

Index

Symbols

-- (decrement) operator, 75, 80
 & (AND) Bitwise logical operator, 72–73, 80
 \$ (dollar sign) in replacement symbols, 185
 % (modulus) operator, 74–75, 80
 () (parentheses) in expressions, 85
 , (comma)
 object literals and, 46
 operator, 75
 “ ” (quotation marks)
 in JSON, 557
 string literals and, 43–44
 ‘ ’ (single quotes) in string literals, 43–44
 . . . (ellipses) for shortening strings, 195
 /* and */ in comments, 48
 // (forward slashes)
 in comments, 48
 in regular expression literals, 46
 ?: (conditional) operator, 75–76
 ? (match zero or one times symbol), 175
 @ (at) symbol, 48
 \ (backslash)
 in instances of regular expressions,
 173–174
 in JSON, 558
 ^ (caret) symbol in string groups, 177
 ^ (XOR) Bitwise logical operator, 73
 _ (underscores) for private properties/
 methods, 262
 { } (curly braces)
 blocks and, 47
 object literals and, 46
 | (OR) Bitwise logical operator, 72
 + (concatenate operator), 159–160
 ++ increment operator, 75, 80
 + (plus) operator, 21
 !== and != equality operators, 65
 === and == equality operators, 65, 66
 !== and != equality operators, 66
 = (equals) operator, 66

| (pipe) as alternation symbol, 179
 32-bit PNG images in IE7, 459–460

A

absolute positioning (CSS), 443–444, 447–451
 accelerator plugins (browsers), 574
 accidental closures, 125
 Acid Tests, 406
 AC_Quicktime.js JavaScript library, 607
 Action Message Format (AMF), 523
 ActionScript
 3.0, 596
 in Flash, 7
 ActiveX, security and, 484–485
 ActiveXObject (global), 725
 addBehavior() method, 492
 Adobe Flash
 cross-domain Ajax and, 523–524
 Flash storage, 502–505
 security and, 485
 AIR (Adobe Integrated Runtime), 7
 Ajax, 511–531
 “Ajax: A New Approach to Web Applications”,
 511
 browser compatibility and, 36
 cross-domain Ajax. See cross-domain Ajax
 history and bookmarking, 525–530
 JSON and, 567–569
 XMLHttpRequest. See XMLHttpRequest (XHR)
 alarmArray, 247–248
 alert() dialogue boxes, 289–290
 alpha channel feature, 459
 alphabetical string comparison, 163–164
 alternation symbol (|), 179
 AMF (Action Message Format), 523
 anchor/link objects, 951–952
 .anchor(anchorName) method, 167
 animation
 with canvases (browsers), 576–577
 DHTML, 451

anonymous functions

- as closures, 47
- defined, 105

Applets, Java, 591–594

apply() method, 126–127, 131

Aptana, 16

Area object, 911–912

arguments

- apply() method, 127
- argument hashes, 115–118
- call() method, 127
- creating SQLite databases, 499
- defining (functions), 104
- for loops and, 87
- object, 114–115
- optional, overloading and, 112–114
- passing by value vs. reference, 106
- passing to functions (replace method), 186–187

arithmetic expressions, 61

arithmetic operators (JavaScript), 74, 654–660

Array object (global), 725–746

arrays

- for .. in loops and, 89
- adding elements to. See elements, adding to arrays
- arguments object and, 114
- array literals, 45
- Array object, basics, 217–218
- array of arrays, 220
- Array.join() method, 233–234
- associative, 114, 229–231
- creating, 218–219
- detecting, 220–222
- indexed, 219
- iterating over, 223–224
- loops and, 643
- multi-dimensional, 220
- objects as, 255–256
- as reference types, 231–233
- removing elements from, 227–229
- size of, 223
- strings and, 196–197, 233–234

ASCII, 40

assignment operators (JavaScript)

- basics, 66–68
- defined, 62
- reference, 660–669

associative arrays

- basics, 229–231

- defined, 229
- iterating and, 224
- passing argument and, 114

associativity, operators and, 80–81

asynchronous connections (XMLHttpRequest), 515–516

Asynchronous JavaScript and XML. See Ajax

attack surface (Ajax applications), 518

attributes

- cookie attributes, 489–490
- CSS, 407–416
- defined (DOM), 356
- Number.POSITIVE_INFINITY/Number.NEGATIVE_INFINITY, 209
- XML/HTML documents, 540

authentication

- Authenticode, 480, 484
- Cross-site request forgery and, 478

autosizing textareas, 395

B

Back buttons, Ajax and, 525–526

background-position attribute (DOM), 440

backward compatibility, comparison operators and, 66

basic event model, 306–308

behavior feature (IE 5), 492

big-endian order, bitwise operators and, 70–71

bindings, event, 306–307, 326

bitwise operators (JavaScript), 62, 70–73, 669–674

block operator ({ }), 84

blocks

- basics, 47
- script, 19

blur() method, 385, 386

body object, 920–922

BOM (Browser Object Model), 9–10, 271

bookmarking, Ajax and, 525–530

Booleans

- Boolean literals, 43
- Boolean object, 199–203, 746
- boolean values, 199
- comparison operators and, 64
- converting types to, 56
- logical operators and, 68–69
- working with, 200–201

BoxObject, 449

break statements, 93–95

breakpoints, setting (Firebug), 623**browsers**

- array iterator performance, testing, 223–224
- browser-based development (JavaScript), 4–5
- comparing features of, 164
- concurrent requests and, 633
- detecting, 291–293
- detection class, building, 295–300
- DOM storage events and, 496, 497
- event compatibility comparison, 336–338
- global object in, 130–131
- gZip and, 635
- navigator object, 292
- numbers and, 204
- performance. *See* performance tuning
- replacement scheme symbols, 185
- screen object, 294–295
- security. *See* security
- serializing XML to text in, 539
- status bar, 276
- storage limits, 506
- string performance in, 160–161
- support for E4X, 553
- support of statements, 82–83
- supporting ExternalInterface (Flash), 503
- supporting global object functions, 132–133
- supporting XPath, 362
- testing with, 18–19
- W3C DOM storage and, 495
- WYSIWYGs and, 396–397
- XMLHttpRequest and. *See* XMLHttpRequest (XHR)
- XPath queries and, 544–545

browsers, JavaScript in

- cross-browser compatibility, 36
- deferred scripts, 30–31
- DOM, 23–24
- dynamically loaded scripts, 33–34
- event-driven scripts, 31–32
- execution and load order, 27
- external scripts, 30
- inline scripts, 28–29
- <noscript> element, 27
- <script> element, 25–26
- script masking, 27
- URLs, JavaScript in, 35

browser features

- accelerator plugins, 574
- animating with canvases, 576–577
- canvas elements, 575–577

- conditional compilation feature, 577–578
- CSS Transforms, 578–580
- Gears plugin (Google), 582–586
- Geolocation API, 580–582
- overview of unique, 573
- search providers, 586
- Vector Markup Language (VML), 587–588
- Web Workers threading module, 588–589
- WorkerPool threading module (Gears), 584–586

bubbled statements, 96**bubbling, event, 319–320, 343****buttons**

- Back and Forward (Ajax), 525
- basics, 386
- BUTTON/SUBMIT/RESET objects, 939
- custom submit, 381
- radio, 387–389

C**caching**

- cache variable, 507–508
- hashes for, 231
- page weight reduction and, 634

calls

- call() method, 127, 131
- call stacks, 96
- repeated math calls, 214–215

canvas elements (browsers), 575–577**Canvas object, 912–914****capitalization, strings and, 168****caption object, 946****capture mode for IE mouse events, 322–323****capture phase (event propagation), 320–322****carriage returns, string literals and, 44****Cascading Style Sheets (CSS). *See* CSS (Cascading Style Sheets)****case sensitivity**

- JavaScript and, 41
- strings and, 167–168

catch() statement, 616**CDNs (Content Delivery Networks), 363, 635–636****Char classes, defined, 156****characters**

