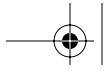


I N D E X

- !, 148
 - \$, 145–46
 - %, 148
 - &, 68, 148
 - &&, 68, 148, 149
 - *, 148, 519
 - *=/=%+=-=<=>>, 148
 - +, 148, 158, 582
 - ++, 148
 - , 148
 - , 148
 - , 148
 - ., 148
 - /, 148, 631, 726
 - /*, 142
 - //, 142
 - ;;, 142–43, 194
 - ::, 142–43
 - <, 68, 367–68
 - <<, 68, 148
 - =, 148
 - ==!=, 148
 - ===!=, 148
 - >, 367–68
 - >>, 148
 - ?:, 148
 - \, 631, 741
 - ^, 148
 - _ , 145–46
 - {}, 143
 - |, 148
 - ||, 148
 - “, ”, 145
 - (), 148
 - ~, 148
 - ..., 85, 519
 - ?:, 149
 - <- ->, 367
 - <>, 367
 - <> <=>=, 148
 - [], 148
 - {}, 148
 - ‘ ’, 145
 - = =, 158
 - >>>=&^= |=, 148
 - <, 158
- ## A
- accel, 222
 - acceltext attributes, 253, 280
 - accessibility, 244, 269–70
 - accesskey attribute, 111, 222–23, 230, 280
 - accumulate() function, 161
 - action, 17
 - action(), 190
 - action() function, 188, 319–20, 321, 356, 689
 - action() method, 188, 232–33, 259, 260–61
 - action attribute, 590
 - <action> tag, 532–34
 - simple syntax content and, 532–34
 - ActiveState, 6
 - Komodo development environment, 138
 - addBinding() method, 597
 - addDownload(), 658
 - addEventListener() method, 207, 234, 245, 476, 569, 589
 - addFile() function, 749
 - adding text, 95–96
 - addObserver() method, 216, 544
 - address book, 28–33
 - Adobe PhotoShop, 267
 - advanced gesture support, 229
 - advanceFocus() method, 316
 - alert(), 156, 194, 297, 337–38, 638, 719
 - align attribute, 50, 56, 112
 - allowevents attribute, 255, 445
 - allownegativeassertions attribute, 521
 - alphanumeric characters, 145–46
 - <Alt> tag, 382, 383, 390
 - Amiga, 22
 - anchor point, 255
 - anonymous content, 568
 - anonymous functions, 154
 - ANSI C, 139
 - anti-aliasing, 90
 - AOL 8.0 for MacOS X, 5
 - AOM
 - interfaces, 447–49, 460–62
 - using commands via, 310–14
 - AOM objects, 429
 - properties and methods of, for <listbox>, 447–48
 - AOM template objects, 542–43
 - AOM/DOM, creation of dialog windows from Java-Script using, 332
 - Apache Web server, 21, 138
 - APIs (Application Programming Interfaces), 138
 - append() method, 679
 - appendChild() tag, 514
 - appendRelativePath() method, 630
 - application, 8
 - application file system aliases, 667–68
 - application install, 711
 - Application Object Model, 165, 174
 - application shell, 13
 - application window, 266
 - resizing, 118–19
 - applyauthorsheet attribute, 591
 - applyauthorstyles attribute, 591
 - applybindingsheet attribute, 591
 - arg2var() method, 608
 - argc count, 623–24
 - argument passing, 158





Array() constructor method, 146
 array literals, 146
 Array objects, 156
 arrays, 146–48
 <arrowscrollbox> tag, 114, 254, 273–74
 Assert() method, 545, 692, 694, 701
 association between objects, 154
 asynchronous submission, 249
 asyncOpen() method, 647
 att2var() method, 607, 608
 attachto attribute, 591
 auto-completion of forms, 20
 <autorepeatbutton> tag, 109, 113–14, 115, 273
 awk, 139

