

Index

SYMBOLS

- +** (add operator), 47–48
- &** (AND operator), 39–40
- =** (assignment operator), 52–53
- /*** (begin multiline comment), 12
- {}** (braces) for code block, 12
- ^** (caret)
 - in regular expressions, 199
 - as XOR operator, 40–41
- ,** (comma operator), 53
- (dash) in regular expression, 199–200
- /** (divide operator), 47
- \$** (dollar sign) and variable name, 13
- ** (double escaping), 197, 198
- "** (double quotation marks)
 - internationalization and, 572–573
 - string literal and, 21
- */** (end multiline comment), 12
- ==** (equal operator)
 - dates and, 380
 - description of, 50–51
- //** (forward slashes) to begin single-line comment, 12
- >** (greater than operator), 49–50
- >=** (greater than or equal to operator), 49–50
- ===** (identically equal operator), 51–52
- <<** (left shift operator), 41–42
- <** (less than operator), 49–50
- <=** (less than or equal to operator), 49–50
- &&** (logical AND operator), 44–45
- !** (logical NOT operator), 43–44
- ||** (logical OR operator), 45–46
- (minus operator), 36–37
- %** (modulus [remainder] operator), 47
- *** (multiply operator), 46
- !=** (not equal operator), 50–51
- !==** (not identically equal operator), 52
- ~** (NOT operator), 39
- |** (OR operator), 40
- (postfix decrement operator), 35–36
- ++** (postfix increment operator), 35–36
- (prefix decrement operator), 34–35
- ++** (prefix increment operator), 34–35
- ;** (semicolon)
 - as optional at end-of-line, 12
 - using, 574
- >** (signed right shift operator), 42
- '** (single quotation mark)
 - internationalization and, 573
 - string literal and, 21
- []** (square brackets)
 - Array class and, 71
 - in regular expressions, 199
 - XPath and, 466
- (subtract operator), 48–49
- +** (Unary plus operator), 36
- _** (underscore)
 - to indicate private scope, 88
 - variable name and, 13
- >>** (unsigned right shift operator), 42–43

A

abort event, 286

abs() method of Math object, 85

abstract class, 104–105

accessing

form field, 338

style sheet, 309–312

ActiveX control

embedding, 558

MSXML and, 445, 448, 450

ActiveX object, 533

ActiveX plugin

description of, 535

detecting, 540–542

ActiveXObject class, 445–446

add operator (+), 47–48

addEventListener() method of EventUtil object, 292

addEventListener() method (DOM), 268–269, 407

additive operators

add (+), 47–48

subtract (-), 48–49

addParameter() method (XSLT), 475

addPostParam() function, 498

addURLParam() function, 497, 500

Adobe SVG Viewer, 532, 541

aggregation, 68

alert, placing in code, 428

alert dialog, 143

alert() method of window object, 143

algorithm, choosing right, 582–584

allowChars() method of TextUtil object, 350–351, 355

allowing valid characters in text box, 350–351

alternation, 207–208

Amazon.com Web search engine, 506

AND operator (&), 39–40

A9 (Amazon.com Web search engine), 506

appendChild() method

document object, 174–175

using, 361, 372

Apple

Macintosh

error reporting, 415–416

operating system, detecting, 247–248

Quicktime, 532, 542

Safari

detecting, 239–241

DOM and, 8, 182, 191

ECMAScript compliance, 6

error handling, 417–418

events and, 262

user-agent string, 227, 233

applet (Java)

description of, 531, 543

embedding, 543–544

JavaScript-to-Java communication, 546–548

Java-to-JavaScript communication, 548–551

referencing in JavaScript, 544–545

writing, 545–546

apply() method and object masquerading, 108–109

appVersion property of navigator object, 228

arguments object, 62–63

Array class

description of, 70

methods, 71–73

size of arrays, 71

sort() method, 367–369

stack and, 73–74

strings and, 70–71

array literals, 578, 592

arrow keys, using in numeric text box, 354–356

ASCII code, 197–198

ASP.NET and cookies, 486–487

assert() function, 431–432

assign() method of location object, 154

assigning

event handler, 267, 269–270

onload event handler, 288

assignment operators listed, 52–53

asterisk and forward slash (*) to end multiline comment, 12

asynchronous mode, loading file in, 447

asyncInvoke() method of SOAPCall object, 519

attachEvent() method, 267–269

attributes

Cascading Style Sheets (CSS), 303–304

description of, 68

<form/> element, 335

<input/> element, 336

<object/> element, 533–534

as properties in HTML DOM, 178–179

XHTML and, 348

autosuggest() method of TextUtil object, 363–365

autosuggest text box

description of, 362

matching and, 362–363

script for, 363–365

autosuggestMatch() method of TextUtil object, 362–363

avoiding

- hard-coded strings, 569–570
- string concatenation, 590
- with statement, 581

B

`back()` **method of window object, 148**

backreferences, 206–207

base class

- creating, 111–112
- description of, 103
- selecting, 104

BEA Systems, 605, 616

begin multiline comment(/*), 12

beginning comment, 12

behavior, 513

Big O notation, 582, 584

binding, 69, 101

Bitflux blog, 506

bitmap graphics, 133

bitwise operators

- AND (&), 39–40
- integers and, 37–39
- left shift, 41–42
- NOT (~), 39
- OR (|), 40
- signed right shift, 42
- unsigned right shift, 42–43
- XOR (^), 40–41

`block()` **method of TextUtil object, 349**

blocking

- invalid characters in text box, 349
- paste function in text box, 351–353

blur event, 287, 345

`blurAllow()` **method of TextUtil object, 354**

`blurBlock()` **method of TextUtil object, 353–354**

`<body/>` **tag, 128**

BOM. See Browser Object Model

BOM methods, 82

Boolean class, 27

Boolean operators

- logical AND (&&), 44–45
- logical NOT (!), 43–44
- logical OR (||), 45–46

`Boolean()` **type cast, 24**

Boolean value

- description of, 18
- as primitive type, 16
- replacing to optimize download time, 576–577
- `toString()` method of, 22

boundaries, 210–212

braces ({} for code block, 12

break statement, 56–57

breakpoint, setting

- Microsoft Script Debugger, 434
- Venkman, 439–440

browser. See also browser detection script;

specific browsers

- detection methods
 - cross-browser code and, 292
 - object/feature, 226
 - user-agent string, 226–233
- DOM and, 7, 8, 167
- ECMAScript and, 3–4, 5
- hiding code from, 129–130
- possessive quantifiers and, 204
- Text node and, 168

browser detection methods

- cross-browser code and, 292
- object/feature detection, 226
- user-agent string detection, 226–233

browser detection script

- first steps, 234–236
- Internet Explorer, detecting, 241–242
- Konqueror/Safari, detecting, 239–241
- login page and, 252–259
- methodology, 234
- Mozilla, detecting, 242–243
- Netscape Communicator 4.x, detecting, 243–244
- Opera, detecting, 237–239
- overview of, 234

Browser Object Model (BOM)

- description of, 9, 136
- document object, 149–153
- history, 148–149
- intervals and timeouts, 146–148
- location object, 153–155
- manipulating windows, 139–140
- navigating and opening new windows, 140–141
- navigator object, 155–156
- onerror event handler, 419–423
- Same Origin Policy and, 564
- screen object, 156–157
- status bar, 144–145
- system dialogs, 142–144
- window object, 136–139

built-in methods, using, 590**built-in objects**

- description of, 81
- Global object, 81–84

byte, 574

C

caching and inline code, 128

`call()` **method and object masquerading, 107**

Call Stack view (Venkman), 441

Call Stack window (Microsoft Script Debugger), 433

callback function, 502–503

caller **property of function, 300**

calling function, 60

`callService()` **method of WebService component, 514**

`callWebService()` **function**

cross-browser example, 529

Internet Explorer example, 516

Mozilla example, 521

WebServiceProxyFactory object, 524–525

Camel Notation, 13

caret (^)

in regular expressions, 199

as XOR operator, 40–41

Cascading Style Sheets (CSS)