- encoding, 40–41
- extracting from strings, 192–193

charAt() method, 192–193**Charles Proxy Debugger, 627**

checkboxes

basics, 387

CHECKBOX/RADIO objects, 940

child frames, 274

`childNodes[]` collection, 354

circular references, 123–125

classes

Char, 156

class definitions in Java, 252

fundamentals of, 253

JavaScript programming and, 40

`clearInterval()` method, 246

client-side data and persistence

basics, 487

cookies, defined, 487

Flash storage, 488, 502–505

HTML5 client-side database, 488, 498–502

methods of persisting data, 488

persistence, defined, 487

SQLite databases, 498–502

UserData in IE, 488, 492–494

W3C DOM storage, 488, 495–498

window.name storage, 488, 505–508

cloning

defined, 251

objects, 251, 257

closures

accidental, 125

basics, 47, 118–119

circular references, 123–125

evaluated code, cleaning up, 119–120

function factories, 120

in JavaScript programming, 40

with loops, 121–123

private methods, simulating, 120–121

privileged members and, 262–263

problems with, 4

timers and, 246–248

code. See also source code, downloading

cleaning up evaluated, 119–120

`eval()` global function, 134–136

JSON as evaluated code, 558

and performance, 632

profiling, 636–638

writing using *with* statements, 100

code optimization

avoiding evaluated code, 639–640

deleting unused objects, 638–639

local vs. global variable lookup, 640

object and function pointers, 640–641

repeated for `..` in loops, 642–643

with statements, avoiding, 641

`try...catch...finally` constructs, 642

tuning loops, 643

col/colgroup objects, 946

collapsing ranges, 371

collections (DOM elements), 359–360

color of elements, modifying (DHTML), 457–458

combinational (connubial) operators, 62, 73–75

comma (,)

object literals and, 46

operator, 75

commands, text editor (iFrame), 398–399

comments

basics, 48

JavaScript (reference), 716–721

comparison

of objects, 256–257

operators, 62–66, 674–680

of strings, 163–164, 189–190

XOR comparison of Boolean values, 202

compatibility

backwards (comparison operators), 66

of events, 336–338

mode. See quirks mode

compiled languages, 5

composite data types vs. primitive data types, 51

computed styles (CSS), 428–429

`+=` (concatenate assignment operators), 159, 160

concatenate operator (+), 160

concatenation

`concat()` function, 226–227

`concat()` method, 159–160, 232

to reduce page weight, 634–635

of strings, 159–161

conditional compilation, 577–578, 716–721

conditional operator (?:), 75

conditional statements, 84–87

`confirm()` dialogue boxes, 289–290

connections, opening (XMLHttpRequest), 514–517

connubial operators, 73–75

console log, Firebug, 622

constants

mathematical, 211–212

`resultType`, 545

constructor property

detecting arrays with, 221–222

- prototype chaining and, 266
 - referring to constructor function via, 259
 - typeof operator and, 144
 - constructors**
 - creating arrays and, 218–219
 - defined, 221, 253
 - Function object constructors, 105
 - functions as, 258–259
 - prototype chaining and, 266
 - contains method, 170, 181–182**
 - content**
 - Content Delivery Network (CDN), 363, 635–636
 - of documents, modifying, 363
 - of DOM, modifying, 343
 - of ranges, changing, 370–371
 - of web pages (DOM), 341
 - of WYSIWYG, 401
 - context**
 - execution context basics, 126–127
 - execution context, eval() global function, 135
 - variable, defined, 107
 - continue statements, loops and, 93–95**
 - cookies**
 - basics, 489–492
 - defined, 487
 - Coordinated Universal Time (UTC), 238**
 - coordinates**
 - geolocation, 581–582
 - mouse, 327
 - Copy on Write (COW) technique, 155, 154**
 - copying**
 - indexed arrays, 232–233
 - primitive data types, 52
 - Core DOM, 344**
 - Crockford, Douglas, 262, 268, 555, 635**
 - cross-browser compatibility, 36**
 - cross-browser event utility, 318–319**
 - cross-domain Ajax**
 - basics, 519
 - document.domain, 521
 - Flash and, 523–524
 - iFrames, 522
 - image injection, 522–523
 - method comparison, 520
 - <SCRIPT> injection, 523
 - server proxy, 521–522
 - XMLHttpRequest and, 524–525
 - Cross-site request forgery, 478, 559**
 - Cross-Site Scripting, 477**
 - CSS (Cascading Style Sheets)**
 - adding/removing style sheets, 422–423
 - basics, 8–9
 - browser compatibility and, 36
 - computed styles, 428–429
 - CSS Transforms (browsers), 578–580
 - cssRule (rule) object, 914
 - DOM and, 407–416
 - dynamic HTML, 432–433
 - embedding in documents, 403–405
 - filter object (IE), 429–430
 - imported style sheets, 418–419
 - iterating over all stylesheets, 419–422
 - order of style cascades, 407
 - overview, 403
 - rules, adding/removing, 426–427
 - rules, iterating over all, 423
 - rules, searching for, 424
 - style properties, reading/writing, 424–425
 - styleSheet/style objects, 416–418
 - styling information for browsers and, 167
 - versions of, 405–406
 - custom errors**
 - throwing, 615–616
 - types, 614
 - custom events, 338–340**
- ## D
- data**
 - changing with GET requests, 519
 - client-side data and persistence. See client-side data and persistence
 - data security, 484
 - loading JSON, 564–565
 - storage of. See client-side data and persistence
 - transforming with XSLT, 548–552
 - data types**
 - converting to Booleans, 68–69
 - manipulating by value vs. reference, 52–53
 - non-numeric, converting to numbers, 207
 - non-string, converting to strings, 161–162
 - null and undefined, 53–54
 - primitive Boolean value, 199
 - primitive vs. reference types, 51–52
 - primitives vs. primitive objects, 58
 - type conversion, 54–58
 - type, determining, 54
 - databases, Gears plugin, 583**

dates

- comparison operators and, 64–65
 - creating, 236–237
 - Date object, 234–236, 754–781
 - date strings, parsing, 238–239
 - reading and writing, 240–244
 - time differences, measuring, 244–245
 - timers and intervals, 245–248
 - world time overview, 237–238
- daysInMonth() function, 243–244**
- debugging**
- Debug object (global), 781–782
 - Charles Proxy Debugger, 627
 - Developer Toolbar (IE), 624–625
 - Dragonfly (Opera), 626
 - error handlers, 616
 - Error object overview, 614–615
 - Fiddler, 626–627
 - Firebug (Firefox), 620–623
 - Firebug Lite (Firefox), 623–624
 - stack trace function, 616–619
 - testing tools, 628
 - throwing custom errors, 615–616
 - tools overview, 619
 - types of errors, 613–614
 - Web Inspector (Safari 4), 627–628
- declaring functions, 104–105**
- decrement (- -) operator, 75, 80**
- default behavior, preventing (events), 310**
- default handlers for events, 323**
- deferred scripts, 30–31**
- delete operators, 76–77**
- deleting**
- cookies, 492
 - nodes, 367
 - properties and objects, 253–254
 - unused objects, 638–639
- Denial of Service attacks, 483–484**
- deserializing text, XML and, 533–536**
- desktop widgets, JavaScript and, 7**
- detecting**
- arrays, 220–222
 - browsers, 291–293
- Developer Toolbar (IE), 624–625**
- DHTML**
- optimization of, 644–647
 - performance problems caused by, 632
- DHTML (dynamic HTML)**
- 32-bit PNG images in IE7, 459–460
 - and absolute positioning (CSS), 443–444