B

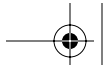
backwards-compatibility issues, 363
 <Bag> tag, 382, 383, 390, 405
 Banshee Screamer, 122
 base tag, 520–25
 baseline, 715
 basetext binding, 571
 BE, 8
 behavior, 17
 BiDi text (bi-directional text), 226
 bindingattached event, 591
 bindingdetached event, 591
 <binding> tag, 531–32, 539, 574–75
 bindings, 17, 621. *See also* XBL bindings
 inner content, 594
 input façade, 593–94
 integrating, 611–13
 primary, 570
 variable, 531
 zero content, 592–93
 bitwise operations, 149
 bloat, 8
 Blob type, 395
 block, 45
 block scope, 159
 blur() method, 269
 <body> tags, 355, 585, 586, 587
 boilerplate text, 95
 Bookmark Manager, 436, 515
 bookmarks, 113
 Boolean type, 144, 156

borders, 92–93
 decorations, 89
 styles, 77–79
 bottom up method, 416, 417
 bound tags, 568, 595
 merging, 581
 Box Model, 45
 <box> tag, 42, 43–44, 54, 89, 212, 213, 422, 573
 boxes, 54–55
 common layout attributes, 49–51
 decorations, 45
 dialog, 329–32
 layout tags, 54–64
 boxes, 54–55
 flex and spacers, 55–58
 grids, 61–64
 <groupbox> tag, 64–65
 stacks and decks, 58–61
 brands, 712
 <broadcaster> tag, 211, 214
 broadcasters
 JavaScript, 214–15
 XPCOM, 215–16
 XUL, 211–14
 <broadcasterset> tag, 211
 Browser Object Model, 165, 174
 browser plugins, 180
 <browser> tag, 174, 343
 B-tree, 109
 Bugzilla bug database, 26
 build number, 714
 builder property, 542
 builders, 174, 463–64, 543–44
 built-in objects and similar effects, 157–58
 built-in property names, 470–71
 <bulletinboard> tag, 58, 61
 bundle, 706
 bundling up NoteTaker, 744–51
 business process modelling standards, 655
 button attribute, 331, 590
 <button> tag, 108–9, 110–12, 213, 240, 245, 297, 312, 313, 571
 button-base binding, 571
 buttons, 104–5, 130
 defining real, 105–7
 diagnosing, 133–35
 NoteTaker, 130–33
 XUL, 109–21
 byte leader character, 145

C

C language, 12, 13, 14, 139, 142, 159, 160, 416
 C# language, 139, 153, 155, 159, 160
 C++ language, 12, 13, 14, 24, 109, 139, 316, 416
 cache control, 662–63
 calcRelatedMatrix() method, 487, 700
 calcTopOfTree(), 488
 calculate() method, 487
 calculate_params() function, 717
 calcViewMap() method, 489
 <caption>, 64–65
 cascadesOverlay.xul, 418
 catch block, 150
 CDATA literals, 68–69
 CellText() view method, 467–68
 centerscreen, 335
 certified security, 672–73
 Change() method, 545
 change notifications, 304–5
 channels, 639, 645–47
 character entity references, 85
 characters
 alphanumeric, 145–46
 byte leader, 145
 prefix, 619–21
 Unicode, 145
 UTF-8-encoded, 142
 charcode attribute, 590, 591
 <checkbox> tag, 116, 120, 245
 checked attribute, 111
 <children> tag, 578–81
 Chimera, 7
 chrome, 17–19, 720
 chrome directories, 28–33
 chrome locale system, 99
 chrome registry, 412, 424–27
 implementation predicates, 427
 top-level predicates, 426–27
 chrome URL, 18, 19–20
 ChromeWindow, 333
 class, 66, 621
 Classic Browser, 4, 417
 Classic Mail & Newsgroup client, 28
 Classic Mozilla, 4
 main versions of, 4–5
 clear() method, 557