description of, 303–304

style, accessing, 309–312

case-sensitivity, 11

CDATA sections, 131–132, 162

`ceil()` **method of Math object, 85**

CEnv, 2

change event

HTML, 286

text box and, 345, 354

character class

combination, 200

description of, 199

negation, 199

predefined, 200–201

range, 199–200

simple, 199

character code and event, 277

character literals and string type, 21

characters

allowing valid in text box, 350–351

blocking invalid in text box, 349

limiting in text box, 347–348

`charAt()` **method of String class, 29**

`CharCode` **property of event object, 296**

`charCodeAt()` **method of String class, 29**

`checkForErrors()` **function, 461–462**

child node, 164

class. See also object; reference types; specific classes

base

creating, 111–112

description of, 103

selecting, 104

character

combination, 200

description of, 199

negation, 199

predefined, 200–201

range, 199–200

simple, 199

combination, 200

description of, 67

ECMAScript for XML, 607–615

clearing breakpoint, 440

`clearInterval()` **method of window object, 147**

`clearTimeout()` **method of window object, 146**

click event, 280

client-server communication

cookies

ASP.NET and, 486–487

description of, 481

ingredients of, 482

in JavaScript, 483–484

JSP and, 485–486

passing between client and server, 488–489

PHP and, 487–488

restrictions on, 482

hidden frame method

description of, 490–491

iframes, 491–492

HTTP requests

copycat implementations, 496

description of, 493–495

GET request, performing, 496–497

header, using, 495–496

POST request, performing, 497–498

intelligent HTTP requests

description of, 502

`get()` method, 502–505

`post()` method, 505–506

LiveConnect requests

description of, 498

POST, performing, 500–502

plugins and, 532

practical applications, 506

client-side scripting language, history of, 2

`clientX` and `clientY` **properties of event object, 275**

`cloneContents()` **method, 325**

`cloneRange()` **method, 328**

cloning range

DOM, 328

Internet Explorer, 332

- close() **method of document object, 152–153**
- closure, 65–66**
- C-minus-minus (Cmm), 1–2**
- code block and braces ({}), 12**
- code injection, 83**
- code maintenance and inline code, 128**
- Codebase Principals, 568**
- collapse() **method (Internet Explorer), 331**
- collapsible section, 308–309**
- collapsing range**
 - DOM, 327
 - Internet Explorer, 331
- combination class, 200**
- combo box**
 - adding options, 359–360
 - moving options, 361
 - options, 357
 - removing options, 360–361
 - reordering options, 361–362
 - retrieving or changing options, 357–359
 - <select/> element and, 356–357
- comma operator (,), 53**
- Command Window window (Microsoft Script Debugger), 434**
- commands**
 - Profile→Save Profile Data (Venkman), 442
 - Tools→JavaConsole, 429
 - Tools→Web Development→JavaScript Debugger (Mozilla), 435
 - View→Script Debugger (Internet Explorer), 432–433
- comments**
 - overview of, 12
 - removing to optimize download time, 574
- communication. See client-server communication**
- compareBoundaryPoints() **method, 328**
- compareEndPoints() **method (Internet Explorer), 332**
- compareVersions() **method, 235–236**
- comparing**
 - ranges
 - DOM, 328
 - Internet Explorer, 332
 - strings, 49–50
- comparison function**
 - multi-column table, sorting, 374–375
 - one-column table, sorting, 371
 - overview of, 367–369
- complex patterns**
 - alternation, 207–208
 - backreferences, 206–207
 - boundaries, 210–212
 - grouping, 205–206
 - lookaheads, 210
 - multiline mode, 212–213
 - non-capturing groups, 209
- complex selection**
 - DOM ranges, 322–323
 - Internet Explorer, 330
- compound assignment operators, 52–53**
- computed style of element**
 - description of, 312
 - DOM and, 313–314
 - Internet Explorer and, 312–313
- concat() **method**
 - Array class, 72–73
 - String class, 30
- conditional operators, 52**
- confirm dialog, 143–144**
- confirm() **method of window object, 143–144**
- conformance with DOM, detecting, 189–191**
- constructor, 68**
- constructor paradigm, 92–93**
- constructor **property of Object class, 26**
- content types, 533**
- continue **statement, 56–57**
- conversion function, creating, 378–379**
- conversions**
 - to number, 23–24
 - overview of, 21
 - to string, 22–23
 - string to array, 72
 - type casting, 24–25
- cookies**
 - description of, 481
 - ingredients of, 482
 - in JavaScript, 483–484
 - passing between client and server, 488–489
 - restrictions on, 482
 - on server
 - ASP.NET and, 486–487
 - JSP and, 485–486
 - PHP and, 487–488
- createDocument() **method (DOM), 450–451**
- createDocumentFragment() **method of document object, 177–178**
- createElement() **method of document object, 174–175**
- createNodeIterator() **method of document object, 184–186**
- createProxy() **method of WebServiceProxy-Factory object, 524**
- createProxyAsync() **method of WebServiceProxyFactory object, 522–523**
- createRange() **method, 317**

createTextNode() method of document object

`createTextNode()` **method of document object, 174–175**

`createXMLDOM()` **function, 446–447**

credit card, validating, 218–221

Crockford, Douglas (software engineer), 432

cross-browser approach

to detection and plugins, 542–543

to Web services

overview of, 525–527

TemperatureService object, 527–529

cross-browser events

adding or removing event handler, 292–294

description of, 292

EventUtil object, 292

CSS (Cascading Style Sheets)

description of, 303–304

style, accessing, 309–312

`cssText` **property of style object, 305**

currentStyle object (Internet Explorer), 312–313

cursor trail script, 399–400

D

dash (-) in regular expression, 199–200

data types

add operator and, 48

sorting table with different, 377–381

`dataTransfer` **object (Internet Explorer)**

description of, 393–394

methods, 394–395

properties, 395–397

date, validating, 216–218

`Date` **class**

description of, 77–78

methods, 78–81

`dblclick` **event, 280**

Debug toolbar

Microsoft Script Debugger, 434

Venkman, 440–441

debuggers

Microsoft Script Debugger, 432–435

overview of, 432

Venkman, 435–443

debugging techniques. See also debuggers

alerts, 428

Java console, 429–430

JavaScript Verifier, 432

overview of, 428

throwing own error, 431–432

declaring

object, 68

variable, 14

`decodeURI()` **method of Global object, 82**

`decodeURIComponent()` **method of Global object, 82**

default behavior for event, preventing, 278

`defaultStatus()` **method of window object, 145–146**

`defineGetter()` **method of XMLSerializer**

object, 453

defining

class or object

constructor paradigm, 92–93

dynamic prototype method, 95–96

example of, 97–98

factory paradigm, 90–92

hybrid constructor/prototype paradigm, 94–95

hybrid factory paradigm, 96

paradigm, choosing, 97

prototype paradigm, 93–94

multiple variables, 591

delete operator, 33

`deleteContents()` **method, 324**

`deleteCookie()` **function, 484**

deployment issues

intellectual property

Microsoft Script Encoder, 594–595

obfuscating, 593–594

overview of, 593

internationalization

detecting language, 568–569

overview of, 568

strategies for, 569–570

string considerations, 570–573

optimization

of download time, 573–578

of execution time, 578–593

overview of, 573

overview of, 563

security

Mozilla-specific issues, 566–568

resource limitations, 568

Same Origin Policy, 563–564

window object issues, 564–566

depth-first search, 182

dereferencing object, 69

`detach()` **method, 329**

`detachEvent()` **method, 267–269**

`detectFlash()` **function, 542–543**

`detectFlashInIE()` **function, 542**

detecting. See also platform/operating system

detection script; user-agent string detection

conformance with DOM, 189–191

Internet Explorer, 241–242

- Konqueror/Safari, 239–241
- language, 569
- Macintosh operating system, 247–248
- Mozilla, 242–243
- Netscape Communicator 4.x, 243–244
- Opera, 237–239
- plugins
 - ActiveX, 540–542
 - cross-browser detection, 542–543
 - Netscape-style, 535–540
 - overview of, 535
- Shift, Alt, or Ctrl key presses, 275
- Unix operating system, 248–249
- Windows operating system, 245–247
- DHTML (Dynamic HTML)**
 - development of, 7
 - drag and drop functionality and, 387
- direct assignment of event handler, 269–270**
- disabling**
 - Mozilla privileges, 567
 - Submit button, 341–342
- Dithered JavaScript compression utility, 593**
- `divide()` **function, 431–432**
- divide operator (/), 47**
- `doAdd()` **function, 63–65**
- document element, 162**
- document object (BOM)**
 - accessing relative nodes, 167–168
 - `appendChild()` method, 174–175
 - attributes, 169–170
 - collections, 150–151
 - cookie property, 483
 - `createDocumentFragment()` method, 177–178
 - `createElement()` method, 174–175
 - `createNodeIterator()` method, 184–186
 - `createTextNode()` method, 174–175
 - creating new nodes, 173–174
 - description of, 149
 - `getElementById()` method, 172–173
 - `getElementsByName()` method, 171–172
 - `getElementsByTagName()` method, 171
 - `insertBefore()` method, 175–177
 - methods, 151–153
 - `nodeType` property, 169
 - properties of, 149–150
 - `removeChild()` method, 175–177
 - `replaceChild()` method, 175–177
- Document Object Model (DOM)**
 - browser support for, 167
 - computed style of element and, 313–314
 - description of, 6, 159
 - event handler methods, 268–269
 - event model and, 265–266
 - event object and, 270, 272–274
 - events and, 261–262
 - hierarchy of nodes, 163–166
 - HTML features
 - attributes as properties, 178–179
 - table methods, 179–182
 - implementation object, 189–191
 - language-specific, 166–167
 - Level 2
 - `NodeIterator` object, 182–187
 - `TreeWalker` object, 187–189
 - Level 3, 191
 - levels of, 7–8
 - manipulation of, 174
 - purpose of, 7
 - range
 - clean up, 329
 - cloning, 328
 - collapsing, 326–327
 - comparing, 328
 - complex selection, 322–323
 - inserting with range content, 325–326
 - interacting with range content, 324–325
 - overview of, 317–318
 - simple selection, 318–322
 - Same Origin Policy and, 564
 - using sparingly to optimize execution time, 592–593
 - Web browser and, 8
 - XML and
 - Internet Explorer and, 445–450
 - Mozilla and, 450–455
 - overview of, 163
- dollar sign (\$) and variable name, 13**
- DOM. See Document Object Model**
- domain and cookie, 482**
- `DOMParser` **object (Mozilla), 451–452**
- double escaping (\\), 197, 198**
- double quotation marks (“)**
 - internationalization and, 572–573
 - string literal and, 21
- do-while statement, 54**
- download time, optimizing**
 - byte count, decreasing, 576–577
 - comments, removing, 574
 - ECMAScript Cruncher tool and, 575–576
 - line breaks, removing, 575
 - overview of, 573–574
 - variable names, replacing, 575
- drag and drop functionality**
 - description of, 387
 - simulated type
 - advantages and disadvantages of, 405
 - code for, 400–402

drag and drop functionality (continued)

drag and drop functionality (continued)

- description of, 399–400
- drop target, creating, 403–405
- system type
 - advantages and disadvantages of, 399
 - dataTransfer object and, 393–397
 - description of, 387–388
 - dragDrop() method, 397–399
 - dragged item events, 388–389
 - drop target, creating, 392–393
 - drop target events, 389–390
 - events, using all, 390–392
- zDragDrop library
 - creating draggable element, 406
 - description of, 405
 - drop target, creating, 406
 - example, 408–409
- drag event, **388**
- dragDrop() method (Internet Explorer), **397–399**
- dragend event, **388**
- dragenter event, **389–390**
- dragleave event, **389–390**
- dragover event, **389–390**
- dragstart event, **388**
- drop target, creating**
 - simulated drag and drop, 403–405
 - system drag and drop, 392–393
 - zDragDrop library, 406
- dropEffect property of dataTransfer object, **395–397**
- Duff's Device, 586–588**
- duplicate() method (Internet Explorer), **332**
- Dynamic HTML (DHTML)**
 - development of, 7
 - drag and drop functionality and, 387
- dynamic prototype method**
 - description of, 95–96
 - inheritance and, 114–115
 - zInherit library and, 118–119

E

- early binding, 69**
- ECMA (European Computer Manufacturers Association), 2**
- ECMAScript. See also ECMAScript for XML; ECMAScript 4**
 - conformance with, 4
 - conversions
 - to number, 23–24
 - overview of, 21

- to string, 22–23
- type casting, 24–25
- description of, 3
- editions of, 4
- error handling and, 411
- functions
 - arguments object, 61–62
 - closures, 65–66
 - description of, 59–61
 - Function class, 62–65
 - overloading of, 61
- keywords, 15
- Language Specification, downloading, 12
- operators
 - additive, 47–49
 - assignment, 52–53
 - bitwise, 36–43
 - Boolean, 43–46
 - comma, 53
 - conditional, 52
 - description of, 32
 - equality, 50–52
 - multiplicative, 46–47
 - relational, 49–50
 - Unary, 33–36
- origins of, 2
- primitive and reference values, 15–16
- primitive types
 - Boolean, 18
 - description of, 16
 - Null, 18
 - Number, 18–20
 - String, 20–21
 - typeof operator, 16–17
 - Undefined, 17–18
- reference types
 - Boolean class, 27
 - description of, 25–26
 - instanceof operator, 32–33
 - Number class, 27–29
 - Object class, 26
 - String class, 29–32
- reserved words, 15
- statements
 - break and continue statements, 55–56
 - if statement, 53–54
 - iterative, 54–55
 - labeled, 55
 - with statement, 57
 - switch statement, 58–59
- syntax, 11–12