- animation and, 451
 - color, 457–458
 - CSS and, 432–433
 - documents, IE4 and, 343
 - element dimensions, 437–438
 - elements, getting absolute position of, 447–451
 - form tooltips, 467–472
 - geometry, window and document, 433–435
 - image swapping and rollovers, 438–440
 - modal dialogues/windows (example), 460–467
 - non-linear animation and tweening, 453–457
 - opacity, 458–459
 - overview, 431–432
 - and relative positioning (CSS), 444–445
 - rollovers and mouseenter/mouseleave events, 441–443
 - scrollbar width, 435–437
 - timers, pseudo-threading with, 452–453
 - yellow-fade technique, 457–458
 - z-index, scripting, 446–447
- dialogue boxes, 289–290**
- directives, conditional compilation, 717–719**
- disabling fields, 383–384**
- do . . while loops**
- basics, 92–93
 - continue statements and, 95
- DOCTYPE**
- DOM and, 347
 - switching, 345
- documents**
- dimensions, defining, 435
 - Document Object Model. See DOM (Document Object Model)
 - document.domain property, 477, 521
 - document.getElementById() function, 361–362
 - document.getElementsByName(name) static function, 360–361
 - document.getElementsByTagName function, 361
 - embedding CSS in, 403–405
 - fragments, 365–366, 646
 - geometry, 433–435
 - loading external (XML), 536–538
 - object, 914–920
 - trees, DOM and, 347–348
 - types, DOM and, 345–347
- DOM (Document Object Model)**
- basics, 341
 - browser compatibility and, 36

- buttons properties, 386
- CSS and, 407–416
- document tree, 347–348
- document types, 345–347
- DOM-base XSS, 477–478
- domready event, 331–333
- element attributes, 356–359
- form elements properties, 380
- grouping changes, 645
- history of, 341–345
- implementation object, 352–353
- movie events, 610–612
- node methods, 351–352
- node properties, 350–351
- node types, 348–349
- nodes, creating/deleting. *See* nodes (DOM)
- overview, 10–11
- ranges. *See* ranges (DOM)
- script execution and, 29
- specific elements, finding, 359–362
- traversing DOM, 353–356, 542–544
- Web pages and, 341

DOM reference

- Area object, 911–912
- body object, 920–922
- BUTTON/SUBMIT/RESET objects, 939
- Canvas object, 912–914
- caption object, 946
- CHECKBOX/RADIO objects, 940
- col/colgroup objects, 946
- cssRule (rule) object, 914
- document object, 914–920
- Event object, 922–926
- external object, 926
- FIELDSET object, 938
- FILE object, 943
- FRAME object, 969–970
- FRAMESET object, 970
- generic element, 927
- History object, 937
- HTML <form> tag, 937–938
- HTML <table> element, 943–945
- IFRAME object, 970–971
- IMAGE object, 940
- image object, 949–951
- INPUT objects, 939
- LABEL object, 939
- LEGEND object, 938
- link/anchor objects, 951–952
- Location object, 952

- navigator, 967
- POPUP object, 971
- Range object, 953–954
- screen, 968–969
- SELECT object, 942
- Selection object, 955
- Storage object, 956
- Style object, 957
- styleSheet object, 957–958
- tbody/tfoot/thead objects, 945–946
- td/th objects, 948–949
- TEXT/PASSWORD/HIDDEN objects, 941
- TEXTAREA object, 941–942
- TextRange object, 958
- tr object, 947
- TreeWalker object, 921–922
- window object, 960

domain attribute (cookies), 489

double submit, preventing (forms), 384–385

download speed

- page weight and, 633
- problem of, 631

downloading

- examples in this book, xxxiv
- from Web sites. *See* Web sites, for downloading

Dragonfly (Opera), 626

duck typing, 222

dynamic HTML (DHTML). *See* DHTML (dynamic HTML)

dynamic languages, 39–40

dynamic Web page content, 8

dynamically loaded scripts, 33–34

E

E4X (ECMAScript for XML), 552–553

ECMAScript. *See also* JavaScript

- defined, 2
- ECMAScript Harmony, 2–3
- ES5, 2–3
- revisions of, 14
- support and engine versions, 14

Edwards, Dean, 268

elements

- adding to arrays, 120–121, 224–227
- element attributes (DOM), 356–359
- element dimensions (DHTML), 437–438
- element object method, 357
- finding specific (DOM), 359–362
- form, 379–381

elements (continued)

- getting absolute position of, 447–451
- hidden, and code optimization, 645
- measuring, 646
- removing from arrays, 227–229
- XML DOM API, 540–542

else keyword, 85

embedding Flash movies, 504–505

enableInputs() function, 384

enabling fields, 383–384

encoding

- encodeURIComponent() function, 507
- encodeURIComponent()/decodeURI methods, 286
- encodeURIComponent()/decodeURIComponent() methods, 287
- HTML entities, 191
- and labels (JSON), 557–558
- RegExp symbols, 187
- strings, 151–154
- strings for URLs, 197, 285–287
- URI encoding, global objects and, 133–134

engines, JavaScript, 12–13, 15–16

Enumerator object (global), 782–783

Enumerator object (JavaScript), 782–783

equality, strict vs. loose, 65

equals (=) operator, 66

equivalence, comparison of strings and, 163–164

errata in this book, xxxiv

errors

- error codes (Safari), 498
- error handlers, 616
- error handling in JSON, 567
- Error object (JavaScript), 783–788
- Error object overview, 614–615
- handling, XML loading and, 538–539
- throwing custom, 615–616
- types of, 613–614

ES5, ECMAScript, 2–3

escape sequences

- string encoding and, 152
- string literals and, 44

escape()/unescape() methods, 286

eval() global function, 134–136

eval() method, security and, 485–486, 639–640

evalTest() function, 135

evaluated code

- avoiding, 638–639
- cleaning up, 119–120
- eval() global function and, 134–136
- JSON as, 558

evaluation expressions, defined, 39

events

- basic event model, 306–308
- body object, 921
- BUTTON/SUBMIT/RESET objects, 940
- CHECKBOX/RADIO objects, 940
- compatibility of, 336–338
- cross-browser event utility, 318–319
- custom, 338–340
- default behavior, preventing, 310
- default handlers for, 323
- document object, 919
- DOM movie events (QuickTime), 610–612
- DOM storage events, 496–498
- DOMContentLoaded event, 331–333
- event bindings, 326
- event bubbling, 343
- event-driven scripts, 31–32
- Event object, 314–318, 922–926
- FILE object, 943
- forms, 376–377
- generic element, 934–937
- getting mouse button, 318
- getting target, 317–318
- getting type, 317
- HTML <form> tag, 938
- IE mouse events, capture mode for, 322–323
- image object, 950–951
- inspecting event listeners, 314
- keystrokes, detecting, 326–327
- load/unload events, 330–331
- logs, 392–393
- mouse positions, 327–329
- mouseenter/mouseleave events, 333–335
- overview, 305–306
- propagation of, 319–324
- registration of, 308–309
- replicating, 324–326
- resize event, 330
- scroll event, 329–330
- SELECT object, 943
- Storage object, 495, 956
- TEXT/PASSWORD/HIDDEN objects, 941
- TEXTAREA object, 942
- this keyword and, 309
- unobtrusive event registration, 312–313
- unobtrusive JavaScript, 311
- window object, 966
- windows, 301–302

exception handling

- defined, 97
- statements, 96–98
- trapping exceptions, 616
- .exec() method, 180, 182**
- execCommand(), 397**
- execution and load order (scripts), 27**
- execution context**
 - eval() global function and, 135
 - functions and, 125–127
- expando properties, 229**
- expires attribute (cookies), 489**
- expressions, JavaScript**
 - basics, 61
 - Boolean objects and, 202
- Extensible Markup Language (XML). See XML (Extensible Markup Language)**
- Extensible Stylesheet Language Transformation (XSLT). See XSLT (Extensible Stylesheet Language Transformation)**
- external object, 926**
- external scripts, 30**
- ExternalInterface API (browsers), 594–600**
- ExternalInterface (Flash), 503, 524**

F

- factories, function. See function factories**
- Fiddler debugging tool, 626–627**
- fields (forms)**
 - enabling/disabling, 383–384
 - FIELDSET object, 938
 - file input, 396
 - hidden, 395
 - rich text fields, 396–401
 - setting focus to, 385
- FILE object, 943**
- filter object (IE), 405, 429–430**
- Firebug (Firefox)**
 - code profiling with, 636
 - Firebug Lite, 623–624
 - overview, 620–623
- Firefox**
 - for . . in loops and, 90–91
 - downloading, 19
 - security policies, 480–482
- FireUnit (testing), 628**
- first class objects, functions as, 103**
- fixed positioning (CSS), 444**
- flags, global, 172**