- click() method, 593
 - click events, 226–27
 - click handler, 188
 - clickcount key, 590
 - clipboardEmpty() method, 313
 - close() method, 357
 - closures in JavaScript, 161
 - coalesceduplicatearcs attribute, 521
 - Cocoa interface, 7
 - collapse, 50
 - collapseToolBar() method, 277
 - <colorpicker> tag, 288, 565
 - column picker, 460
 - <column> tag, 62, 64
 - COM (Common Object Management), 14
 - command attribute, 66, 308
 - command design, 317–22
 - command handler, 300
 - command line, 76, 712
 - arguments, 623–24
 - options, 28, 29
 - <command> tag, 217, 297, 307, 308, 309–10, 313, 321
 - commands, 216–17, 295–323, 389
 - catching unexpected, 322–23
 - change notifications, 304–5
 - command pattern and controllers, 300–303
 - concepts, 298–305
 - controller sites, 303
 - design, 317–22
 - dispatchers and dispatching, 303–4
 - enhancing, to process RDF content, 689–700
 - existing, 316–17
 - Functor pattern, 299–300
 - Mozilla and, 296–98
 - names, 712
 - origin of, 305–7
 - pattern and controllers, 300–303
 - using
 - via AOM, 310–14
 - via XUL, 307–10
 - XPCOM components and, 314–16
 - <commandset> tag, 217, 307, 309–10, 430
 - commandupdate event, 309
 - commandupdate, 66
 - comparison operators, 158
 - component name service, 14
 - component registry, 14
 - components, 2, 621–22
 - Components array, 185
 - Composer, 3
 - Compuserve 7.0, 5
 - concrete fact, 378
 - <conditions> tag, 529–31
 - confirm(), 338, 719
 - Constructor() method, 179
 - <constructor> attribute, 606
 - <constructor> tag, 587–88
 - consume1() function, 212–13
 - Consumer design pattern, 20
 - container tags, 392
 - container test, 522–24
 - containers, 374–75, 383–84, 534
 - containment, 154
 - containment attribute, 539
 - content, 8, 362
 - content delivery events, 217
 - Content Management, 398
 - content options, 650, 652
 - content processing architecture, 640–41
 - content selection, 224–27
 - content sinks, 20
 - content support, 648
 - <content> tag, 530, 568, 576–81
 - contentdestroyed event, 591
 - contentgenerated event, 591
 - content-like information, 362
 - content_poll() function, 558, 683, 691
 - contenttooltip attribute, 328
 - context menus, 113, 242, 327, 328, 329
 - continue, 150
 - Contract IDs, 2
 - Control-C, 198
 - Control-left-clicking, 226–27
 - controller sites, 303
 - Control-V, 198
 - Control-X, 198
 - CORBA (Common Object Request Broker Architecture), 14
 - CORBA IDL specification technology, 14
 - core XML standards, 9–10
 - corner rounding, 93–94
 - COTS (common off-the-shelf software), 25
 - createBundle(), 191
 - createElement() method, 345
 - createInstance() method, 178, 179, 180
 - createRange() method, 225
 - createTransportOfType() method, 634–35
 - creative extensions, 648
 - crop attribute, 85, 111, 253
 - cross-browser compatibility, 204
 - cross-platform HTML, 204
 - CSS (Cascading Style Sheet) styles, 2, 15
 - from HTML forms, 257–58
 - CSS2, 15, 45
 - CSS2 scrollbars, 89
 - CSS2 standard, 125–26
 - CSS2 styles, 43, 415
 - CSS3, 94
 - custom panel systems, 286–87
 - custom RDF tree views and data sources, 700
 - custom tags and objects, 17
 - custom tree views, 484–93
- ## D
- data, 362
 - creating test, 547–49
 - meta, 363, 376
 - moving user-entered, into RDF, 688–89
 - relational, 369
 - seminal view, 462–63
 - using URNs for plain, 394–96
 - data sources, 20, 516, 640, 647–52
 - custom, 700
 - design of, 677–78
 - dynamically allocating, to templates, 680–84
 - in-memory, 650
 - internal, 640, 650
 - issue of, 516
 - RDF, 641
 - revealing content, 701–2
 - setup, 678–80
 - working with, 700–702
 - data structures
 - algorithms and, 624–25
 - facts versus, 366–69