- try...catch statement, 423–427
- variables, 12–14
- Web browsers and, 5
- ECMAScript Cruncher tool, 575–576**
- ECMAScript for XML (E4X)**
 - classes, 607–615
 - for each...in loop, 607
 - implementation, 616
 - overview of, 605–607
- ECMAScript 4**
 - implementations of
 - Microsoft, 605
 - Mozilla, 604–605
 - Netscape proposal for
 - classes, 601–603
 - functions, 599–600
 - inheritance, 603
 - keywords and reserved words, 598–599
 - numeric literals, 600
 - overview of, 598
 - types, 601
 - variables, 599
 - overview of, 597
- ECMA-262 standard, 4**
- effectAllowed **property of dataTransfer object, 395–397**
- E4X. See ECMAScript for XML**
- Eich, Brendan (developer), 2**
- Element **node methods, 169–170**
- e-mail address, validating, 222**
- <embed/> **element, 534**
- embedding**
 - ActiveX control, 558
 - Flash movie, 552
 - Java applet, 543–544
 - plugin, 533–534
- enabledPlugin **object (Netscape), 536–537**
- enabling**
 - Microsoft Script Debugger, 435
 - Mozilla privileges, 566–567
- encapsulation, 68**
- encode() **method of SOAPCall object, 518–519**
- encodeURIComponent() **method of Global object, 81–82**
- encodeURIComponent() **method of Global object, 81–82**
- end multiline comment (/ *), 12**
- ending comment, 12**
- e-notation, 19**
- envelope, 510**
- equal operator (==)**
 - dates and, 380
 - description of, 50–51
- equality operators**
 - equal (==) and not equal (!=), 50–51
 - identically equal (===) and not identically equal (!==), 51–52
- error **event, 286**
- error handling**
 - ECMAScript and, 411
 - importance of, 411–412
 - onerror event handler, 419–423
 - Opera, 418–419
 - Safari, 417–418
 - syntax versus runtime errors, 412–413
 - try...catch statement, 423–427
 - using with both IE and Mozilla, 459–463
- error messages**
 - developer-thrown error, 428
 - “Identifier expected,” 15
 - JavaScript required, 258
 - line number given in, 415, 416
 - Mozilla, 416
 - object expected message, 431
 - “User did not grant privilege,” 568
 - for wrong browser or wrong operating system, 253, 257
- Error **object, 424–425, 426**
- error reporting**
 - Internet Explorer
 - Macintosh, 415–416
 - Windows, 413–415
 - Mozilla, 416
 - overview of, 413
- European Computer Manufacturers Association (ECMA), 2**
- eval() **method of Global object, 82–83**
- evaluate() **method of XPathEvaluator object, 467–471**
- event bubbling, 262–264, 278–279**
- event capturing, 264–265**
- event flow**
 - DOM and, 265–266
 - Internet Explorer and, 262–264
 - Netscape Navigator and, 264–265
- event handler**
 - adding or removing, 292–294
 - assigning, 266–267, 269–270
 - description of, 266
 - Internet Explorer and, 267–268
 - keyup, 363
 - onblur, 353–354, 506
 - onclick, 266
 - ondrag, ondragend, ondragstart, 388–389
 - onerror, 419–423
 - onkeydown, 354–355

event handler (continued)

- onkeypress, 348
- onkeyup, 363
- onload, 174, 287–288, 451
- onmouseover, onmouseout, 281
- onreadystatechange, 448
- onresult, 514–515
- onsubmit, 341
- onunload, 288
- this object and, 281
- using JavaScript as, 128–129

event object

- character code and, 277
- clientX and clientY properties, 275
- description of, 270
- detecting Shift, Alt, or Ctrl key presses, 275
- differences between IE and DOM versions of, 276–279
- DOM and, 270
- formatting, 294–299
- getting, 274, 299–300
- Internet Explorer and, 270
- keyCode property, 274
- preventing default behavior for, 278
- properties/methods
 - DOM, 272–274
 - Internet Explorer, 271–272
- screenX and screenY properties, 276
- similarities between IE and DOM versions of, 274–276
- target and, 277

eventPhase **property of event object, 296**

events. See also event flow; event handler;

specific events

- cross-browser
 - adding or removing event handler, 292–294
 - description of, 292
 - EventUtil object, 292
- description of, 261
- DOM and, 261–262
- drag-and-drop, using all, 390–392
- dragged item, 388–389
- drop target, 389–390 form field, 338–339
- types of
 - HTML, 286–291
 - keyboard, 284–286
 - mouse, 280–284
 - mutation, 291
 - overview of, 279
- zDragDrop library, 406–407

EventUtil object

- code example, 300–301
- description of, 292

exception

- description of, 412
- JavaScript-to-Java communication, 547–548

exec() **method of RegExp object, 194–195**

execution time, optimizing

- algorithm, choosing right, 582–584
- gotchas, avoiding, 590–591
- if statements, optimizing, 589
- loops
 - flipping, 584–585
 - reversing, 584
 - unrolling, 585–588
- minimizing statement count, 591–592

overview of, 578

scope awareness and, 579–581

switch versus if statements, 589

exp() **method of Math object, 85–86**

expand() **method (Internet Explorer), 330**

expando property on table, creating, 376–377

expiration of cookie, 482

eXtensible HTML (XHTML)

- attributes and, 348
- description of, 159–160
- example of, 161–162
- <script/> tag, 131–132

eXtensible Markup Language (XML). See also

ECMAScript for XML

API for, 162–163

DOM support for

- in Internet Explorer, 445–450

- Mozilla and, 450–455

HTTP request

- copycat implementations, 496

- description of, 493–495

- GET request, performing, 496–497

- header, using, 495–496

- POST request, performing, 497–498

using with both browsers

- code for, 463–465

- error handling, 459–463

- IE branch, 456

- modifying DOM creation, 455–456

- Mozilla branch, 456–459

eXtensible Stylesheet Language Transformations (XSLT)

overview of, 471–473

support for

- in Internet Explorer, 473–477

- in Mozilla, 477–479

external file

- format for, 126–127

- inline code versus, 127–128

extractContents() **method, 325**

F**factory paradigm, 90–92**`findPlugin()` **function (Netscape), 538**`findText()` **method (Internet Explorer), 329–330****Flash Detection Kit (Macromedia), 543****Flash movie**

description of, 551

embedding, 552

Flash-to-JavaScript communication, 555–557

JavaScript-to-Flash communication and, 553–555

referencing, 552

Flash Player (Macromedia), 532, 541**Flash-to-JavaScript communication, 555–557****flipping loop, 584–585****floating-point value**

creating, 19

sorting and, 379

`floor()` **method of Math object, 85, 87****foci, 103**`focus` **event, 287**`focus()` **method of form field, 338, 339–340**`for each...in loop (ECMAScript for XML), 607``for` **statement, 55**`for-in` **statement, 55****form**

input elements, 336–337

`<label/>` element, 337

list box or combo box

adding options, 359–360

moving options, 361

options, 357

removing options, 360–361

reordering options, 361–362

retrieving or changing options, 357–359

`<select/>` element and, 356–357

text box

allowing valid characters in, 350–351

autosuggest, 362–365

blocking invalid characters in, 349

blocking paste function, 351–353

description of, 342–343

limiting textarea characters, 347–348

numeric, with up and down arrow keys, 354–356

retrieving or changing value, 343–344

selecting text in, 344–346

tabbing forward automatically in, 346–347

validating `onblur`, 353–354`<form/>` **element**

accessing fields, 338

attributes of, 335

field properties, methods, and events, 338–339

first field, 339–340

getting form references, 337

resetting form, 342

Submit button, 340–342

format of WSDL file, 510–511`formatEvent()` **method of EventUtil object, 349****formatting event object, 294–299**`FormUtil` **object, 339**`forward()` **method of window object, 148****forward slash and asterisk (/*) to begin multiline comment, 12****forward slashes (//) to begin single-line comment, 12****Fox, Geoffrey (author), 578****frame. See hidden frame method**`fscommand()` **function (Flash), 556–558**`Function` **class, 63–65****functions**`addPostParam()`, 498`addURLParam()`, 497, 500`arguments` object, 62–63`assert()`, 431–432`callback`, 502–503`caller` property, 300`callWebService()`

cross-browser example, 529

Internet Explorer example, 516

Mozilla example, 521

`WebServiceProxyFactory` object, 524–525`checkForErrors()`, 461–462

checking for existence of, 226

closures, 65–66

comparison

multi-column table, sorting, 374–375

one-column table, sorting, 371

overview of, 367–369

conversion, creating, 378–379

`createXMLDOM()`, 446–447`deleteCookie()`, 484

description of, 59–61

`detectFlash()`, 542–543`detectFlashInIE()`, 542`divide()`, 431–432`doAdd()`, 63–65

ECMAScript

`arguments` object, 61–62

closures, 65–66

description of, 59–61

`Function` class, 62–65

overloading of, 61

ECMAScript 4, Netscape proposal for, 599–600

`findPlugin()` (Netscape), 538

functions (continued)

- `fscommand()` (Flash), 556–558
- `generateCompareTRs()`, 374–375, 379–380, 383–384
- `getCookie()`, 484
- `getServerInfo()`, 490, 491
- `getURL()` (Flash), 556
- `handleMouseDown()`, 401, 402, 404
- `handleMouseMove()`, 402, 404
- `handleResponse()`, 490–491
- `httpGet()`, 499, 504
- `httpPost()`, 500–502
- `initError()`, 461–462
- invalid value and, 411–412
- `isOverDropTarget()`, 403–405
- `onWebServiceResult()`
 - Internet Explorer example, 516
 - Mozilla example, 521
- overloading, 61
- paste
 - blocking in text box, 351–353
 - validating `onblur`, 353–354
- return, 60–61
- `setCookie()`, 483–484
- `sortTable()`
 - conversion function and, 380
 - descending order sort, 376–377
 - multi-column table, 375–376
 - one-column table, 371–373
- `XmlDom()`, 455–456