- Flash**
 - ActionScript in, 7
 - cross-domain Ajax and, 523–524
 - security and, 485
 - storage, 502–505
- Flash movies**
 - creating, 503–504
 - methods/properties, accessing, 599–600
 - overview, 594–595
 - setting up, 595–598
- floating-point literals, 43**
- floating point values, 204**
- flow-based layout models, 644**
- focus() method, 385–386**
- for each . . in loops**
 - basics, 91–92
 - continue statements and, 95
- for . . in iterator**
 - defining instance methods, 260
 - objects as arrays, 255
- for . . in loops**
 - basics, 89–91
 - code optimization and, 642–643
 - continue statements and, 95
- for loops**
 - basics, 87–88
 - continue statements and, 95
- formatting strings, 165–167**
- forms**
 - buttons, 386
 - checkboxes, 387
 - double-submit, preventing, 384–385
 - elements, 379–381
 - fields, enabling and disabling, 383–384
 - fields, file input, 396
 - fields, hidden, 395
 - fields, setting focus to, 385
 - Form object, 375–378
 - form tooltips (DHTML), 467–472
 - onsubmit event, 382
 - radio buttons, 387–389
 - rich text fields, 396–401
 - select and multiselect, 389–391
 - submissions on enter, preventing, 383
 - submitting and resetting, 381
 - textboxes/textareas/passwords, 391–395
- Forward buttons, Ajax and, 525–526**
- fragments, document, 646**

frames

- creating, 273
- defined, 273
- FRAME object, 969–970
- frame object model, 274
- referencing, 274–275

framesets

- basics, 273
- FRAMESET object, 970

Friedl, Jeffrey, 46

fromCharCode() method, 193

functions

- arguments, passing by value vs. *reference*, 106
- Boolean object and, 201
- closures. See closures
- declaring, 104–105
- execution context, 125–127
- extending replacement patterns with, 186
- Function object (JavaScript), 103–105, 788–798
- function pointers and code optimization, 640–641
- functRef, 264
- in Global object, 132–136
- isType(), 145–147
- JavaScript global, 899–907
- in JavaScript programming, 39
- nested, 118
- overloading. See function overloading
- passing as arguments (replace method), 186–187
- return values, 107
- scope, 125–127
- variable scope, 107–109

function factories

- basics, 120
- closures within loops and, 122–123
- pattern, 248

function overloading

- argument hashes, 115–118
- arguments object, 114–115
- basics, 109–112
- optional arguments, 112–114

function statements

- basics, 95–96
- browser support of, 82

G

garbage collection

- basics, 51

- circular references and, 123

Garrett, Jesse James, 511

Gears browser plugin (Google), 582–586

generic element, 927

Geolocation API (browsers), 580–582, 584

geometry, window and document, 433–435

GET requests (XMLHttpRequest), 519

getBoundingClientRect() function, 448

getElementById function, 361–362

getElementsByClassName static function, 360–361

getElementsByTagName function, 361

getStackTrace() function, 618

getters

- access to private members and, 263–264
- defined, 263
- getting date/time variables, 242
- reading/writing dates and, 240–241

global context, 107

global functions, JavaScript, 899–907

Global object

- in browser, 130–131
- defined, 272
- features of, 129
- functions added to, 105
- functions in, 132–136
- global objects, 136–137
- global objects, JavaScript. See JavaScript global objects
- numeric helpers, 136
- properties of, 131–132
- referencing, 131

global properties, JavaScript, 895–897

global scope, defined, 107

global variables, 272

GMT (Greenwich Mean Time), 236–237

Google Chrome, downloading, 19

groups within strings, 177–178

GZIP, 26, 631, 635–636

H

handlers

- default (events), 323
- event, 306

handling errors. See errors

Harmony, ECMAScript, 2–3

hasFeature() method, 368

hashes

argument hashes, 115–118
 basics, 229–231
 iterating and, 224

hasOwnProperty(), 140

headers, request/response (XMLHttpRequest), 518

Hewitt, Joe, 620

HIDDEN/TEXT/PASSWORD objects, 941

history
 History object, 937
 windows, 288

HTML (HyperText Markup Language)
 basic document structure, 23–24
 custom HTML tag formatters, 167
 dynamic. See dynamic HTML (DHTML)
 encoding entities, 191
 form object, 375–378
 <form> tags, 937–938
 formatting strings and, 165–167
 HTML5 client-side database, 498–502
 HTMLFormElement, 375–378
 JavaScript and, 8
 <table> elements, 943–945
 tags, stripping, 191

I

identifiers (strings), 49–50

IE (Internet Explorer)
 for . . in loops and, 90
 32-bit PNG images and, 459–460
 Developer Toolbar, 624
 DOCTYPE switching and, 345
 DOM and, 342–344
 DOM Inspector and, 358–359
 event object properties, 314–317
 expanded privileges in, 480
 filter object, 429–430
 JScript profiler, 636–637
 mouse events, capture mode for, 322–323
 parseError property, 538
 Same Origin Policy and, 477
 security zones, 482–483
 serializing XML to text in, 539
 signed scripts and, 480
 userData in, 492–494
 WYSIWYGs and, 396
 XPath queries in, 544
 XSL templates and, 550–551

if . . else statements, 83–85

iFrames (inline frames)
 basics, 273
 cross-domain Ajax and, 522
 iFrame approach to bookmarking, 526
 IFRAME object, 970–971
 WYSIWYGs as, 396

IIS (Internet Information Services) server, 18

IMAGE object, 940, 949–951

images
 DOM plus, 342
 image buttons, 386
 image injection, 522–523
 sprites, 634
 swapping (DHTML), 438–440

immutability of strings, 155

implementation property (DOM), 352–353

imported style sheets, 418–419

in operators, 77, 80

increment (++) operators, 75, 80

indexed arrays, 219

.indexOf() method
 RegExp object and, 182–183
 searching strings within strings, 169–170

Infinity property (Global object), 131

Infinity value, 209

inheritance
 alternate approaches, 268–270
 basics, 137, 264–265
 prototypal inheritance, 140
 prototype-based subclassing, 265–267
 prototypes, problems with, 267–268

initializing
 initializers for event types, 325
 UserData, 492–493

inline event binding, 32

inline scripts, 26, 28–29

innerHTML property, 363

inputs
 forms, 379–380
 INPUT objects, 939

Inspector, DOM, 358–359

installing Gears plugin, 582–583

instances
 instance properties (RegExp object), 180–181
 instanceof operator, 78, 80, 221–222
 of objects, creating, 138
 of regular expressions, creating, 172–173

integers
 integer literals, 42
 Number object and, 204

interfaces

- defined, 268
- function interfaces, arguments object and, 114–115
- prototype problems and, 268

Internet Explorer. See **IE (Internet Explorer)**

interpreted languages, 5

intervals, timers and, 245–248

Ippolito, Bob, 569

isArray() method, 145

isBoolean() method, 145

isDate() method, 146

isFinite() function, 136

isFunction() method, 146

isNaN() function, 136, 208

isNull() method, 146

isNumber() method, 146

isObject() method, 146

isRegex() method, 146

isString() method, 147

isType() functions, 145–147

isUndefined() method, 147

iterating

- iterators and for loops, 87
- over arrays, 223–224
- over DOM nodes, 358

J

Jagged Arrays, 220

Java Applets, 591–594

JavaScript. See also **browsers, JavaScript in**

- ActionScript in Flash, 7
- AIR and, 7
- BOM and, 9–10
- browser development, 4–5
- comments (reference), 716–721
- controlling QuickTime movies from, 608–609
- CSS and, 8–9
- desktop widgets and, 7
- developers path for learning, 3–4
- DOM and, 10–11
- embedding Silverlight movies with, 603
- engines, 12–13, 15–16
- global functions, 899–907
- global properties, 895–897
- Hello World application, 19–21
- history of, 1–2
- HTML and, 8
- limitations in browsers, 12

literal notation, 556–557

object model equivalencies, 15

Object Notation (JSON). See **JSON (JavaScript Object Notation)**

object oriented development and, 251–253

online resources for development, 973–974

prevalence of, xxix

server-side, 5

Silverlight communication and, 604–606

statements (reference), 699–715

testing with browsers, 18–19

text editors, 17

unobtrusive, 311

uses for, 11–12

Web servers and, 17–18

for XML. See **E4X (ECMAScript for XML)**

JavaScript basics, 39–59

- blocks, 47
- case sensitivity, 41
- character encoding, 40–41
- closures, 47
- comments, 48
- data types. See **data types**
- dynamic languages, 39–40
- identifiers, 49–50
- implicit declaration, 49
- literals, 42–46
- memory and garbage collection, 51
- prototype-based languages, 40
- reserved words, 48
- statements, 46
- variables, 48–51
- weak typing, 50
- whitespace and semicolons, 42