- data transfer, 638–52
 - channels, 645–47
 - content processing, 638–69
 - sinks, 638–40
 - sources, 638–40, 647–52
 - streams, 640–44
 - transports, 644–45
- data types, 143–45
 - Boolean, 144
 - Floating point, 144
 - Null, 143, 144
 - Number, 144
 - Object, 143, 145
 - String, 145
 - typeof, 143–45
 - undefined, 143
 - void, 143
- database, 174
- database property, 542
- databases, 625–27
 - Bugzilla bug, 26
 - overlay, 417–20
- data-driven listboxes and trees, 481–83
 - custom tree views, 484–93
 - dynamic listboxes, 483–84
- datasource, 174
- datasources attribute, 520, 539, 542
- Date object, 156–57
- dblclick, 205
- DCOM (Distributed COM), 14
- debug attribute, 77
- Debugger, 37
- debugger keyword, 195
- debugging, 76
 - building up template, 560–61
 - catching unexpected commands, 322–23
 - detecting bad XUL, 75–79
 - detecting events, 235–36
 - diagnosing buttons and skins, 133–35
 - diagnosing events and forms, 261–63
 - diagnosing key problems, 235–36
 - diagnostic windows, 357–59
- DOM Inspector in, 37, 38, 99–101, 126, 333, 417, 429, 430
- dumping RDF, 407–8
- finding object properties for forms and menus, 263
- important preferences in, 35–36
- logging and testing, 751–52
- making <listbox> and <tree> work, 493–94
- multiwindow development, 37–38
- navigation problems, 292–93
- from outside, 35–38
- overlay tips, 432–33
- picking apart complex widgets, 262–63
- script diagnosis, 193–95
- springs and struts, 77–79
- template survival guide, 559–61
- unpopping menus, 261–62
- warnings and silence, 76–77
- working with data sources, 700–702
- XBL diagnosis, 614–15
- <deck> tag, 58, 60–61, 186, 188, 291, 327
 - scripting, via AOM and XBL, 189–90
- decrement, 115
- default actions, 568–69
- delegates, 544–45
- delegation, 153, 465
- dependent, 335
- deploy() function, 746
- deployment, 705–52
 - bundling up NoteTaker, 744–51
 - install strategies, 707–12
 - install technologies, 725–43
 - file uses and formats, 725–26
 - Mozilla registries, 726–28
 - Web-side objects, 731–32
 - XPCOM objects, 743
 - XPIInstall-side objects, 732–42
 - XUL wizards, 728–31
 - remote, 706, 712–25
 - assigning names, 712–14
 - assigning versions, 712–14
- <Description> tag, 33, 48, 75, 84–85, 96, 98–99, 105, 133, 244, 270, 312, 313, 330, 382, 386–88, 390, 392, 393, 405, 603
- <destructor> attribute, 606
- <destructor> tag, 587–88
- destructured testing, 25
- DHTML objects, 83
- diagnostic stylesheet, 262
- diagnostic windows, 357–59
- dialog, 335
- <dialog> attribute, 331
- dialog boxes, 329–32
 - modal, 329
 - special-purpose XPCOM, 339–40
- <dialog> tag, 330–32, 344, 728
- <dialogheader> tag, 281, 332
- digital certificates, 672
- digital security, 707
- digital signing, 721
- dir attribute, 49, 111
- disabled attribute, 221–22, 307
- disableKeyNavigation attribute, 443
- dispatchers and dispatching, 303–4
- dispatchEvent() method, 305, 306, 310
- display attribute, 47, 575
- dlgtype attribute, 331
- doCommand() method, 298, 305–6, 694, 695
- DocShell, 341
- <!DOCTYPE> declaration, 574
- document type definitions, 85–86
- document.getElementById() method, 597
- document.getSelection() method, 225
- documents, 8
 - embedding, in panes, 340–44
 - fragments of, 10
 - mixing, 344–45
- do_custom_click() method, 590
- dogfood, 8
- DOM (Document Object Model), 2, 42, 52
 - scripting deck via, 186–88
 - scripting styles via, 188–89
- DOM Event Model, 204–8
- DOM events, 217
- DOM hierarchies, 412
 - relationship with XBL bindings, 568–69
- DOM Inspector, 37, 38, 99–101, 126, 333, 417, 429, 430
- DOM layer of functionality, 12





DOM standards, 166–73, 180
 DOM 0, 166
 DOM 1, 167–69, 184
 Document interface, 12
 DOM 2, 169–71
 DOM 2 Events, 21, 198
 DOM 3, 17, 171–72, 204
 DOM 3 Events, 198, 205, 218, 224
 <DOMImplementation> interface, 172
 dont-build-content, 520–21, 524
 dont-test-empty, 521
 doPaste () function, 312
 dot-dashed lines, 72
 double-headed arrows, 72
 Download Manager, 393–94, 396–97, 658
 drag and drop, 206, 227–29
 dragdrop, 227
 dragenter, 227
 dragexit, 227
 draggesture, 227
 dragover, 227
 drill-down algorithm, 508–9
 drop-down menus, 242
 <dropmarker>, 115–16
 DSSSL (Document Style Semantics and Specification Language), 15
 DTDs (document type definitions), 15
 Dublin Core, 380, 398
 dump () method, 194, 357
 dumpFromRoot () function, 702
 dumpRDF (), 408
 Dynamic HTML, 83, 138
 scripting in, 12
 dynamic <listbox>, 483–84