G

garbage collection routine, 69

Gecko style plugins, 535

`generateCompareTRs()` **function**, 374–375, 379–380, 383–384

`get()` **method for intelligent HTTP requests**, 502–505

GET request, performing

- HTTP, 496–497

- LiveConnect, 498–500

`getAllResponseHeaders()` **method**, 495

`getAttribute()` **method and sorting**, 382, 384

`getComputedStyle()` **method**, 313–314

`getCookie()` **function**, 484

`getData()` **method of dataTransfer object**, 394–395

`getElementById()` **method of document object**, 172–173

`getElementsByName()` **method of document object**, 171–172

`getElementsByTagName()` **method of document object**, 171

`getLeft()` **method of zDragDrop library**, 408

`getParameter()` **method of XSLTProcessor object**, 479

`getParameters()` **method of SOAPResponse object**, 520

`getPropertyPriority()` **method of style object**, 305

`getPropertyValue()` **method of style object**, 305

`getResponseHeader()` **method**, 495

`getSelectedIndexes()` **method of options collection**, 358–359

`getServerInfo()` **function**, 490, 491

`getTimezoneOffset()` **method of Date class**, 79

getting reference to <form/> element, 337

`getTop()` **method of zDragDrop library**, 408

`getURL()` **function (Flash)**, 556

`GetVariable()` **method (Flash)**, 553

Ghosh, Koushik (author), 582

Global object

- description of, 81

- methods, 81–84

- properties, 83–84

global variable, creating, 14

`go()` **method of window object**, 148

greater than operator (>), 49–50

greater than or equal to operator (>=), 49–50

greedy quantifier, 202–204

Greenburg, Jeff, Web site of, 586, 587–588

grouping, 205–206

H

`handleMouseDown()` **function**, 401, 402, 404

`handleMouseMove()` **function**, 402, 404

`handleResponse()` **function**, 490–491

hard-coded strings, avoiding, 569–570

`hasOwnProperty()` **method of Object class**, 26

`<head/>` **tag**, 128

header and HTTP requests, 495–496

heap, 15, 16

helper application, 531–532

helper object, 166

hexadecimal literal, 19

hidden frame method

- description of, 490–491

- iframes, 491–492

hiding code from browser, 129–130

history

- of browser window, 148
- of JavaScript, 1–2
- of Windows operating system, 246

history **property of window object, 148–149**

host object, 87

HotJava browser, 531

href **property of location object, 153**

HTML

- assigning event handler, 267
- DOM and
 - attributes as properties, 178–179
 - overview of, 159
 - table methods, 179–182
- external file format, 126–127
- <form/> element
 - accessing fields, 338
 - attributes of, 335
 - field properties, methods, and events, 338–339
 - first field, 339–340
 - getting form references, 337
 - resetting form, 342
 - Submit button, 340–342
- hiding code from browsers, 129–130
- iframes, 491–492
- inline code versus external files, 127–128
- <noscript/> tag, 130
- <script/> tag, 125–126
- <select/> element, 356–357
- style attribute, 304
- tag placement, 128–129

HTML component, 513

HTML event, 286–287

HTTP request

- copycat implementations, 496
- description of, 493–495
- GET request, performing, 496–497
- header, using, 495–496
- intelligent
 - get() method, 502–505
 - post() method, 505–506
- POST request, performing, 497–498
- Web services and, 509

httpGet() **function, 499, 504**

httpPost() **function, 500–502**

Hungarian Type Notation, 14

hybrid constructor/prototype paradigm, 94–95

hybrid factory paradigm, 96

hybrid method, 110–111

HyperText Applications (Microsoft), 568

I

identically equal operator (===), 51–52

“Identifier expected” error messages, 15

IE. See Internet Explorer

if statement

- description of, 53–54
- optimizing, 589

iframes, 491–492

image loading error, 421

implementation

- Browser Object Model (BOM), 9
- Document Object Model (DOM), 6–8
- ECMAScript, 3–5
- parts of, 3

implementation **object (DOM), 189–191**

indexOf() **method of String class, 30, 237**

infinity value, 20

information in WSDL file, locating, 511–513

inheritance

- in action, 103–104
- alternative paradigms
 - overview of, 115–116
 - xObjects, 120–124
 - zInherit, 116–120
- description of, 68, 103
- dynamic prototyping and, 114–115
- example of, 111–115
- implementing, 104–105
- methods of, 105–111

inheritFrom() **method of Object class, 116**

initError() **function, 461–462**

initializing variable, Java compared to JavaScript, 13

innerHTML **property (Internet Explorer), 315**

innerText **property (Internet Explorer), 314–315**

<input/> **element**

- description of, 336
- text attribute, 342–343

inRange() **method (Internet Explorer), 332**

insertBefore() **method of document object, 175–177**

insertBefore() **method of options collection, 362**

inserting

- item into array, 77
- iterative values, 591
- range content (DOM), 325–326

insertNode() **method, 325–326**

instance of class, 67

instance properties of RegExp object, 213–214

instanceOf() **method of Object class, 116–117**

instanceof **operator, 32–33**

instantiation, 67, 68

integer value

- bitwise operator and, 37–39
- creating, 18–19

intellectual property issues

- Microsoft Script Encoder, 594–595
- obfuscating, 593–594
- overview of, 593

intelligent HTTP request

- get() method, 502–505
- post() method, 505–506

IntelliSense, 69

interacting with range content

- DOM, 324–325
- Internet Explorer, 331

interface of object, 67

International Organization for Standardization and International Electrotechnical Commission (ISO/IEC), 2

internationalization concerns

- detecting language, 569
- overview of, 568
- strategies, 569–570
- string considerations
 - browser versus operating system support, 571
 - double quotes, using, 573
 - error-proofing strings, 571–573
 - representation and, 571
 - Unicode and, 570–571

Internet Explorer (IE, Microsoft)

- callWebService() function example, 516
- cloning ranges, 332
- collapse() method, 331
- compareEndpoints() method, 332
- complex selection, 330
- computed style of element and, 312–313
- currentStyle object, 312–313
- dataTransfer object
 - description of, 393–394
 - methods, 394–395
 - properties, 395–397
- detecting, 241–242
- developer-thrown error, 428
- DOM and, 8
- drag and drop functionality and, 387–388
- dragDrop() method, 397–399
- duplicate() method, 332
- ECMAScript compliance, 5–6
- Error object and, 425
- error reporting
 - Macintosh, 415–416
 - Windows, 413–415
- event bubbling, 262–264
- event handler, 267–269

- event object and, 270, 271–272
- events and, 261
- expand() method, 330
- findText() method, 329–330
- getElementById() method and, 173
- getElementsByName() method and, 172
- innerHTML property, 315
- innerText property, 314–315
- inRange() method, 332
- isEqual() method, 332
- loading XML, 447–448
- move() method, 330
- moveEnd() method, 330
- moveStart() method, 330
- moveToElementText() method, 330
- MSXML versions and, 446–447
- onWebServiceResult() function example, 516
- outerHTML property, 315–317
- outerText property, 315–317
- parentElement() method, 330
- parseError property (XML DOM), 449–450
- parsing errors, 449–450
- pasteHTML() method, 331
- range
 - cloning, 332
 - collapsing, 331
 - comparing, 331
 - complex selection, 330
 - interacting with range content, 330
 - overview of, 329
 - simple selection, 329–330
- readyState property (XML DOM), 448
- RegExp.multiline and, 215
- retrieving XML, 449
- selectNodes() method, 467
- selectSingleNode() method, 467
- setAttribute() and, 179
- stopping event bubbling, 278–279
- syntax error, 449–450
- Temperature Service (XMethods) example, 515–516
- testing for, 455–456
- transformNode() method (XSLT), 473–475
- user-agent string
 - description of, 227
 - version 3.0, 228–229
 - version 4.0, 229
 - version 5.0 and higher, 230
- View→Script Debugger, 432–433
- WebService component
 - example, 515–516
 - using, 513–515

XML DOM support in
 Mozilla and, 455–465
 overview of, 445–446
 xml property (XML DOM), 449
 XPath and, 467
 XSLT support in, 473–477

intervals, 146, 147

invalid value, 411–412

isChar **property of event object**, 296

isEqual() **method (Internet Explorer)**, 332

isFinite() **method**, 20

ISO/IEC (International Organization for Standardization and International Electrotechnical Commission), 2

isOverDropTarget() **function**, 403–405

isPrototypeOf() **method of Object class**, 26

item() **method of style object**, 305

iterative statements

do-while statement, 54
 flipping, 584–585
 for each...in (ECMAScript for XML), 607
 for statement, 55
 for-in statement, 55
 reversing, 584
 unrolling, 585–588
 while statement, 55

J

JAR (Java Archive) file, specifying, 544

Java

JavaScript compared to variables, 13
 switch statement, 59

Java applet

description of, 531, 543
 embedding, 543–544
 JavaScript-to-Java communication, 546–548
 Java-to-JavaScript communication, 548–551
 referencing in JavaScript, 544–545
 writing, 545–546

Java Archive (JAR) file, specifying, 544

Java console, 429–430

Java Development Kit (Sun Microsystems), 545

Java Plugin (Sun Microsystems), 532, 543

Java Server Pages (JSP) and cookies, 485–486

JavaScript console

Mozilla, 416
 Opera, 418–419, 430

“JavaScript Performance Issues” seminar (Geoffrey Fox), 578

JavaScript Verifier, 432

JavaScript-to-Flash communication, 553–555

JavaScript-to-Java communication, 546–548

Java-to-JavaScript communication, 548–551

join() **method of Array class**, 72

.js **extension**, 126

JScript (Microsoft), 2, 5

JScript.NET (Microsoft), 605

JSObject **class**, 548–551

JSP (Java Server Pages) and cookies, 485–486

K

keyboard event

description of, 284
 properties, 285
 sequencing, 286

keyCode **property of event object**, 274

keypress **event**, 348, 355

keyup **event handler**, 363

keywords

description of, 15
 new keyword, 68
 this keyword
 ActiveX objects and, 448
 description of, 89–90
 object masquerading and, 105–107
 style object and, 305

KHTML, 233, 239

Konqueror

detecting, 239–241
 DOM and, 8, 182, 191
 ECMAScript compliance, 6
 Safari and, 233

L

<label/> **element**, 337

labeled statement, 56, 57

language

detecting, 569
 object-oriented
 inheritance and, 103
 requirements of, 68

language **attribute of <script/> tag**, 125–126

language **property of navigator object**, 569

lastIndexOf() **method of String class**, 30

last-in-first-out (LIFO) **structure**, 73

last-in-last-out (LIFO) **structure**, 75

late binding, 69

left shift operator (<<), 41–42

length **property**

Array class, 71
 Function class, 64
 window object, 149

less than operator (<), 49–50

less than or equal to operator (<=), 49–50

LIFO (last-in-first-out) structure, 73

LILO (last-in-last-out) structure, 75

limiting textarea characters in text box, 347–348

line breaks, removing to optimize download time, 575

linear algorithm, 583

list box

adding options, 359–360

moving options, 361

options, 357

removing options, 360–361

reordering options, 361–362

retrieving or changing options, 357–359

<select/> element and, 356–357

literals

array, 578, 592

hexadecimal, 19

numeric, 600

object, 578, 592

octal, 19

string, 21

LiveConnect

debugging and, 429–430

Java-to-JavaScript communication, 548–551

request

description of, 498

GET, performing, 498–500

POST, performing, 500–502

LiveScript, 2

load event, 286, 287–288

load() method (XML DOM)

Internet Explorer and, 447–448

Mozilla and, 451–453

updating for use with IE and Mozilla, 458–459

Loaded Scripts pane (Venkman), 437–439

loadXML() method (XML DOM)