JavaScript global objects

- ActiveXObject, 725
- Array object, 725–746
- Boolean object, 746
- Date object, 754–781
- Debug object, 781–782
- Enumerator object, 782–783
- Error object, 783–788
- Function object, 788–798
- JSON object, 798–802
- listed, 723–724
- Math object, 802–817
- Number object, 818–826
- Object() object, 827–841
- RegExp object, 841–851
- String object, 852–875

VArray object, 875–882
XMLHttpRequest object, 882–894

JavaScript operators (reference)

arithmetic, 654–660
assignment, 660–669
bitwise, 669–674
comparison, 674–680
listed by category, 649–654
logical, 680–682
miscellaneous, 688–698
string, 682–688

Johnson, Dave, 268

join() method, 222, 233–234

JScript, Microsoft

background, 5–6
profiler, 636–637
versions of, 14

JSLint, 635

JSON (JavaScript Object Notation)

Ajax and, 567–569
custom replacement function, 564
custom revivers, 565–566
custom toJSON() methods, 562–563
error handling, 567
eval() (security), 485–486
as evaluated code, 558
global object, 798–802
JavaScript literal notation and, 556–557
JSONP (JSON with Padding), 569–570
labels and encoding, 557–558
loading JSON data, 564–565
object literal format and, 230
overview, 555–556
security and, 559
serializing objects to, 560–562
vs. XML, 559–560

JSUnit (testing), 628

K

keystrokes, detecting (events), 326–327

keywords

reserved words, 48, 909
var keyword, 49

L

labels

break statements and, 94, 95

and encoding (JSON), 557–558
LABEL object, 939
label statements, 93–95

languages

detecting in browsers, 293–294
dynamic, 39–40
prototype-based, 40

.lastIndexOf() method, 169–170

latency, defined (servers), 633

layers in Netscape Navigator, 343

Layout Engine, 12

left() function, 192–193

LEGEND object, 938

length

of arrays, 223–224
property (strings), 155–156

line breaks

in string literals, 44
in strings, 154

linear animation, 453

link/anchor objects, 951–952

.link(linkUrl) method, 166–167

listeners, event, 312, 314

literal notation

array literal notation, 219
JSON and, 556–557

literals

array, 45
Boolean, 43
floating-point, 43
integer, 42
number, 204–205
object, 46
regular expression, 46
string, 43, 151, 154
vs. variables, 42

little endian, defined, 71

loading

dynamic, 33
JSON data, 564–565
load/unload events, 330–331
post-loading JavaScript, 634
XML. See XML, loading

local scope, defined, 107

localeCompare() method, 164

localStorage (DOM storage), 495

Location object (window object), 284, 952

logical AND/OR statements, 70

logical expressions, 61

logical operators

- basics, 62, 68–70
- reference, 680–682

long-form operations, 67

lookup of variables, local vs. global, 640

loops

- for . . in, 89–91, 642–643
- basics, 87
- break/label/continue statements and, 93–95
- closures with, 121–123
- do . . while, 92–93
- for each . . in, 91–92
- for, 87–88
- tuning and code optimization, 643
- while, 92

loose typing, 50, 161

M

masking, script, 27

Mastering Regular Expressions (Friedl), 46

.match() method, 184

Math object

- basics, 199, 210–211
- math utility methods, 212–213
- mathematical constants, 211–212
- random numbers, 213–214
- reference, 802–817
- repeated math calls, 214–215
- rounding numbers, 213–214

math operations on date values, 242–244

max-age attribute (cookies), 490

memory

- accidental closures and, 125
- basics of, 51
- pointers, primitive data types and, 52

merging objects, 142–143

methods, 514, 803–804, 945, 946

- to add elements to arrays, 225
- Array class, 218
- Array object, 726–728
- body object, 921
- Boolean object, 747, 750–753
- Boolean wrapper class, 200
- BUTTON/SUBMIT/RESET objects, 939
- Canvas object, 913–914
- capitalization in strings and, 167–198
- CHECKBOX/RADIO objects, 940
- comparison of (cross-domain Ajax), 520
- for creating nodes, 363–364

- for cutting pieces from strings, 192
- date instance methods, 240–241
- defined, 253
- document object, 918–920
- Error object, 615, 784
- Event object, 925–926
- external object, 926–927
- FILE object, 943
- for formatting strings, 165–167
- forms, 376
- Function object, 790
- generic element, 931–934
- History object, 937
- HTML <form> tag, 938
- JSON object, 799
- Location object, 953
- Math object, 211
- math utility methods, 212–213
- navigator, 293, 968
- node methods, 351–352, 541–542
- Number object, 819
- Number wrapper class, 204
- Object() object, 139, 828
- POPUP object, 971
- Range object, 953–954
- RegExp object, 180, 842
- for removing items from arrays, 227
- for returning strings from numbers, 205–206
- for search and replace, 169
- SELECT object, 943
- Selection object, 955–956
- simulating private methods, 120–121
- Storage object, 495, 956
- String object, 852–854
- String wrapper class, 150
- styleSheet object, 417, 958
- TEXT/PASSWORD/HIDDEN objects, 941
- TEXTAREA object, 942
- TextRange object, 959–960
- tr object, 948
- TreeWalker object, 922
- VBAArray object, 876
- window object, 963–965

Microsoft

- DOM history and, 341
- IE Developer Toolbar, 624–625
- Internet Explorer. See IE (Internet Explorer)
- security model, 475–476
- Silverlight. See Silverlight movies
- Visual Web Developer Express Edition, 17

Miller Device, 222**Miller, Mark, 222****milliseconds in units of time, 236****minifying JavaScript, 634–635****modal dialogues/windows (example), 460–467****modal/modeless windows, 272****moduleName values, 351–352****moduleVersion values, 351–352****modulus operators (%), 74–75****mouse**

button, getting, 318

events, capture mode for (IE), 322–323

mouseenter/mouseleave events, 333–335,
441–443

positions, getting (events), 327–329

movies

Flash. See Flash movies

movie events with DOM, 610–612

QuickTime, controlling from JavaScript,
608–609

Silverlight. See Silverlight movies

moving/resizing windows, 288**Mozilla**

expanded privileges in, 479–480

JavaScript Edition, versions of, 14

security policies, 480–482

signed scripts and, 479–480

multi-dimensional arrays, 45, 220**multiline comments, 716****multiple inheritance, defined, 268****multiselect boxes (forms), 389–391****mutators**

mutator methods (DOM), 364

performance comparison of, 366–367

N

name attribute (cookies), 489**NamedNodeMap property, 356****NaN property**

non-numeric values and, 131–132

Number object and, 208–209

navigator

navigator.plugins array, 606

object (browsers), 292

properties and methods, 967

nesting

functions, 118

nested arrays, 219

quotes, string literals and, 43–44

Netscape Navigator

history of DOM and, 341–343

security model, 475–476

Neuberg, Brad, 529**new keyword**

class definitions and, 258

creating objects and, 253

Object object and, 138

new operators, 78, 80**nodes (DOM)**

adding new, 363–365

document fragments, 365–366

methods, 351–352

mutators, comparison of, 366–367

NodeList property, 354–355

nodeType property, 350, 352–353

properties, 350–351

removing, 367

repaints and reflows, 365

swapping, 367–368

types, 348–349

XML DOM API, 540–542

non-deferred scripts, 31**non-linear animation (DHTML),
453–457****nonpersistent XSS, 478****<noscript> element, 27****Notepad++, 17****null type, 53–54****number literals, 204–205****Number object**