E

ebXML (enterprise business XML), 655
 ECMA-262 Edition 4 enhancements, 163
 ECMA-262 standard, 140
 ECMA-290 “ECMAScript Components,” 140
 ECMA-327 “ECMAScript Compact Profile,” 140
 ECMAScript, 140
 standard, 620

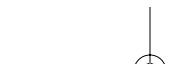
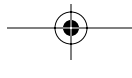
ECMAScript Edition 3, 141–62
 objects, 152–58
 built-in, and similar effects, 157–58
 host, 155–56
 native, 156–57
 plain, 153–55
 processing concepts, 158–62
 argument passing, 158
 closures, 161
 operator precedence, 158
 prototype chains, 161–62
 run-time evaluation, 160–61
 scope, 159–60
 stacks and garbage collection, 160
 type conversion, 158–59
 syntax, 141–52
 arrays, 146–48
 data types, 143–45
 expressions, 148, 149
 flow control, 150–51
 functions, 151–52
 regular expressions, 152
 statements, 142–43
 text layout, 142
 variables, 145–46
 Edit dialog changes, 683–84
 Edit panel, 688
 editable attribute, 256
 <editor> tag, 342
 <element> tag, 591
 embedded software, 24
 enableColumnDrag, 460
 encodeURI () function, 249
 end user, 266, 630–31
 entity references, 85
 Entity Relationship Modeling, 398

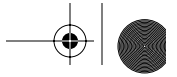
enumerators, 624
 environment, 628
 equal-precedence operators, 149
 equalsize attribute, 50
 Error object, 157
 errorHandler () function, 717
 European Computer Manufacturers' Association (ECMA), 140
 eval () method, 160–61, 208
 event.init (), 209
 event handlers, 203, 588
 systematic use of, 476–80
 XUL support for, 207–8
 event queue, 203
 event target, 204

event-oriented technologies, 632
 events, 2, 197–236, 638
 broadcasters, 210–16
 click, 226–27
 commands, 216–17
 content delivery, 217
 debugging, 235–36
 detecting, 235–36
 diagnosing, 261–63
 DOM Event Model, 204–8, 217
 XUL support for event handlers, 207–8, 217
 keystrokes, 217–23
 mouse gestures, 223–29
 NoteTaker user input, 230–34
 observers, 210–16, 217
 origin of key, 219
 scheduling input and output, 199–204
 style options, 229
 timed, 208–10, 217
 examineForProxy () method, 634
 exec () method, 300
 execute () function, 298, 320, 321, 356
 execute () method, 26
 explicit content, 568
 expressions, 148, 149
 extended template syntax, 517
 extended template variables, 517, 518
 extends attribute, 574
 extensibility, 23
 extrachrome feature, 334

F

fact objects, 380
 fact store, 375, 378
 fact subjects, 380
 factlike information, 362–63
 factoring, 370
 factory objects, 648–50
 facts, 362, 366, 400
 about facts, 375–78
 component types, 395
 concrete, 378
 data structures versus, 366–69
 ground, 378, 503
 single queries, 535





standard arrangement, 526–27
types, 395
ways to organize, 372–75

FastLoad, 708

FE, 7

fetch() method, 586

<field> attribute, 606

<field> tag, 583–84, 584

file descriptors, 632

file handles, 632

file pointers, 632

file system directory, 630, 663–70

file uses and formula, 725–26

FilePicker, 289, 339–40

<FilePicker> tag, 289

FilePicker dialog box, 289

files, 628–32, 638

- downloading, 657–58
- uploading and posting, 659
- working on, 632

FileSpecObject, 737

finally block, 150

Find As You Type, 223

finished() function, 250

Firebird, 3

flags attribute, 520

flex attribute, 50, 56, 85

float, 47

floating-point literals, 144

floating-point numbers, 149

floating-point operations, 139

<floatingview> tag, 287

flow control, 150–51

Flush() method, 541

flyover help (tooltips), 327–28

focus() method, 244, 269

focus ring, 268–69

folders, 628–32, 638

- working on, 632

font systems, 89–92

<fontpicker> XUL tag, 289

footprint, 715

for statement, 150

foreign-type systems, 622–23

forms, 242–50

- accessibility of, 244
- common properties of XUL elements, 243–44
- comparison of XUL and HTML, 240–42
- control of, 243
- diagnosing, 261–63