Internet Explorer and, 447

Mozilla and, 452–453

updating for use with IE and Mozilla, 458

localeCompare() method of String class, 30–31, 368

location object (BOM), 153–155

log() method of Math object, 85–86

logging message to Java console, 429–430

logical AND operator (&&), 44–45

logical NOT operator (!), 43–44

logical OR operator (||), 45–46

login page example, 252–259

lookaheads, 210

loop statements

do-while statement, 54

flipping, 584–585

for each...in (ECMAScript for XML), 607

for statement, 55

for-in statement, 55

reversing, 584

unrolling, 585–588

while statement, 55

loosely typed variable, 11, 13

Luhn algorithm, 218–221

M

Macintosh (Apple)

error reporting, 415–416

operating system, detecting, 247–248

Macromedia

Flash Detection Kit, 543

Flash movies

description of, 551

embedding, 552

Flash-to-JavaScript communication, 555–557

JavaScript-to-Flash communication, 553–555

referencing, 552

Flash Player, 532, 541

Shockwave, 533

match() method of String object, 195

Math object

methods, 84–87

properties, 84

Mathematical Markup Language (MathML), 8

max() method of Math object, 84–85

memory and dereferencing object, 69

message property of Error object, 425

metacharacters for regular expression, 197

methods. See also specific methods

Array class, 71–77

BOM, 82

built-in, using, 590

compareVersions(), 235–236

Cookie object (JSP), 485

creating new, 99–100

dataTransfer object (Internet Explorer), 394–395

description of, 68

document object

creating new nodes, 173–174

overview of, 151–153

Element node, 169–170

event handler, 267–269

event object

DOM, 272–274

Internet Explorer, 271–272

overview of, 294–295

EventUtil object, 292–294

Flash movies, 553, 554

form field, 338–339

Function class, 64–65

- getAllResponseHeaders(), 495
 - getComputedStyle(), 313–314
 - getResponseHeader(), 495
 - GetVariable() (Flash), 553
 - Global object, 81–83
 - isFinite(), 20
 - JSONObject class, 549
 - location object, 154–155
 - Math object, 84–87
 - navigator object, 155–156
 - Node interface, 165–166
 - Number class, 28
 - Object class, 26
 - parseFloat(), 23, 235
 - parseInt(), 23
 - range, 321–323
 - redefining existing, 100–101
 - refresh() (Netscape), 540
 - RegExp object, 194–195
 - setRequestHeader(), 495
 - SetVariable() (Flash), 553
 - String class, 29–32
 - style object, 305–307
 - table (DOM HTML), 179–182
 - toString()
 - Array class, 71–72
 - Date class, 78–79
 - description of, 22
 - Function class, 64–65
 - location object, 154–155
 - Object class, 26
 - TotalFrames() (Flash), 553
 - TreeWalker object (DOM), 187
 - WebService object, 526
 - window object (BOM), 139–148
 - window.close(), 142
 - window.open(), 140–142, 565
 - XML class (ECMAScript for XML), 611–614
 - XMLList class (ECMAScript for XML), 615
 - zDragDrop library, 408–409
- Microsoft. See also Internet Explorer**
- ECMAScript 4 implementation, 605
 - HyperText Applications, 568
 - JScript, 2, 5
 - JScript.NET, 605
 - OLE/COM Object Viewer, 540, 541
 - Script Debugger, 432–435
 - Script Encoder, 594–595
 - WebService component, 513
 - Windows
 - Media Player, 532, 542
 - operating system, detecting, 245–247
 - XP Service Pack 2, 516, 558, 565
- MIME types**
- description of, 533
 - invisible, 537
 - Java applet and, 543
 - Netscape-style plugins and, 535–539
- min() method of Math object, 84–85**
- minus operator (-), 36–37**
- modal dialog, 144**
- modifying**
- CSS rule, 311–312
 - object
 - creating new method, 99–100
 - redefining existing method, 100–101
 - text between starting and ending tags, 314–315
- modulus (remainder) operator (%), 47**
- mouse event**
- description of, 280–281
 - properties, 281–283
 - sequencing, 284
- move() method (Internet Explorer), 330**
- moveBy() method of window object, 139**
- moveEnd() method (Internet Explorer), 330**
- moveStart() method (Internet Explorer), 330**
- moveTo() method**
- window object, 139
 - zDragDrop library, 408
- moveToElementText() method (Internet Explorer), 330**
- moving**
- options between list or combo boxes, 361
 - window, 139–140
- Mozilla. See also Venkman**
- callWebService() function example, 521
 - createDocument() method and, 450–451
 - detecting, 242–243
 - developer-thrown error, 428
 - DOM and, 182, 191
 - DOM Range implementation and, 323
 - DOMParser object, 451–452
 - ECMAScript for XML and, 616
 - ECMAScript 4 implementation, 604–605
 - error handling, 459–463
 - error messages, 416
 - Error object and, 425
 - error reporting, 416
 - event bubbling and, 264
 - events and, 262
 - Gecko style plugins, 535
 - getComputedStyle() method and, 314
 - keypress event and, 348
 - loading XML, 451–453
 - onWebServiceResult() function example, 521

Mozilla (continued)

- parseError property (XML DOM), creating for, 459–463
- parsing error, 454–455
- privileges, 566–567
- readyState property (XML DOM), updating for use with, 457–459
- retrieving XML, 453–454
- security mechanisms, 566–568
- selectNodes() method, creating for, 468–469
- selectSingleNode() method, creating for, 469–470
- SignTool, 567–568
- Simple Object Access Protocol (SOAP) and, 518–521
- SOAPCall object, 518
- SOAPParameter constructor, 518
- SOAPResponse object, 519–520
- syntax error, 454–455
- Temperature Service example, 520–521
- testing for, 455–456
- Tools→Web Development→JavaScript Debugger, 435
- transformNode() method (XSLT), creating for, 478
- Universal Browser Read privilege, 517–518, 567
- Universal Browser Write privilege, 567
- user-agent string, 227, 230–231
- Web services and
 - enhanced privileges, 517–518
 - overview of, 516
 - SOAP methods, using, 518–521
 - WSDL proxies, using, 522–525
- Web site, 563
- WebServiceProxyFactory object, 522
- XML DOM support in
 - Internet Explorer and, 455–465
 - overview of, 450
- xml property (XML DOM), 453–454
- XMLSerializer object, 453, 454
- XPath and, 467–471
- XPathEvaluator object, 467–468
- XPathResult object, 468–471
- XSLT and, 477–479
- XSLTProcessor object, 477–479

Mozilla Foundation, 598

MSXML

- overview of, 445–446
- versions of, 446–447
- Windows and, 450
- XSLT support and, 473, 477

multiline comment, beginning and ending, 12

multiline mode, 212–213

multiple inheritance

- description of, 106
- zInherit library and, 119–120

multiplicative operators

- divide (/), 47
- modulus (remainder, %), 47
- multiply (*), 46

multiply operator (*), 46

mutation event, 291

N

name of cookie, 482

named property lookup, 583

Namespace class (ECMAScript for XML), 607–608

naming variable, 13–14

NaN value, 20

native object

- Array class, 70–77
- Date class, 77–81
- description of, 70

Navigator (Netscape)

- DOM and, 8
- ECMAScript compliance and, 5
- event capturing, 264–265
- user-agent string and, 227–229

navigator object (BOM)

- appVersion property, 228
- description of, 155–156
- language property, 569
- platform property, 245
- as standard, 225
- userAgent property, 226–227

navigator.plugins collection, 538–540

negation class, 199

negative lookahead, 210

negative test, shortening, 577

nested try...catch statement, 424

Netscape

- client-side scripting language and, 2
- Communicator 4.0, 229
- Communicator 4.x, detecting, 243–244
- DevEdge Web site, 120
- disbanding of, 598
- ECMAScript 4 proposal
 - classes, 601–603
 - functions, 599–600
 - inheritance, 603
 - keywords and reserved words, 598–599
 - numeric literals, 600
 - overview of, 598
 - types, 601
 - variables, 599
- <embed/> element and, 534
- enabledPlugin object, 536–537

findPlugin() function, 538
 helper applications, 531–532
 Navigator
 DOM and, 8
 ECMAScript compliance and, 5
 event capturing, 264–265
 user-agent string and, 227–229
 plugins, 535–540
 refresh() method, 540
 netscape.javascript **package**, 548
Netscape-style plugins, 535–540
 new **keyword**, 68
 NodeFilter **object (DOM)**, 186
 NodeIterator **object (DOM)**, 182–187
 nodeKind() **method of XML class (ECMAScript for XML)**, 613
nodes
 events and, 266
 hierarchy of, 163–166
 nodeType **property of document object**, 169
no-drop symbol, 388
Nombas, 1–2
non-capturing group, 209
 <noscript/> **tag, 130, 255–256**
not equal operator (!=), 50–51
not identically equal operator (!==), 52
NOT operator (~), 39
Null value
 description of, 18
 as primitive type, 16
 Number **class, 27–29**
 Number() **type cast, 24–25**
Number value
 converting to, 23–24
 description of, 18–20
 as primitive type, 16
 toString() method of, 22
 Number.MAX_VALUE, **20**
 Number.MIN_VALUE, **20**
 Number.NEGATIVE_INFINITY, **20**
 Number.POSITIVE_INFINITY, **20**
numeric text box, 354–356
 numericScroll() **method of TextUtil object, 355–356**

