****substring() method, String object, 468****subtraction operator (-), JavaScript, 34, 430****sum() function, XPath, 261****summary element, Atom, 330****sup() method, String object, 468****switch statement, JavaScript, 38, 435****synchronous model, 16–17****SyntaxError type, JavaScript, 160**

## T

**\t (tab character), JavaScript, 433**

**table, MySQL**

- adding data to, 370
- creating, 368–369

**tag clouds, Flickr API, 211**

**tags, XML**

- attributes of, syntax for, 139
- creating, 138
- nesting, 139
- opening and closing, requirements for, 139

**tan() method, Math object, 459**

**target property**

- Form object, 452
- Link object, 456

**templates (style sheets). See CSS (Cascading Style Sheets); XSLT (Extensible Stylesheet Language Transformation)**

**test() method, RegExp object, 464**

**text nodes, DOM node tree, 49**

**Text object, JavaScript, 469–470**

**text property, Option object, 461**

**Textarea object, JavaScript, 470–471**

**<textarea> tag, 470**

**“thin-client” ideology, 1**

**this keyword, JavaScript, 44**

**throw statement, JavaScript, 436**

**tilde (~), JavaScript bitwise NOT operator, 432**

**title element**

- Atom, 329, 330
- DOM family tree, 48
- RSS, 325, 326

**title property, Document object, 448**

**toDateString() method, Date object, 447**

**toFixed() method, Number object, 460**

**toJSONString() function, JSON, 354–355**

**toLocaleString() method, Date object, 447**

**toLowerCase() method, String object, 468**

**Tomcat@Jakarta, 92**

**toolbar property, Window object, 472**

**top property, Window object, 472**

**toString() method**

- Array object, 442
- Boolean object, 443

Date object, 447

Function object, 453

Number object, 460

Object object, 461

RegExp object, 464

String object, 468

**toUpperCase() method, String object, 468**

**toUTCString() method, Date object, 447**

**traditional event registration, 55**

**transparent pop-ups on mouseovers, 284–291**

**troubleshooting. See also errors and exceptions**

aggressive caching, controlling, 110–117

Ajax applications

- with Firebug console, 172–173
- with ieHTTPHeaders Explorer Bar, 175–176
- with Live HTTP Headers, 173–175

DOM inspectors

- Firefox DOM inspector, 168–170
- IE DOM inspector, 170–171
- Mouseover DOM Inspector (MODI), 171–172

event registration, 109

JavaScript syntax errors, 157

responseXML property, XMLHttpRequest object, 72

XMLHttpRequest object, 109, 157, 172–173

**true operator, XPath, 261**

**truth table, for JavaScript logical operators, 36**

**try statement, JavaScript, 158–160, 436**

**tutorials for JavaScript, 427**

**type property**

- Button object, 443
- Checkbox object, 444
- Event object, 450
- FileUpload object, 451
- Hidden object, 454
- Password object, 462
- Radio object, 463
- Reset object, 465
- Select object, 466
- Submit object, 469
- Text object, 470
- Textarea object, 470

**TypeError type, JavaScript, 159, 160**

**typeof operator, JavaScript, 432**

## U

- ul element, DOM family tree, 48**
- Undefined **type, JavaScript, 33**
- underscore (\_), in JavaScript variables, 33**
- unregistering browser events, 56, 57**
- unshift() **method, Array object, 442**
- updated **element, Atom, 329, 330**
- URIError **type, JavaScript, 160**
- URL, location object containing, 42, 457**
- URL **property, Document object, 448**
- userAgent **property, Navigator object, 459**
- user-defined objects, 41, 43–44**
- users, ability to use Ajax, 20**
- “Using PHP 5’s SimpleXML”, online article, 348**
- UTC() **method, Date object, 447**

## V

- valid XML, 139–140**
- value **property**
  - Button object, 443
  - Checkbox object, 444
  - FileUpload object, 451
  - Hidden object, 454
  - Option object, 461
  - Password object, 462
  - Radio object, 463
  - Reset object, 465
  - Submit object, 469
  - Text object, 470
  - Textarea object, 470
- valueOf() **method**
  - Array object, 442
  - Boolean object, 443
  - Date object, 447
  - Function object, 453
  - Number object, 460
  - Object object, 461
  - String object, 468
- Vandersluis, Bobby (article by), 428**
- var **keyword, JavaScript, 32–33, 437**
- variables**
  - JavaScript, 32–33
  - in XSLT, 236