- finding object properties for, 263
- navigation of, 243
- simple templates for elements, 549–52
- submission of, 83, 247–50
- XUL tags for, 245–47

4GL tools, 138

frame, 45, 52

frame object, 108

<FRAME> tag, 344

FTP session, creating, 635–36

full install, 711

Function object, 157

functions, 151–52, 153. *See also specific function names*

Functor pattern, 299–300

G

Galeon, 7

gaps, 204

garbage collection, 160

Gecko, 4

Gecko content display model, 15–16

Gecko Runtime Engine, 4

general-purpose scripting, 623–38

- command line arguments, 623–24
- data structures and algorithms, 624–25
- databases, 625–27
- environment, 628
- files, 628–32
- folders, 628–32
- interrupts, 632
- network protocols, 633–36
- processes, 636–38
- signals, 632
- threads, 636–38

general-purpose XUL

- attributes, 65–66

generate_html() function, 355

generate_style() function, 355

gestalt() method, 737

gestures

- advanced support, 229
- mouse, 223–29

getAEBB() method, 608

getAEBB property, 607

getAnonymousElementByAttribute(), 598

getAnonymousNodes(), 598

getAttribute(), 274

getAttributeNS() method, 345

getBindingParent() method, 597

getControllerForCommand() method, 304, 305, 308

getCoords() function, 610

getCurrentSession() method, 228

GetDataSourceBlocking(), 700–701

GetDataSources() method, 545, 700

getElementByID(), 378

getElementByTagName(), 355

getEnumerator() method, 346

getEnvironment() method, 628

getFile() method, 665

getFileFromURLSpec() method, 629

getFullYear() method, 157

getHandlerForParam(), 624

getParentIndex(), 490

getResource() method, 700

getSelection() factory method, 226

getService() method, 178, 179, 645

GetTargets() method, 701

<getter> tag, 584

getXULWindowEnumerator(), 347

getYear() method, 157

GIMP (Gnome Image Manipulation Program), 91

glimpseindex(1), 26

Global object, 157

glyph, 10, 218

GNOME, 91

GNOME Panel, 28

GNU/Linux, 22

GNU/Linux GNOME desktop interface, 7

goDoCommand() function, 304, 306, 320–21

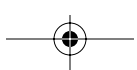
goto statement, 150

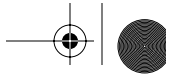
<grid> tag, 26, 61–64

<gripper> tag, 114, 119

grippy, 116–19

<grippy> tag, 109, 116–17, 283





grippytooltiptext attribute, 328
 ground facts, 378, 503
 <groupbox> tag, 64–65, 234
 GTK, 21, 104, 107
 graphics library, 129
 GUI integration, 21–22
 GUI toolkit library, 107

H

handle_click() code, 258
 <handler> tag, 208, 221, 589–90
 <handlers> content, 610
 <handlers> tag, 588–90
 hasFeature() method, 172
 hash table, 109
 hasOverlays, 418
 hasStylesheets, 418
 <hbox> tag, 54, 78
 <head> tag, 355
 heap, 109, 160
 “hello, world,” 33–34, 297–98
 helper objects, 648–50
 helper tools and utilities, 648
 hierarchical organization, 10–11
 horizontal <box>, 48
 host objects, 155–56, 165
 host software, 139
 HTML, 242
 comparison of XUL and, 82–83
 difference between XUL and, 46
 HTML equivalents, 224
 HTML forms
 comparison with XUL forms, 240–42
 CSS styles from, 257–58
 HTML submission, of XUL forms, 248
 <html> tag, 344
 HTML tags, 45–46
 HTTP, 655, 661
 hyperlinks, 269
 hypertext documents, 8

I

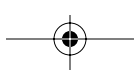
I18n, 8
 id attribute, 307, 387, 414, 574
 id tag identity, 66
 Id-based merging, 420–23
 identifiers, 369, 384–85