O

obfuscating, 593–594
object. See also class; specific objects
 composition of, 68
 declaring and instantiation, 68
 defining
 constructor paradigm, 92–93
 dynamic prototype method, 95–96
 example of, 97–98
 factory paradigm, 90–92
 hybrid constructor/prototype paradigm, 94–95
 hybrid factory paradigm, 96
 paradigm, choosing, 97
 prototype paradigm, 93–94
 dereferencing, 69
 description of, 67
 early versus late binding, 69
 helper, 166
 modifying
 method, creating new, 99–100
 method, redefining existing, 100–101
 very late binding, 101
 types of
 built-in, 80–87
 host, 87
 native, 70–80
Object class, 26
object definition, 67
 <object/> **element**
 attributes, 533–534
 plugin and, 531
 object expected **message, 431**
object literal, 578, 592
object masquerading
 apply() method and, 108–109
 call() method and, 107
 overview of, 105–107
object/feature detection, 226
object-oriented language
 inheritance and, 103
 requirements of, 68
octal literal, 19
OLE/COM Object Viewer (Microsoft), 540, 541
 onblur **event handler, 353–354, 506**
 onclick **event handler, 266**
 ondrag **event handler, 388–389**
 ondragend **event handler, 388–389**
 ondragstart **event handler, 388**
 onerror **event handler**
 description of, 419–420
 extracting error information, 420
 image loading errors, 421
 syntax errors, 422
 onError() **method of WebServiceProxyFactory object, 522–523**
 onkeydown **event handler, 354–355**
 onkeypress **event handler, 348**
 onkeyup **event handler, 363**
 onload **event handler, 174, 287–288, 451**

onLoad () method of WebServiceProxyFactory object

`onLoad()` **method of** `WebServiceProxyFactory` **object**, 522, 523

`onmouseover` and `onmouseout` **event handlers**, 281

`onreadystatechange` **event handler**, 448

`onresult` **event handler**, 514–515

`onsubmit` **event handler**, 341

`onunload` **event handler**, 288

`onWebServiceResult()` **function**

Internet Explorer example, 516

Mozilla example, 521

`open()` **method**

document object, 152–153

window object, 565

opening new window, 140–143

Opera

detecting, 237–239

detecting key presses and, 282

DOM and, 8, 191

ECMAScript compliance, 6

error handling, 418–419

events and, 262

`getComputedStyle()` method and, 314

`getElementsByName()` method and, 172

JavaScript console and, 418–419, 430

`postError()` method, 430

`RegExp.multiline` and, 215

regular expressions and, 238

style sheet access and, 310

user-agent string, 227, 232–233

operating system

drag and drop functionality and, 387–388

support for and internationalization, 571

operating system detection script. *See* **platform/operating system detection script**

operation in WSDL, 510

operators

additive

add (+), 47–48

subtract (-), 48–49

assignment, 52–53

bitwise

AND (&), 39–40

integers and, 37–39

left shift (<<), 41–42

NOT (~), 39

OR (|), 40

signed right shift (>), 42

unsigned right shift (>>), 42–43

XOR (^), 40–41

Boolean

description of, 43

logical AND (&&), 44–45

logical NOT (!), 43–44

logical OR (||), 45–46

conditional, 52

equality, 50–52

`instanceof` operator, 32–33

multiplicative

divide (/), 47

modulus (remainder, %), 47

multiply (*), 46

relational, 49–50

`typeof` operator, 16–17

Unary

delete operator, 33

plus and minus, 36–37

postfix increment/decrement, 35–36

prefix increment/decrement, 34–35

void operator, 33–34

optimization

download time and

array and object literals and, 578

byte count and, 576–577

comments and, 574

ECMAScript Cruncher and, 575–576

line breaks and, 575

overview of, 573–574

tabs, spaces, and, 574

variable names and, 575

execution time and

basics, remembering, 582–589

DOM manipulation and, 592–593

gotchas, 590–591

overview of, 578

scope awareness and, 578–581

statement count and, 591–592

OR operator (|), 40

`outerHTML` **property (Internet Explorer)**, 315–317

`outerText` **property (Internet Explorer)**, 315–317

overloading function, 61

P

`<param/>` **element**, 534

parent node, 164

parent object (BOM), 137–138

`parentElement()` **method (Internet Explorer)**, 330

`parse()` **method of Date class**, 78

`parseError` **property (XML DOM)**

creating for Mozilla, 459–463

Internet Explorer, 449–450

`parseFloat()` **method**, 23, 235

`parseInt()` **method**, 23

parsing error

- description of, 412–413
- Internet Explorer, 449–450
- Mozilla, 454–455
- onerror event handler and, 422

Pascal Notation, 14**passing cookies between client and server, 488–489****paste function**

- blocking in text box, 351–353
- validating onblur, 353–354

pasteHTML() method (Internet Explorer), 331**path and cookies, 482****PHP and cookies, 487–488, 489****platform property of navigator object, 245****platform/operating system detection script**

- first steps, 245
- login page and, 252–259
- Macintosh OS, detecting, 247–248
- methodology, 244–245
- overview of, 244
- text of full script, 249–252
- Unix OS, detecting, 248–249
- Windows OS, detecting, 245–247

plugins. See also Flash movie; Java applet

- ActiveX controls, 558–560
- description of, 531
- detecting
 - ActiveX, 540–542
 - cross-browser detection, 542–543
 - Netscape-style, 535–540
 - overview of, 535
- embedding, 533–534
- MIME types and, 533
- popular, 532–533
- purpose of, 531–532

plus operator (+), 36**polymorphism, 68****pop() method of Array class, 74–75****popping item from stack, 73–74****pop-up blocker, 143, 565–566****positive lookahead, 210****possessive quantifier, 202–204****post() method for intelligent HTTP requests, 505–506****POST request, performing**

- HTTP, 497–498
- LiveConnect, 500–502

postError() method (Opera), 430**postfix decrement operator (--), 35–36****postfix increment operator (++), 35–36****pow() method of Math object, 85, 86****predefined class, 200–201****prefix decrement operator (--), 34–35****prefix increment operator (++), 34–35****prefixes for Hungarian Type Notation, 14****preventDefault() method of event object, 297****preventing default behavior for event, 278****primitive type and typeof operator, 16–17****primitive value, 15–16****private scope, 88****processing instruction, 162****Profile→Save Profile Data (Venkman), 442****profiling (Venkman), 442–443****prompt dialog, 144****prompt() method of window object, 144****properties**

- appVersion property of navigator object, 228
- attributes in HTML DOM elements as, 178–179
- caller property of function, 300
- charCode property of event object, 296
- constructor property of Object class, 26
- cookie property of document object, 483
- cssText property of style object, 305
- dataTransfer object, 395–397
- description of, 68
- document object, 149–150
- dropEffect property of dataTransfer object, 395–397
- effectAllowed property of dataTransfer object, 395–397
- event object
 - DOM, 272–274
 - Internet Explorer, 271–272
- eventPhase property of event object, 296
- expando, on table, creating, 376–377
- form field, 338–339
- Function class, 64
- Global object, 83
- history property of window object, 148–149
- href property of location object, 153
- HttpCookie (ASP.NET), 486–487
- innerHTML property (Internet Explorer), 315
- innerText property (Internet Explorer), 314–315
- isChar property of event object, 296
- keyboard events, 285
- keyCode property of event object, 274
- language property of navigator object, 569
- length
 - Array class, 71
 - Function class, 64
 - window object, 149
- Math object, 84
- message property of Error object, 425
- mouse events, 281–283
- navigator object, 155–156
- Node interface, 165–166
- nodeType property of document object, 169

properties (continued)

- Object class, 26
- options collection, 357
- outerHTML property (Internet Explorer), 315–317
- outerText property (Internet Explorer), 315–317
- parseError (XML DOM)
 - creating for Mozilla, 459–463
 - Internet Explorer, 449–450
- platform property of navigator object, 245
- prototype property of Object class, 26
- range, 318–320
- readyState (XML DOM)
 - Internet Explorer, 448
 - updating for use with IE and Mozilla, 457–459
- RegExp object
 - instance, 213–214
 - static, 214–215
- relatedTarget property of event object, 297
- removeProperty() method of style object, 305
- result property, 514–515
- screen object, 157
- selectorText property for CSS rule, 310
- setProperty() method of style object, 305
- SOAPResponse object (Mozilla), 519
- style object, 304
- style property for CSS rule, 310
- target property of event object, 298–299
- value property of text box, 343–344
- WebService object, 526
- xml (XML DOM)
 - Internet Explorer, 449
 - Mozilla, 453–454
- propertyIsEnumerable() **method of Object class, 26**
- protected scope, 88**
- protecting intellectual property**
 - Microsoft Script Encoder, 594–595
 - obfuscating, 593–594
 - overview of, 593
- prototype chaining, 109–110**
- prototype paradigm, 93–94**
- prototype **property of Object class, 26**
- pseudo-object, 22**
- public scope, 88**
- push() **method of Array class, 74–75, 76**

Q

QName class (ECMAScript for XML), 608

quantifiers

description of, 201

greedy, reluctant, and possessive, 202–204
simple, 201–202

queue and Array class, 75–76

Quicktime (Apple), 532, 542

quotation marks

double ("")

- internationalization and, 572–573
- string literal and, 21

single ('')

- internationalization and, 573
- string literal and, 21

R

raising exceptions, 426–427

random() **method of Math object, 87**

range

description of, 317

DOM and

- clean up, 329
- cloning range, 328
- collapsing range, 326–327
- comparing ranges, 328
- complex selection, 322–323
- inserting with range content, 325–326
- interacting with range content, 324–325
- overview of, 317–318
- simple selection, 318–322

Internet Explorer and

- cloning range, 332
- collapsing range, 331
- comparing ranges, 331
- complex selection, 330
- interacting with range content, 330
- overview of, 329
- simple selection, 329–330

practicality of, 332

range class, 199–200

raster graphics, 133

readyState **property (XML DOM)**

Internet Explorer, 448
updating for use with IE and Mozilla, 457–459

Real Player, 532, 541, 542

redefining existing method, 100–101

reference type

Boolean class, 27
description of, 25
instanceof operator, 32–33
Number class, 27–29
Object class, 26
String class, 29–32

reference value, 15–16

references to object, 69**referencing**

- Flash movie, 552
- Java applet, 544–545

`refresh()` **method (Netscape), 540**

RegExp object

- description of, 193–194
- instance properties, 213–214
- methods, 194–195
- static properties, 214–215

regular expression. See also `RegExp` **object**

character classes

- combination, 200
- description of, 199
- predefined, 200–201

- range, 199–200
- simple, 199

common patterns

- credit card, validating, 218–221
- date, validating, 216–218
- e-mail address, validating, 222

complex patterns

- alternation, 207–208
- backreferences, 206–207
- boundaries, 210–212
- grouping, 205–206
- lookaheads, 210
- multiline mode, 212–213
- non-capturing groups, 209

description of, 193

metacharacters and, 197

Opera and, 238

quantifiers

- description of, 201
- greedy, reluctant, and possessive, 202–204
- simple, 201–202

special characters and, 197–198

String methods and, 195–197

support for, 193–194

`relatedTarget` **property of event object, 297**

relational operators, 49–50

`reload()` **method of location object, 154**

reluctant quantifier, 202–204**Remember Me check box, 488–489**

`removeChild()` **method of document object, 175–177**

`removeEventHandler()` **method of EventUtil object, 293**

`removeEventListener()` **method (DOM), 268–269, 407**

`removeParameter()` **method of XSLTProcessor object, 479**

`removeProperty()` **method of style object, 305**

removing

- event handler, 292–294
- options from list or combo box, 360–361

renaming variable, 575**reordering options in list or combo box, 361–362****replace() method**

- location object, 154
- String object, 195–196

`replaceChild()` **method of document object, 175–177**

representation in Unicode, 571**reserved words, 15****reset event, 286****reset() method**

- HTML form, 342
- XSLT, 477

resetting form, 342**resize event, 287, 288–289**

`resizeBy()` **method of window object, 139**

`resizeTo()` **method of window object, 140**

resizing window, 139–140**resource limitations, 568****result property, 514–515****retrieving**