- version of browser**
  - ability to use Ajax based on, 20
  - in Navigator object, 42, 459
- version of RSS (RDF Site Summary), 324**
- vertical bar (|)**
  - |, JavaScript bitwise OR operator, 431
  - ||, JavaScript OR operator, 35–36, 431
- Virtual Earth API, 199**
- visual effects**
  - definition of, 415
  - Scriptaculous JavaScript library for, 376
- vspace **property, Image object, 455**

## W

- Wait Till I Come! (Heilmann), 425**
- web services**
  - compared to APIs, 181, 199
  - consuming
    - example of, 181–183
    - future solutions for, 198
    - process for, 183–184
    - with a proxy, 188–195
    - with a proxy server, 194–195
    - with script tag hack, 195–197
    - with XMLHttpRequest object, 187–188
  - cross-domain browsing of, 198
  - definition of, 179, 180–181
  - disadvantages of, 180
  - endpoint for, 181
  - public, 181
  - REST approach to submitting requests, 184–185
  - SOAP approach to submitting requests, 185–186
  - standards for, 180
- web site resources**
  - Amazon Web Services, 181
  - A9.com (Amazon search engine), 5
  - APIs, 198–199
  - ASPNET, 74
  - Atlas (ASPNET Ajax), 423
  - Basecamp, 4
  - ContextAgnosticXMLHttpRequest object, 198
  - Curl library, 194
  - del.icio.us API, 199
  - Dojo library, 418, 419

### web site resources (continued)

- dynamic script loading, online article about, 128
- eBay, 181
- eBay API, 199
- ECMA-262 standard, 42
- ECMA-327 standard, 429
- Firebug, 166, 168
- FlashXMLHttpRequest object, 198
- flickr, 3
- Flickr API, 198, 210
- Google Maps API, 198, 199
- Google Suggest, 6
- IE Developer Toolbar, 171
- ieHTTPHeaders Explorer Bar, 175
- Java, 91
- JavaScript, 425–428
- JSON (JavaScript Object Notation), 351, 364
- JSON parser, 354
- JSON-PHP library, 364
- JSONRequest object, 198
- J2EE SDK, 91
- Last.FM, 181
- list of Ajax mistakes, 8
- list of reasons not to use Ajax, 19
- list of top 10 Ajax applications, 7
- Live HTTP Headers, 173
- Microsoft Script Debugger, 164
- MochiKit library, 421
- MODI (Mouseover DOM Inspector), 171
- MySQL, 368
- namespace modules, RSS, 327
- Opera, 100
- patterns, 276
- PHP, 85
- Prototype library, 416, 417
- RSS 2.0 specification, 328
- RSS, history of, 325
- Ruby on Rails, 422
- Scriptaculous JavaScript library, 376
- Scriptaculous library, 417
- simpleXML, 348
- tasks to use Ajax for, 7–8
- Tomcat@Jakarta, 92
- A Venture Forth's web site, 7
- Virtual Earth API, 199
- W3C (World Wide Web Consortium), 10
- web services, public, 181
- web services standards, 180
- WebServiceX.Net, 181
- XML specification, 139
- XMLHttpRequest object, standards for, 100
- XPath specification, 256
- Yahoo! User Interface (YUI) library, 419, 421
- Yahoo Web Services, 181
- YouTube API, 198
- Web Standards Programmer's Reference: HTML, CSS, JavaScript, Perl, Python, and PHP (Schafer), 429**
- Web Standards with Imagination (Diaz), 425**
- Webdeveloper.com web site, 426**
- Webreference.com web site, 426**
- WebServiceX.Net web site, 181**
- well-formed XML, 139–140**
- West, Mike (article by), 427**
- while loop, JavaScript, 38–39, 434, 437**
- widgets, 415**
- width property**
  - Image object, 455
  - Screen object, 466
- Williams, Mike (tutorial by), 427**
- Willison, Simon (blog by), 426**
- Window object, JavaScript**
  - definition of, 42, 471–474
  - error event for, 160–162
- window property, Window object, 472**
- window.event property, 57**
- window.location property, 457**
- Winer, Dave (RSS format modified by), 324**
- with statement, JavaScript, 437**
- World Wide Web Consortium (W3C)**
  - DOM specified by, 10
  - RDF recommendation by, 324
  - web services standards, 180
- write() method, Document object, 449**
- Write() method, Response object, 68–69**
- writeln() method, Document object, 449**
- W3Schools, JavaScript tutorials by, 427**