IDL (Interface Definition Language), 14
 if keyword, 139
 Iface, 675
 <iframe> tag, 268, 340–42, 343, 344
 illegal template queries, 539
 image attribute, 111
 image resizing, 20
 image rollovers, 88
 <image> tag, 87–89, 245, 575
 images, 87–89
 <implementation> tag, 581–88
 @import, 123
 include() function, 182
 includes attribute, 591
 increments, 115
 index argument, 449
 information
 content-like, 362
 datalike, 362
 factlike, 362–63
 finding, on menus, 242
 inherit, 50
 inherit style attribute, 575
 inheritance chains, 570, 571, 585–86
 init() method, 679, 694
 init_dialog() function, 684
 init_handlers() function, 477
 initInstall() method, 718, 719, 721
 init_toolbar() function, 682, 683
 init_views() function, 482–83
 inline styles, 66
 in-memory data sources, 650
 inner content bindings, 594
 input, scheduling, 199–204
 input façade bindings, 593–94
 insertafter, 423
 insertbefore, 423
 install() function, 747
 install() method, 716, 722
 Install object, 732
 install strategies, 707–12
 install technologies, 725–43
 install trigger script, 748
 installation, 27–28
 installChrome(), 722
 installed-chrome.txt file, 135

Install.GetWinRegistry() method, 741
 InstallTrigger object, 716
 InstallTrigger.install() method, 716
 InstallVersion object, 716
 instance, 8
 Intall object, 737
 Integrated Development Environments (IDEs), 515
 interface, 622
 interface method, 585
 internal data sources, 640, 650
 Internet Protocol (IP) address, finding, 634
 interrupts, 632
 invokeDragSession() method, 227–28
 isCommandEnabled() method, 309
 isCommandEnables() method, 302
 iscontainer attribute, 527
 isempty attribute, 527
 isEnabled() method, 313
 isFinite() method, 144
 island interfaces, 185–86
 isNaN() method, 144
 ISO-16262, 140
 iterators, 624–25

J

Japanese Hiragana, 219
 JAR archives, 708, 749
 working on, 632
 JAR file, 28–29, 32
 JAR naming convention, 30
 Java, 139
 Java applets, 180
 Java Console, 332
 JavaScript, 2, 10, 11, 12, 138, 139, 198, 332, 624. *See also* ECMAScript Edition 3
 broadcasters, 214–15
 interpreter, 637
 language enhancements, 162–64
 observers, 214–15
 operators, 148–49
 role as language, 139–40
 script, 164, 201, 204
 support, 228
 syntax, 138–40, 141–52
 window creation, 332–40





JavaScript Access, 515–16
 JavaScript Console, 37, 38, 195, 358, 416
 JavaScript debugger, 31, 35, 195, 357
 join() method, 638
 JSLib library, 181, 182, 364
 junk electronic mail filtering, 20

K

KDE, 91
 key, 218
 diagnosing problems with, 235–36
 key attribute, 66, 590
 key stroke, origin of key, 219
 <key> tag, 217, 221–22, 253, 269, 588
 <keybinding> tag, 221–22
 keyboard, 219
 keyboard maps, 220
 keycode attribute, 590
 KeyEvents, 172
 keypress, 201, 219
 keypress, 206, 207
 <keyset> tag, 217, 221–22
 keystrokes, 217–23
 availability of keys, 220–21
 Find As You Type, 223
 <key>, 221–22
 <keybinding>, 221–22
 <keyset>, 221–22
 origin of key events, 219
 SBL keyboard maps, 220–21
 XUL accesskey attribute, 223
 keywords, 386, 399, 401
 dropdown menu for, 680
 related, 399, 402
 Keywords, 675
 Keywords button, 131
 keywords pane, 473–93, 688
 data-driven listboxes and trees, 481–83
 custom tree views, 484–93
 dynamic listboxes, 483–84
 laying out <listbox> and <tree>, 473–76
 systematic use of event handlers, 476–80
 K-Meleon, 7
 knowledge bases, 375