basics, 199, 203–204

converting to numbers, 207

integer and floating point
values, 204

minimum/maximum values, 209

NaN and, 208–209

number literals, 204–205

numbers and strings, 205–206

reference, 818–826

numbers

comparison operators and, 64

converting strings to, 207

converting types to, 56

global functions dealing with, 136

Number object and, 203

random, 213–214

rounding, 206, 213

strings and, 161, 205–206

numeric helpers (global functions), 136

O

objects

- for . . in loops and, 89
 - alteration at runtime, 40
 - arguments object, 114–115
 - basics, 253
 - comparison operators and, 65
 - creating, 253
 - deleting, 253–254
 - deleting unused, 638–639
 - first class, 103
 - global, defined, 105. *See also* JavaScript global objects
 - object initializers, 45
 - object literal notation, 138
 - object literals, 46, 114, 555–556
 - object model equivalencies, 15
 - object pointers and code optimization, 640–641
 - object prototypes, 4
 - object signing, 479
 - Object.__defineSetter__() method, 264
 - objectType() operand, 78
 - or primitives, strings as, 156–158
 - primitive, vs. primitive data types, 58
 - serializing to JSON, 560–562
- objects, utility functions for**
- isType() functions, 145–147
 - merging objects, 142–143
 - typeof operator and, 143–145
- Object object class**
- basics, 137–139
 - object prototypes, 139–140
 - properties, 140
 - reference, 827–841
 - utility functions for objects. *See* objects, utility functions for
 - valueOf() and toString() functions, 141
- object oriented development**
- arrays, objects as, 255–256
 - cloning objects, 257
 - comparing objects, 256–257
 - constructors, 258–259
 - creating objects, 253
 - inheritance. *See* inheritance
 - JavaScript and, 251–253
 - overview, 251
 - private members, 261–264
 - properties and objects, deleting, 253–254

- prototypes, 259–260
- reference types, objects as, 254
- static members, 258
- this keyword, 261

obtrusive event binding, 32

onstorage event, 496

onstoragecommit event, 496

onsubmit event, 382

OO programming. *See* object oriented development

opacity of elements (browsers), 458–459

opening/closing windows, 276–281

OpenSearch standard (browsers), 586

Opera, Dragonfly debugging tool for, 626

operators

- arithmetic operators, 654–660
 - assignment, 66–68, 660–669
 - bitwise, 70–73, 669–674
 - combinational (connubial), 73–75
 - comparison, 63–66, 674–680
 - defined, 61
 - to identify groups within strings, 177
 - listed by category, 649–654
 - logical, 68–70, 680–682
 - miscellaneous, 688–698
 - precedence and associativity of, 79–81
 - string, 163, 682–688
 - types of, 62
 - uncommon, 75–79
- opposite quotation marks (string literals), 44**
- OS (operating system) detection class, 295–300**

overloading

- function. *See* function overloading
- operators, 74

owningElement (IE), 423

P

page weight, reducing, 632–636

parent frames, 274

parent windows

- communicating with, 283–284
- defined, 274

parentheses in complex expressions, 85

parsing

- date strings, 238–239
- parseError property, 538
- parseFloat() function, 136
- parseInt() function, 136

passwords

- forms and, 391–395
- PASSWORD/TEXT/HIDDEN objects, 941

path attribute (cookies), 489**Penner, Robert, 453****performance**

- comparison of mutators and, 366–367
- regular expressions and, 171
- string operations and, 159–191

performance tuning

- categories of problems, 631–632
- code optimization. *See* code optimization
- code profiling, 636–638
- DHTML optimization, 644–647
- page weight reduction, 632–636

Perini, Diego, 332**persistence**

- client-side data and persistence. *See* client-side data and persistence
- persistent XSS, 478

phases, connection (XMLHttpRequest), 516**phishing attempts, 477****plugins, scripting**

- Flash. *See* Flash movies
- Java Applets, 591–594
- movie events with DOM, 610–612
- overview, 591
- QuickTime (Apple), 606–609
- Silverlight. *See* Silverlight movies

plus (+) operator, 21**PNG (Portable Network Graphics) format, 459****pointers, primitive data types and, 52****policies, security, 480–482****polymorphism. *See* function overloading****pop() method, 227****POPUP object, 971****positioning, absolute/relative (CSS), 443–445****post-loading JavaScript, 634****precedence of operators, 79–81****primitive data types**

- composite to primitive conversion, 57–58
- conversion of, 55–57
- defined, 51
- passed by value and reference, 106
- vs. primitive objects, 58
- vs. reference data types, 51–52, 231

primitives

- Boolean, creating, 200–202
- or objects, strings as, 156–158
- string literals and, 151

- string primitives, 149
- testing alphabetically, 163
- valueOf() function and, 141

printing date strings, 244**private/privileged members, 261–264****privileges, expanded (Mozilla), 479–480****programmatic event registration, 308****prompt() dialogue boxes, 289–290****propagation of events, 319–324****properties**

- Area object, 911–912
- Array class, 217
- Array object, 726
- body object, 920–921
- Boolean object, 746, 750
- Boolean wrapper class, 200
- BoxObject, 449
- BUTTON/SUBMIT/RESET objects, 939
- Canvas object, 912–913
- caption object, 946
- CHECKBOX/RADIO objects, 940
- col/colgroup objects, 946–947
- cssRule (rule) object, 914
- custom event object, 497–498
- Date class, 235
- defined, 253
- deleting, 253–254
- determining existence of, 140
- document object, 915–918
- DOM buttons, 386
- DOM, CSS attributes and, 407–416
- Error object, 614–615, 784
- Event object, 922–925
- external object, 926
- FIELDSET/Legend objects, 938
- FILE object, 943
- form elements, 380
- forms, 376, 380
- FRAME object, 969–970
- FRAMESET object, 970
- Function objects, 103–104, 789
- generic element, 927–931
- Global object, 131–132
- History object, 937
- HTML <form> tag, 937–938
- IE event object, 314–317
- IFRAME object, 970–971
- IMAGE object, 940, 949–950
- INPUT objects, 939
- JavaScript global, 895–897

properties (continued)

- LABEL object, 939
- link/anchor objects, 951–952
- Location object (windows), 284–285, 952
- Math object, 210, 803
- mathematical constants and, 211
- for mouse coordinates, 327
- navigator, 292–293, 967–968
- nodes, 350–351, 541–542
- Number object, 818
- Number wrapper class, 203
- Object() object, 138, 827
- POPUP object, 971
- Range object, 953
- RegExp object, 180–183, 841–842
- rule object, 418
- screen object, 294, 968–969
- <script> element, 24–26
- SELECT object, 942
- Selection object, 955
- Storage object, 495, 956
- String object, 852
- String wrapper class, 149
- Style object, 957
- style, reading/writing, 424–425
- styleSheet object, 417, 957–958
- td/th objects, 948–949
- TEXT/PASSWORD/HIDDEN objects, 941
- TEXTAREA object, 942
- TextRange object, 958–959
- tr object, 947
- TreeWalker object, 921–922
- window object, 960–963
- XMLHttpRequest, 513

prototypes

- basics, 259–260
- defined, 40
- object, 4, 139–140
- problems with, 267–268
- prototype-based languages, 40
- prototype chaining, 265–267
- prototype method, 195

prototypal inheritance, 139, 252, 259

prototype property

- comparing objects and, 256
- creating properties with, 260
- extending String object with, 158
- merging objects and, 142
- prototypal inheritance and, 139–140

proxy scripts (Ajax), 521–522

proxy trace debugging tools, 626–627

pseudo-random numbers, 213

pseudo-threads (animation), 451–453

public members, defined, 260

push() method, 225, 226–227

Q

QuickTime (Apple), 606–609

quirks mode, 328, 345–346

quotation marks in string literals, 43–44

R

radio buttons, 387–389

random() method, 213

ranges (DOM)