**X****XHTML (eXtensible HyperText Markup Language), 9–10****XML**

creating tags, 138  
 CSS applied to, 152–154  
 definition of, 12, 137  
 example using, 21–29  
 extracting data from  
   attribute nodes, 148–152  
   by element name, 147–148  
   with nodes, 145–147  
 querying, using XPath, 262–266  
 requesting from server, example of, 140–144  
 root element for, 138–139  
 syntax for documents, 138–139  
 valid, 139–140  
 W3C specification for, 139  
 well-formed, 139–140  
 <?xml?> **declaration, 138**

**XML DOM, 47****XML news feeds. See news feeds****XML Schema, 139**XMLHttpRequest **object, 14**XMLHttpRequest **factory method, Google Maps API, 201–202**XMLHttpRequest **object**

aggressive caching, controlling, 110–114  
 asynchronous usage of, 101–102  
 consuming web service using, 187–188  
 creating, 100–101  
 definition of, 14–15, 100  
 example using, 21–29, 103–109, 383–387  
 feed reader created using, 323, 331–337, 339–348  
 history of, 100  
 methods of, 103  
 open() method, 67  
 properties of, 102–103  
 same origin issues with, 110, 187–188  
 security issues with, 110, 187–188  
 synchronous usage of, 101  
 troubleshooting, 157, 172–173

**XOR operator, bitwise (^), JavaScript, 432****XPath. See also XSLT (Extensible Stylesheet Language Transformation)**

current context in, 258  
 definition of, 14, 227, 256–257  
 document root in, 258  
 expressions in, 257–259  
 functions in, 259–261  
 performance of, poor, 230  
 querying XML document using, 262–266  
 recursive descent, 259  
 root element in, 258  
 specification for, 256

**XSL style sheets. See XSLT (Extensible Stylesheet Language Transformation)**

xsl:apply-templates **element, 232–234**  
 xsl:call-template **element, 232–234**  
 xsl:choose **element, 234–235**  
 xsl:for-each **element, 235**  
 xsl:if **element, 234**  
 xsl:includes **element, 231**  
 xsl:otherwise **element, 234**  
 xsl:output **element, 230–231**  
 xsl:param **element, 233–234**  
 xsl:sort **element, 236**  
 xsl:stylesheet **element, 230**

**XSLT (Extensible Stylesheet Language Transformation). See also XPath**

browsers supporting, 237  
 decision-making structure for, 234–235  
 definition of, 12–14, 227, 228–229  
 elements of, 230–237  
 including files in stylesheet, 231  
 looping structure for, 235  
 output format for transformation, 230–231  
 reasons not to use, 229–230  
 reasons to use, 229  
 shopping cart example using, 247–256  
 sorting, 236  
 special characters in, escaping, 235  
 style sheets (templates) for  
   creating, 232–233  
   example using, 21–29  
   invoking by match, 232  
   invoking by name, 233  
   parameters for, 233–234

### **XSLT (Extensible Stylesheet Language Transformation)(continued)**

transform, performing  
  with Firefox, 242–244  
  with IE (Internet Explorer), 237–242  
  on server, with .NET Framework, 245–247  
variables, 236  
`xsl:template` **element, 232–234**  
`xsl:transform` **element, 230**  
`xsl:value-of` **element, 236**  
`xsl:variable` **element, 236–237**  
`xsl:when` **element, 234**

## **Y**

**Yahoo! User Interface (YUI) library, 419–421**  
**Yahoo Web Services web site, 181**  
**YouTube API, 198**

## **Z**

**Zakas, Nicholas**  
  blog by, 426  
  dynamic script loading technique by, 128  
**Zend Technologies, 85**  
**zero fill right shift operator (>>>), JavaScript, 432**