- list of registered plugins, 538–539
- text in text box, 343–344
- XML
 - Internet Explorer and, 449
 - Mozilla and, 453–454

return statement and function, 60–61**reverse() method**

- Array class, 76
- sorting and, 369

reversing loop, 584**rollover effect, applying, 304–305**

`round()` **method of Math object, 85**

rows collection, 371**running**

- Microsoft Script Debugger, 432–433
- Venkman, 435

Running Documents window (Microsoft Script Debugger), 433**runtime error, 412–413****S****Safari (Apple)**

- detecting, 239–241
- DOM and, 8, 182, 191
- ECMAScript compliance, 6
- error handling, 417–418
- events and, 262
- user-agent string, 227, 233

Same Origin Policy, 563–564

SAX (Simple API for XML), 162–163

Scalable Vector Graphics (SVG)

- description of, 8, 133
- example of, 133–134
- `<script/>` tag, 134–135
- tag placement, 135

scope

- description of, 88
- execution time and
 - local variables, using, 580–581
 - overview of, 579–580
 - with statement, avoiding, 581
- public, protected, and private, 88
- static, 88–89
- `this` keyword, 89–90

screen object (BOM), 156–157

screenX and screenY properties of event object, 276

Script Debugger (Microsoft), 432–435

Script Encoder (Microsoft), 594–595

`<script/>` tag

- HTML, 125–126
- SVG, 134–135
- XHTML, 131

ScriptEase, 2

scripting styles, 303–305

scroll event, 287, 289–291

scrolling message code, 146

scrollLeft and scrollTop properties of event object, 296

search() method of String object, 195

secure flag and cookie, 482

security

- inline code and, 127
- JavaScript-to-Java communication, 548
- Mozilla-specific issues, 566–568
- resource limitations, 568
- Same Origin Policy, 563–564
- window object issues, 564–566

`<select/>` element

- description of, 336
- list and combo boxes and, 356–357
- options collection, 357
- `selectedIndex` attribute, 357–358

select event

- HTML and, 286
- text box and, 345

select() method and text box, 344–345

selecting text in text box, 344–346

selectNode() method, 318–319

selectNodeContents() method, 318–319

selectNodes() method

- creating for Mozilla, 468–469
- Internet Explorer, 467
- `selectorText` property for CSS rule, 310

selectSingleNode() method

- creating for Mozilla, 469–470
- Internet Explorer, 467

semicolon (;)

- as optional at end-of-line, 12
- using, 574

send() method of WebService object, 526, 528–529

serializeToString() method of XMLSerializer object, 453, 454

setCookie() function, 483–484

setData() method of dataTransfer object, 394–395

setEnd() method, 322–323

setInterval() method of window object, 147

setParameter() method of XSLTProcessor object, 478–479

setProperty() method of style object, 305

setRequestHeader() method, 495

setStart() method, 322–323

setStartMode() method (XSLT), 477

setTimeout() method of window object, 146–148

SetVariable() method (Flash), 553

SGML (Standard Generalized Markup Language), 159–160

shift() method of Array class, 75, 76

shiftUp() method of options collection, 362

Shockwave (Macromedia), 533

sibling node, 164

signed integer, 37

signed right shift operator (>), 42

signed script, 567–568

signing applet, 548

SignTool (Mozilla), 567–568

Simple API for XML (SAX), 162–163

simple class, 199

Simple Object Access Protocol (SOAP)

- description of, 509–510
- Mozilla and, 518–521

simple quantifier, 201–202

simple selection

- DOM ranges, 318–322
- Internet Explorer, 329–330

simulated drag and drop

- advantages and disadvantages of, 405
- code for, 400–402
- description of, 399–400
- drop target, creating, 403–405

single quotation mark (')

- internationalization and, 573
- string literal and, 21

- size of array, 71**
- `slice()` **method**
 - Array class, 72–73
 - String class, 31–32
- SMIL (Synchronized Multimedia Integration Language), 8**
- `snapshot Length()` **and** `snapshotItem()` **methods**
 - of XPathResult object, 469**
- SOAP (Simple Object Access Protocol)**
 - description of, 509–510
 - Mozilla and, 518–521
- SOAPCall **object (Mozilla), 518**
- SOAPParameter **constructor (Mozilla), 518**
- SOAPResponse **object (Mozilla), 519–520**
- `sort()` **method of Array class, 76–77, 367–369**
- sorting**
 - advanced, 381–385
 - comparison function and, 367–369
 - multi-column table
 - comparison function, 374–375
 - overview of, 373–374
 - `sortTable()` function, 375–376
 - one-column table
 - comparison function, 371
 - overview of, 369–371
 - `sortTable()` function, 371–373
 - overview of, 367
 - `reverse()` method, 369
 - table
 - in descending order, 376–377
 - with different data types, 377–381
- `sortTable()` **function**
 - conversion function and, 380
 - descending order sort, 376–377
 - multi-column table, 375–376
 - one-column table, 371–373
- Source Code view (Venkman), 439**
- spaces, removing to optimize download time, 574**
- special characters**
 - regular expressions and, 197–198
 - XHTML and, 131–132
- specifying string literal, 21**
- `splice()` **method of Array class, 77**
- `split()` **method of String object, 72, 196**
- `sqrt()` **method of Math object, 85, 86**
- square brackets ([])**
 - Array class and, 71
 - in regular expressions, 199
 - XPath and, 466
- src attribute of <script/> tag, 126**
- stack**
 - description of, 15, 16
 - as last-in-first-out structure, 73
 - popping item from, 73–74
- Standard Generalized Markup Language (SGML), 159–160**
- statements**
 - break and continue, 56–57
 - description of, 53
 - if statement, 53–54, 589
 - iterative
 - do-while, 54
 - flipping, 584–585
 - for, 55
 - for-in, 55
 - reversing, 584
 - unrolling, 585–588
 - while, 55
 - labeled, 56, 57
 - minimizing count of to optimize execution time, 591–592
 - return, 60–61
 - switch, 58–59, 589
 - throw, 426–427
 - `try...catch`, 423–427
 - with
 - avoiding to improve execution time, 581
 - description of, 58
- static properties of RegExp object, 214–215**
- static scope, 88–89**
- status bar, 144–145**
- `status()` **method of window object, 145–146**
- stopping event bubbling, 278–279**
- `stopPropagation()` **method of event object, 297–298**
- storing commonly used value, 590–591**
- String **class. See String object**
- string concatenation, avoiding, 590**
- String **object**
 - description of, 29–32
 - `indexOf()` method, 237
 - `localeCompare()` method, 368
 - `match()` method, 195
 - `replace()` method, 195–196
 - `search()` method, 195
 - `split()` method, 72, 196
 - `trim()` method, 205–206
- String() **type cast, 25**
- String value**
 - converting to, 22–23
 - description of, 20–21
 - as primitive type, 16
- StringBuffer **class/object, 97–98, 528**
- strings**
 - arrays and, 70–71
 - comparing, 49–50
 - concatenating, 97–98, 590
 - error-proofing, 571–573
 - hard-coded, avoiding, 569–570

strings (continued)

- as immutable, 97
- relational operator and, 49
- Unicode and, 570–571

style object

- accessing style sheet, 309–312
- collapsible section and, 308–309
- custom tooltips and, 307–308
- description of, 303–305
- methods, 305–307

style property for CSS rule, 310

style sheet. See CSS (Cascading Style Sheets); XSLT (eXtensible Stylesheet Language Transformations)

style sheet object, 309–310

subclass

- creating, 104–105, 112–114
- description of, 103

submit event, 286

submit() method, 340–341

submitting form, 340–342

substring() method of String class, 31–32

subtract operator (-), 48–49

Sun Microsystems

- HotJava, 531
- Java Development Kit, 545
- Java Plugin, 543
- LiveScript and, 2

superclass, 103

surroundContents() method, 326

SVG (Scalable Vector Graphics)

- description of, 8, 133
- example of, 133–134
- <script/> tag, 134–135
- tag placement, 135

SVG Viewer (Adobe), 532, 541

switch statement

- description of, 58–59
- if statement compared to, 589

Synchronized Multimedia Integration Language (SMIL), 8

synchronous mode, loading file in, 447

syntax

- ECMAScript, 11–12
- variables, 12–14

syntax error

- description of, 412–413
- Internet Explorer, 449–450
- Mozilla, 454–455
- onerror event handler and, 422

system dialog, 143–144

system drag and drop

- advantages and disadvantages of, 399
- dataTransfer object and, 393–397

- description of, 387–388

- dragDrop() method, 397–399
- dragged item events, 388–389
- drop target, creating, 392–393
- drop target events, 389–390
- events, using all, 390–392

T

tabbing forward in text box, 346–347

table, sorting

- in descending order, 376–377
- with different data types, 377–381
- multi-column
 - comparison function, 374–375
 - overview of, 373–374
 - sortTable() function, 375–376
- one-column
 - comparison function, 371
 - overview of, 369–371
 - sortTable() function, 371–373

table methods, 179–182

tabs, removing to optimize download time, 574

tabular data control, 558–560

tags

- <body/> tag, 128
- <head/> tag, 128
- modifying text between starting and ending, 314–315
- <noscript/>, 130, 255–256
- placement of
 - HTML, 128–129
 - SVG, 135
- <script/>
 - HTML, 125–126
 - SVG, 134–135
 - XHTML, 131
- XML, 159, 160

target of event, 277

target property of event object, 298–299

Technical Committee 39, 597

Temperature Service (XMethods)

- code for, 512
- cross-browser approach to, 527–529
- Internet Explorer example, 515–516
- Mozilla example, 520–521
- WSDL proxy example, 523–525