- basics, 368–369
- boundaries, 369–370
- collapsing, 371
- content, changing, 370–371
- defined, 368
- Range object, 953–954
- user selection, 371–373

reading

- cookies, 490–491
- dates, 240–244
- DOM storage, 496
- SQLite data, 500–501
- UserData, 493–494

readyState property (XMLHttpRequest), 516

Really Simple History (Neuberg), 529

redraws. See repaints

reference data types

- arrays as, 231–233
- comparing, 65
- defined, 156
- objects and, 117, 142–143
- objects as, 254
- vs. primitive data types, 51–52, 231

references

- circular, 123–125
- to forms, 377, 378
- manipulating by value vs. reference, 52–53
- passing argument by, 106
- referencing frames, 274–275

reflows (DOM)

- basics, 365

defined, 644
 DHTML optimization and, 644–645
 measuring elements and, 646
 multiple, 646

RegExp object
 basics, 180–181
 reference, 841–851
 searching with, 182
 static properties, 182–183
 symbols, 185–186

RegisterScriptableObject() function, 603–604

registration of events
 basics, 308–309
 unobtrusive, 312–313

regular expressions (RegExp)
 alternatives, 178–179
 basics, 171–172
 contains method, improved, 181–182
 defining, 172–173
 groupings, 177–178
 literals, 46
 object. *See* RegExp object
 pattern reuse, 179–180
 position, 176
 repetition, 174–175
 special characters, 173–174
 splitting on, 196–197
 symbols, encoding, 187

relative positioning (CSS), 444–445

remote debuggers, 626

repaints
 basics, 365
 defined, 644
 DHTML optimization and, 644–645

repetition symbols (RegExp), 174–175

replace() method, 184–187

replacement function, custom (JSON), 564

replicating events, 324–326

request headers (XMLHttpRequest), 518

reserved keywords, 48, 909

RESET/BUTTON/SUBMIT/ objects, 939

reset() method, 376, 381

resetting forms, 381

Resig, John, 3

resize event, 330

resizing/moving windows, 288

response headers (XMLHttpRequest), 518

resultType constants, 545

return statements, 95–96, 107

reverse loops, defined, 224

revivers, custom (JSON), 565–566

right() function (Visual Basic), 192–193

rollovers (dynamic HTML)
 image swapping and, 438–440
 and mouseenter/mouseleave events, 441–443

rules (CSS)
 adding/removing, 426–427
 iterating over all, 423
 searching for, 424

runtime errors, 614

S

S3 (Simple Storage Service), 636

Safari

downloading, 19
 Safari 4 SQLite database browser, 502
 Web Inspector, 627

Same Origin Policy, 274, 476–479, 519, 574

Same Site Policy, 476–479

sandbox, security, 476

saving Flash movies, 504

scope (variables)

basics, 47
 execution context and, 125–127
 functions and, 107–109
 scope chain, 108

screens

object, 294–295
 properties, 968–969

scripts

blocks, 19
 masking, 27
 <script> element, 25–26
 <SCRIPT> injection (cross-domain Ajax), 523
 scriptable objects (browsers), 343
 scripting plugins. *See* plugins, scripting
 scripting z-index (CSS), 446–447
 security and, 476
 signed, 479–480
 types of, 28–34

scroll event, 329–330

scrollbars, width of (DHTML), 435–437

scrolling windows, 289

search providers, Internet, 586

searching and replacing (RegExp)

based on user input, 188
 basics, 168–169
 .indexOf() and .lastIndexOf() methods, 169–170
 .match() method, 184

searching and replacing (RegExp) (continued)

replace() method, 184–187
.search() method, 183–184

security

ActiveX, 484–485
data security, 484
Denial of Service attacks, 483–484
eval() function and JSON, 485–486
Flash, 485
JSON and, 559
new windows, 483
policies and zones, 480–483
Same Origin Policy, 476–479
secure attribute (cookies), 490
security models, 475–476
signed scripts, 479–480
XMLHttpRequest, 518

select boxes (forms), 389–391

SELECT object, 942

Selection object, 955

selectors, CSS, 405

selectorText property (CSS rules), 418

Selenium testing suite, 628

semantic errors, 614

semicolons (;) in JavaScript programming, 42

serializing

objects to JSON, 560–562
XML to text, 539–540

server-side JavaScript, 5

server-side proxy method, 559

servers

opening connections (XMLHttpRequest),
514–517
server proxies (cross-domain Ajax), 521–522

sessionStorage (DOM storage), 495

setEnd() function, 369, 370

setInterval() function, 245–246, 452

setStart() function, 369

setters

access to private members and, 263–264
defined, 263
reading/writing dates and, 240–241

setTimeout()

function, 245–248
method, 119
timer function, 452

SharedObject object (Flash), 502–503

shift() method, 228

short-form operations, 67

signed scripts, 479–480

Silverlight movies

communication with JavaScript, 604–606
embedding with JavaScript, 603
RegisterScriptableObject() function, 603–604
setting up, 601–602

single comments, JavaScript, 716

Single Origin Policy, 476–479

single quotes (') in string literals, 151, 153

single-threaded, defined (JavaScript), 245

size

of arrays, 223
autosizing textareas, 395

slice() method, 193–194, 228

sMethod argument (Ajax), 514

source code, downloading, xxxiv

special characters

in regular expressions, 173–174
stripping, 190–191
Unicode and, 41

splice() method, 228

split() method, 196, 233–234

spriting

in DHTML, 439
to improve page download times, 634

SQLite databases, 498–502

square brackets ([]) in regular expressions, 173

stack trace function, 616–619

standards mode. See quirks mode

statements

basics, 46, 82–84
browser support of, 82–83
conditional, 84–87
conditional compilation, 719
defined, 20
exception handling, 96–98
function, 95–96
JavaScript (reference), 699–715
loops and. See loops
miscellaneous, 99–100

static Date.parse() method, 238

static Math.round() function, 213

static members, 258

static methods

defined, 258
referencing from String.fromCharCode(), 193

static properties

defined, 258
RegExp object, 182–183

static scoping, 109

status bar

- browser, 276
 - windows, 276
 - status codes, Ajax requests and, 516**
 - storage. See also persistence**
 - Storage object, 495, 956
 - W3C DOM, 495–498
 - strict equality (comparison operators), 66**
 - .strike() method, 165**
 - strings**
 - arrays and, 196–197, 233–234
 - assignment operators and, 68
 - case and, 167–168
 - comparing, 163–164, 189–190
 - comparison operators and, 64
 - concatenation of, 159–161
 - converting primitive types to, 57
 - converting to, 161–162
 - creating, 156
 - cutting pieces from, 192–195
 - encoding, 151–154
 - encoding for URLs, 197, 285–287
 - extracting characters from, 192–193
 - formatting, 165–167
 - HTML entities, encoding, 191
 - HTML tags, stripping, 191
 - immutability of, 155
 - length of, 155–156
 - line breaks in, 154
 - literals, 151
 - non-alphanumeric characters, stripping, 190–191
 - numbers and, 161, 205–206
 - operators, 682–688
 - primitive data types and, 58
 - primitives and objects, 156–158
 - printing date strings, 244
 - searching and replacing. *See* searching and replacing (RegExp)
 - string literals, 43–44
 - String object, extending, 158–159
 - String object, overview, 149–150
 - stringify() method (JSON), 562–563
 - String.split() method, 233–234
 - trimming whitespace and, 188–189
 - working with, 164
 - String object (JavaScript)**
 - extending, 158–159
 - overview, 149–150
 - reference, 852–875
 - strong typing, 50**
 - style sheets**
 - adding/removing, 422–423
 - imported, 418–419
 - iterating over all, 419–422
 - StyleSheet object, 416–418, 957–958
 - styles**
 - computed (CSS), 428–429
 - grouping style changes, 646
 - properties, reading/writing, 424–425
 - Style object, 957
 - submissions, Web form, 381, 383–385**
 - SUBMIT/BUTTON/RESET objects, 939**
 - submit() method, 376, 381**
 - substring() method, 193–194**
 - SunSpider JavaScript performance, 16**
 - sUrl argument (Ajax), 515**
 - swapping nodes, 367–368**
 - SWFObject, embedding Flash movies with, 598–599**
 - switch statements**
 - basics, 86–87
 - browser support of, 83
 - symbols**
 - encoding RegExp symbols, 187
 - to identify groups within strings, 177
 - position syntax and, 176
 - in RegExp expressions, 173–174
 - RegExp object, 185–186
 - repetition symbols, 174–175
 - synchronous requests (XMLHttpRequest), 515**
 - syntax errors, 613**
- ## T
- tags**
 - HTML, preserving, 191
 - tag formatters (strings), 167
 - targets, event, 312**
 - tbody/tfoot/thead objects, 945–946**
 - td/th objects, 948–949**
 - templates, XSL, 550–552**
 - testing**
 - with browsers, 18–19
 - .test() method, 180–182
 - tools for, 628
 - text**
 - automatic selection in textboxes, 394
 - deserializing (XML), 533–536
 - editors, JavaScript, 17
 - inputs, 391

text (continued)

- masking inputs, 393–394
- serializing XML to, 539–540
- text entry fields, 391
- TEXT/PASSWORD/HIDDEN objects, 941
- TEXTAREA object, 941–942
- textboxes/textareas (forms), 391–395
- Textmate, 17
- TextRange object, 958
- values, toString() function and, 141
- this keyword, 261, 309, 381**
- threading for long-running tasks, 647**
- threeArguments() function pointer, 110–111**
- throw statements, 83, 96–97**
- throwing custom errors, 615–616**
- time**
 - dates and, 236–237
 - measuring differences, 244–245
 - world time overview, 237–238
- timers**
 - and intervals, 245–248
 - pseudo-threading with, 452–453
- toExponential() method, 206**
- toFixed() method, 206**
- toJSON() methods, 560–563**
- toLocaleString() method, 206**
- .toLowerCase() method, 168**
- tooltips, form (DHTML), 467–472**
- toPrecision() method, 206**
- toString() function, 141**
- toString() method, 57–58, 205, 222, 244**
- .toUpperCase() and toLowerCase() methods, 150, 168**
- tr object, 947**
- traditional event registration, 308**
- transmissions, piggy-back, 479**
- trapping exceptions, 616**
- TreeWalker object, 921–922**
- Trusted Sites zone, 484**
- try . . . catch . . . finally statements**
 - code optimization and, 642
 - error handlers and, 97–98
- tuning loops, code optimization and, 643**
- Turner, Doug, 580**
- tweening (DHTML animation), 453–457**
- two's complement format (Bitwise operators), 70–71**
- type casting, 57**
- type coercion**
 - defined, 53

- optional arguments and, 112–113
- type property (cssRule object), 418**
- typeof operators, 54, 78–80, 143–145, 220**

U

- undefined data types, 53–54**
- undefined global properties, 131–132**
- undefined values, return statements and, 96, 107**
- underflow values, defined, 209**
- Unicode character standard**
 - basics, 40
 - character encoding and, 40
 - string encoding and, 151–152
- unload/load events, 330–331**
- unobtrusive event attachment, 382**
- unobtrusive event registration, 312–313**
- unobtrusive JavaScript, 311**
- unshift() method, 225**
- URI encoding, 133–134**
- URLs (Uniform Resource Locators)**
 - encoding strings for, 197, 285–287
 - JavaScript in, 35
 - length limits of, 287
- user selection ranges (DOM), 371–373**
- UserData in IE, 492–494**
- UTC (Coordinated Universal Time), 238**

V

- validation of forms, 383**
- values**
 - date/time values, 242
 - deleting from storage, 508
 - floating point values, defined, 204
 - minimum/maximum, 209
 - objects as, 199
 - passing arguments by value, 106
 - primitive, Booleans as, 200
 - radio button fields and, 387
 - return values and functions, 107
 - underflow, defined, 209
 - value property for text input, 392
 - valueOf() function, 141, 201, 203
 - valueOf() method, 57, 237
- var keyword, 49**
- var statements, 99**
- variables**
 - conditional compilation, 720–721

- declaring, 48–51
- global, 272
- identifiers, 49–50
- implicit declaration, 49
- vs. literals, 42
- local vs. global variable lookup, 640
- memory and garbage collection, 51
- modifying and comparing, 55
- optional arguments and, 113
- variable scope, 107–109
- weak typing, 50

VBAArray object (JavaScript), 875–882**Vector Markup Language (VML), 587–588****versioning, database (SQLite), 498****vertical pipe (|) alternation symbol, 179****Visual Web Developer Express Edition (Microsoft), 17****VML (Vector Markup Language), 587–588****void operators, 78–80****W****W3C**

- DOM, 344–345
- DOM storage, 495–498

weak typing, 50**Web Inspector (Safari 4), 627–628****Web pages, DOM and, 341****Web servers, JavaScript, 17–18****Web sites, for downloading**

- AC_Quicktime.js JavaScript library, 607
- Charles Proxy Debugger, 627
- data provider plugin for geolocation, 580
- Fiddler, 626
- Firebug, 620
- gears_init.js file, 583
- Silverlight Tools for Visual Studio, 601
- testing tools, 628
- text editors, 17
- YSlow, 638

Web sites, for further information

- Ajaxslt (Google), 362
- DOM standard, 23
- errata in this book, xxxiv
- JavaScript development, 973
- JavaScript engines, 13
- JavaScript-XPath library, 362
- JSLint, 635
- P2P, xxxiv–xxxv
- Penner, Robert, 453

- RegExp syntax tutorials, 46

- server-side JavaScript, 6

- Web Accelerator (Google), 519

Web Standards Project (WaSP), 406**Web Workers threading module (browsers), 588–589****while loops**

- basics, 92
- continue statements and, 95

whitespace

- in JavaScript programming, 42
- trimming (strings), 188–189

widgets, desktop (JavaScript), 7**window.name attribute**

- data security and, 484
- storage and, 505–508

windows

- events, 301–302
- geometry, 433–435
- history, 288
- loading content into new, 282–283
- manipulating, 275
- moving/resizing, 288
- new windows, security and, 483
- object, 272, 960
- opening/closing, 276–281
- parent, communicating with, 283–284
- property, 130
- scrolling, 289
- setting location of, 284–285
- status bar, 276
- window feature values, 277–279
- window.close() method, 282
- window.external.AddService() method, 574
- windowFactory() function, 281, 283

with keyword, 99–100**with { } statement**

- avoiding, 641
- defined, 99–100
- repeated math calls and, 214–215

WorkerPool threading module (Gears), 584–586**world time overview, 237–238****World Time Standard, 236–237****wrapper classes**

- global Array object and, 217
- strings and, 149
- wrapper class String, 156

wrappers, 138

writing

- dates, 240–244
- to DOM storage, 496
- SQLite data, 500–501
- UserData, 493–494

WYSIWYG editor, 396–401

X

XHR. See XMLHttpRequest (XHR)

XML (Extensible Markup Language), 533–553

- data, transforming with XSLT, 548–552
- E4X, 552–553
- vs. JSON, 559–560
- loading. See XML, loading
- serializing to text, 539–540
- XML Path. See XPath (XML Path language)

XML DOM API

- elements and nodes, 540–542
- traversing DOM, 542–544
- XPath queries, 544–547

XML, loading

- deserializing text, 533–536
- external documents, 536–538
- handling errors, 538–539
- with XMLHttpRequest (XHR), 538

XMLHttpRequest (XHR)

- basics, 512–514
- connections, opening, 514–517
- cross-domain Ajax and, 524–525
- GET requests to change data, 519
- object, 882–894
- request/response headers, 518
- security, 518

XOR operator, 202–203

XPath (XML Path language)

- locating elements in DOM, 362
- queries, 544–547

XSL (Extensible Stylesheet Language), 550–552

XSLT (Extensible Stylesheet Language Transformation), 548–552, 560

XSS (Cross-Site Scripting), 477–478

Y

yellow-fade technique, 457–458

YSlow plugin, 638

Z

z-index, scripting (CSS), 446–447

zones

- security, 480–482
- time, 237–238