TemperatureService object, 527–529

template (XSLT), 472, 476

test() method of RegExp object, 194

testing

- for browser, 455–456
- form submission, 340

- MIME type, 537–538
 - range collapse, 327
 - string concatenation, 98
 - text box**
 - allowing valid characters in, 350–351
 - autosuggest
 - description of, 362
 - matching and, 362–363
 - script for, 363–365
 - blocking invalid characters in, 349
 - blocking paste function, 351–353
 - description of, 342–343
 - limiting textarea characters, 347–348
 - numeric, with up and down arrow keys, 354–356
 - retrieving or changing value, 343–344
 - selecting text in, 344–346
 - tabbing forward automatically in, 346–347
 - validating `onblur`, 353–354
 - Text **node**, 168
 - text range**
 - description of, 317
 - DOM and
 - clean up, 329
 - cloning range, 328
 - collapsing range, 326–327
 - comparing ranges, 328
 - complex selection, 322–323
 - inserting with range content, 325–326
 - interacting with range content, 324–325
 - overview of, 317–318
 - simple selection, 318–322
 - Internet Explorer and
 - cloning range, 332
 - collapsing range, 331
 - comparing ranges, 331
 - complex selection, 330
 - interacting with range content, 330
 - overview of, 329
 - simple selection, 329–330
 - practicality of, 332
 - `<textarea/>` **element**, 336, 343
 - TextUtil **object**, 348
 - this** **keyword**
 - ActiveX objects and, 448
 - description of, 89–90
 - object masquerading and, 105–107
 - style object and, 305
 - this** **object and event handler**, 281
 - throw** **statement**, 426–427
 - throwing own error**, 431–432
 - timeout, setting**, 146–148
 - `toExponential()` **method of Number class**, 28
 - `toFixed()` **method of Number class**, 28
 - `toLocaleLowerCase()` **method of String class**, 32
 - `toLocaleString()` **method of Array class**, 72
 - `toLocaleUpperCase()` **method of String class**, 32
 - `toLowerCase()` **method of String class**, 32
 - Tools**→**JavaConsole**, 429
 - Tools**→**Web Development**→**JavaScript Debugger (Mozilla)**, 435
 - tooltip**, 146–147, 307–308
 - top** **object (BOM)**, 137
 - `toPrecision()` **method of Number class**, 28
 - `toString()` **method**
 - Array class, 71–72
 - Date class, 78–79
 - description of, 22
 - Function class, 64–65
 - location object, 154–155
 - Object class, 26
 - `TotalFrames()` **method (Flash)**, 553
 - `toUpperCase()` **method of String class**, 32
 - `transformNode()` **method (XSLT)**
 - creating for Mozilla, 478
 - Internet Explorer, 473–475
 - transversal**, 182–189
 - tree-based API**, 163–166
 - TreeWalker object (DOM)**, 187–189
 - `trim()` **method of String object**, 205–206
 - trusted site**, 565
 - `try...catch` **statement**
 - description of, 423–424
 - determining type of error, 426
 - Error object and, 424–425
 - nested, 424
 - raising exceptions, 426–427
 - two's complement format**, 38
 - type**, 16
 - type casting**, 24–25
 - `typeof` **operator and primitive type**, 16–17
- ## U
- UML (Universal Modeling Language)**, 103–104
 - Unary operators**
 - `delete` operator, 33
 - plus and minus, 36–37
 - postfix increment/decrement, 35–36
 - prefix increment/decrement, 34–35
 - void operator, 33–34
 - Undefined value**
 - description of, 17–18, 226
 - as primitive type, 16

underscore (`_`)

underscore (`_`)

- to indicate private scope, 88
- variable name and, 13

Unicode characters, 570–571

Unicode Standard

- description of, 4, 20
- representing character using, 198

Uniform Resource Identifier (URI), 81

Universal Browser Read privilege (Mozilla), 517–518, 567

Universal Browser Write privilege (Mozilla), 567

Universal Modeling Language (UML), 103–104

Universal Time Code (UTC), 77

Unix operating system, detecting, 248–249

unload event, 286, 287–288

unrolling loop, 585–588

unshift() method of Array class, 75

unsigned integer, 37

unsigned right shift operator (>>), 42–43

untrusted site, 565

URI (Uniform Resource Identifier), 81

URL and location object (BOM), 153

user-agent string detection

- history of
 - Internet Explorer 5.0 and higher, 230
 - Mozilla, 230–231
 - Netscape Communicator 4.0 and Internet Explorer 4.0, 229
 - Netscape Navigator 3.0 and Internet Explorer 3.0, 227–229
 - Opera, 232–233
 - Safari, 233
- overview of, 226–227
- storing in local variable, 234–235

userService() method of WebService component, 513–514

UTC (Universal Time Code), 77

UTC() method of Date class, 78

V

validating

- credit card, 218–221
- data against database, 506
- dates, 216–218
- e-mail address, 222
- form before submission, 341
- onblur, 353–354

value attribute and sorting, 382–383

value property of text box, 343–344

valueOf() method

- Array class, 71–72
- Date class, 78–79

Function class, 64–65

Object class, 26

values

Boolean

- description of, 18
- as primitive type, 16
- replacing to optimize download time, 576–577
- `toString()` method of, 22

cookie and, 482

- `dropEffect` property of `dataTransfer` object, 396
- `effectAllowed` property of `dataTransfer` object, 396

floating-point

- creating, 19
- sorting and, 379

infinity, 20

integer

- bitwise operator and, 37–39
- creating, 18–19

invalid, 411–412

iterative, inserting, 591

Null, 16, 18

Number

- converting to, 23–24
- description of, 18–20
- as primitive type, 16
- `toString()` method of, 22

primitive, 15–16

reference, 15–16

storing commonly used, 590–591

String

- converting to, 22–23
- description of, 20–21
- as primitive type, 16
- text box, retrieving or changing, 343–344

Undefined, 16, 17–18, 226

variables and, 13

var operator

- description of, 12–14
- loosely typed variable and, 11

variables

- declaring, 14
- defining, 12
- defining multiple, 591
- local, using, 580–581
- as loosely typed, 11, 13
- names, replacing to optimize download time, 575
- naming, 13–14
- values and, 13
- watching for, 441–442

VBScript, 557–558

vector graphics, 133**Venkman**

- breakpoint, setting, 439–440
- description of, 435
- Loaded Scripts pane, 437–439
- profiling, 442–443
- running, 435
- stepping through code, 440–441
- watches, 441–442
- window, 435–437

very late binding, 101**View→Script Debugger (Internet Explorer), 432–433****void operator, 33–34****W****Watches view (Venkman), 442****watching for variable, 441–442****Web browser. See also browser detection script; specific browsers**

- detection methods
 - cross-browser code and, 292
 - object/feature, 226
 - user-agent string, 226–233
- DOM and, 7, 8, 167
- ECMAScript and, 3–4, 5
- hiding code from, 129–130
- possessive quantifiers and, 204
- Text node and, 168

Web services

- cross-browser approach to
 - overview of, 525
 - TemperatureService object, 527–529
 - WebService object and, 525–527
- description of, 509–510
- Internet Explorer and
 - overview of, 513
 - WebService component, 513–516
- Mozilla and
 - enhanced privileges, 517–518
 - overview of, 516
 - SOAP methods, using, 518–521
 - WSDL proxies, using, 522–525

Web Services Description Language (WSDL), 510–513**Web sites**

- algorithms, 584
- A9 (Amazon.com Web search engine), 506
- applet security and signing, 548
- Bitflux blog, 506
- credit card number patterns, 221
- Dithered JavaScript compression utility, 593
- ECMAScript Cruncher tool, 576
- ECMAScript for XML, 616

- ECMAScript Language Specification, 12
 - Greenburg, Jeff, 586
 - “JavaScript Performance Issues,” 578
 - “JavaScript Security: Signed Scripts” document, 566
 - JavaScript Verifier, 432
 - language-specific DOMs, 167
 - Luhn algorithm, 221
 - Macromedia Flash Detection Kit, 543
 - Microsoft
 - HyperText Applications, 568
 - JScript.NET, 605
 - OLE/COM Object Viewer, 540
 - Script Debugger, 432
 - Script Encoder, 594
 - WebService component, 513
 - Mozilla, 563
 - Netscape DevEdge, 120
 - Sun Microsystems
 - Java Development Kit, 545
 - Java Plugin, 543
 - technical comparison of SGML and XML, 160
 - Venkman, 435
 - “Writing Efficient C and C Code Optimization,” 582
 - Wrox, 563
 - XMethods, 512
 - XMethods WSDL analyzer tool, 527
 - zDragDrop library, 405
 - zInherit library, 116
- Weblogic Workshop (BEA Systems), 616**
- WebService component (Microsoft)**
- example, 515–516
 - using, 513–515
- WebService object, 525–527**
- WebServiceProxyFactory object (Mozilla), 522**
- while statement, 55**
- white space, trimming, 205–206**
- Wikipedia Web site, 584**
- window object (BOM)**
- alert() method, 143
 - back() method, 148
 - clearInterval() method, 147
 - clearTimeout() method, 146
 - close() method, 142
 - confirm() method, 143–144
 - description of, 136–139
 - forward() method, 148
 - go() method, 148
 - history property, 148–149
 - length property, 149
 - manipulating windows, 139–140
 - onerror event handler, 419–420
 - onload event handler and, 288
 - open() method, 140–142, 565

window object (BOM) (continued)

window **object (BOM) (continued)**

- `prompt()` method, 144
- `setInterval()` method, 147
- `setTimeout()` method, 146–148
- `status()` and `defaultStatus()` methods, 145–146

windows. See also **window object**

- Microsoft Script Debugger, 433–434
- moving and resizing, 139–140
- navigating and opening new, 140–143
- security issues related to, 564–566
- Venkman, 435–437

Windows (Microsoft)

- error reporting and, 413–415
- operating system, detecting, 245–247
- XP Service Pack 2, 516, 558, 565

Windows Media Player (Microsoft), 532, 542

with **statement**

- avoiding to improve execution time, 581
- description of, 58

World Wide Web Consortium (W3C), 7

`write()` method of **document object**, 151–152

“Writing Efficient C and C Code Optimization” (Koushik Ghosh), 582

writing Java applet, 545–546

WSDL (Web Services Description Language), 510–513

WSDL proxies, 522–525

W3C (World Wide Web Consortium), 7

X

xbObjects

- example of, 122–124
- purpose of, 120–122

XHTML (eXtensible HTML)

- attributes and, 348
- description of, 159–160
- example of, 161–162
- `<script/>` tag, 131–132

XMethods

- Temperature Service
 - code for, 512
 - cross-browser approach to, 527–529
 - Internet Explorer example, 515–516
 - Mozilla example, 520–521
 - WSDL proxy example, 523–525
- WSDL analyzer tool, 527

XML (eXtensible Markup Language). See also **ECMAScript for XML**

- API for, 162–163
- DOM support for
 - in Internet Explorer, 445–450
 - Mozilla and, 450–455

HTTP request

- copycat implementations, 496
- description of, 493–495
- GET request, performing, 496–497
- header, using, 495–496
- POST request, performing, 497–498
- using with both browsers, 463–465
 - error handling, 459–463
 - IE branch, 456
 - modifying DOM creation, 455–456
 - Mozilla branch, 456–459

XML class (ECMAScript for XML), 608–614

XML prolog, 162, 166

xml property (XML DOM)

- Internet Explorer, 449
- Mozilla, 453–454

XmlDom() function, 455–456

XMLList class (ECMAScript for XML), 614–615

XMLSerializer object (Mozilla), 453, 454

XOR operator (^), 40–41

XPath

- Internet Explorer and, 467
- Mozilla and, 467–471
- overview of, 466
- support for, 465

XPathEvaluator object (Mozilla), 467–468

XPathResult object (Mozilla), 468–471

XSLT (eXtensible Stylesheet Language Transformations)

- overview of, 471–473
- support for
 - in Internet Explorer, 473–477
 - in Mozilla, 477–479

XSLTProcessor object (Mozilla), 477–479

Z

zDragDrop library

- creating draggable element, 406
- description of, 405
- drop target, creating, 406
- events, 406–407
- example, 408–409

zInherit

- description of, 116
- dynamic prototyping and, 118–119
- example of, 116–117
- multiple inheritance and, 119–120