knowledge engineers, 362
 Komodo, 6

L

L10n, 8
 label attribute, 111
 <label> tag, 48, 84–85, 119–20, 244, 245, 270, 501, 603
 landed, 8
 language, role of JavaScript as, 139–40
 language enhancements, 162–64
 ECMA-262 Edition 4, 163
 Mozilla SpiderMonkey, 162–63
 security, 163–64
 launch() method, 658
 layout design process, 71–74
 layout strategies
 Mark I, 16
 Mark Ib, 16
 Mark II, 16
 Mark III, 16
 lazy building, 514–15
 tag, 382, 383, 390, 393, 432
 libraries, 622
 graphics, 129
 GUI toolkit, 107
 JSLib, 181, 182, 364
 object oriented, 104
 RDFLib, 181
 RDFLib JavaScript, 677
 scripting, 181–84
 3GL, 104
 line box, 45
 line-wrapping, 84
 links, 8
 navigation of, 83
 Linux kernel, 714
 Lisp, 372, 403, 565
 list processing languages, 565
 <listbox>, 61, 64, 242, 440
 debugging, 493–94
 dynamic, 483–84
 style options for, 469
 <listbox> getRowCount() XBL method, 478
 <listbox> tag, 226, 251, 436, 443
 extended syntax template for, 552–53
 laying out, 473–76
 making work, 493–94
 <listboxbody> tag, 440, 442
 listboxes, 224, 251, 438–39
 AOM and XPCOM interfaces, 447–49
 construction, 439–43
 data-driven, 481–83
 <listbox> tag, 443
 <listcell> tag, 446
 <listcols> tag, 443–44
 <listhead> tag, 444
 <listheader> tag, 444–45
 <listitem> tag, 445
 multicolumn, 439
 RDF and sorting, 446
 user interactions, 446–47
 visual appearance, 438–39
 <listcell> tag, 446
 <listcol> tag, 440, 444
 <listcols> tag, 440, 442, 443–44
 listeners, 203
 differences between observers and, 210
 <listhead> tag, 440, 442, 444
 <listheader> tag, 444–45
 <listheaditem> tag, 442
 <listitem> tag, 440, 442, 445
 <listrows> tag, 440, 442
 literals
 array, 146
 CDATA, 68–69
 floating-point, 144
 number, 144
 types, 395
 using, 631–32
 LiveConnect, 181
 loadBindingDocument() method, 597, 615
 load_data() function, 192, 258
 loadSubScript() method, 178
 loadURI() method, 336
 locales, 412
 configuring and using, 97–99
 localization, 122
 localstore.rdf, 428
 locked toolbars, 275–76
 _LoggedAssert() interface, 698
 _LoggedAssert() method, 694, 695
 _LoggedUnasset() method, 694, 695
 logs, 726
 lug, 78



**M**

Macintosh toolkit, 21
 mailEditorOverlay.xul, 418
 major number, 714
 makefile, 366
 makefile dependency rule, 366
 malloc(), 180
 manifests, 725–26
 manual installation, 708
 many-to-one relationship, 401
 Mark I strategy, 16
 Mark Ib strategy, 16
 Mark II strategy, 16
 Mark III strategy, 16
 markup languages, 105. *See also*
 HTML; XUL
 mask, 121–29
 mastheads, 712
 matching CSS2 properties, 471–
 72
 Math object, 156, 157
 MathML (mathematical con-
 tent), 8, 15
 Mozilla support for, 17
 mediation system, 12
 <member> tag, 531
 menu, 115
 menu bars, 278–80
 <menu> tag, 252, 255, 268, 278–
 79
 <menubar> tag, 269, 278, 280
 <menubutton> tag, 120, 256
 <menuitem> tag, 244, 252–53,
 256
 <menulist> tag, 242, 244, 256
 <menupopup> tag, 112, 115, 254–
 55, 256, 261–62, 327, 329,
 430
 menus, 112, 224, 250–56, 269
 <arrowscrollbox> tag, 254
 context, 242, 329
 drop-down, 242
 finding information on, 242
 finding object properties for,
 263
 <menu> tag, 252, 255, 256,
 268, 278–79
 <menuitem> tag, 244, 252–
 53, 256
 <menulist> tag, 242, 244,
 256
 <menupopup> tag, 112, 115,
 254–55, 256, 261–62,
 327, 329, 430

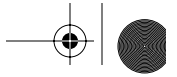
 <menuseparator> tag, 253–
 54
 ordinary, 327, 328
 popup, 250
 <scrollbox> tag, 254
 unpopping, 261–62
 variations of, 256, 279–80
 <menuseparators> tag, 57, 252,
 253–54
 menubutton, 115
 <menubutton> tag, 120
 merging
 attributes with xbl:inherits=,
 576–78
 bound tags, 581
 id-based, 420–23
 options, 423–24
 simple, 420
 tags with <children>, 578–
 81
 Messenger, 3
 metadata, 363, 376, 401
 <method> attribute, 606
 <method> tag, 585–87
 Microsoft Windows behavior, 37
 MIME types, 396–97, 658
 shorthand for, 722
 minor number, 714
 modal, 335
 modal dialog boxes, 329
 Model-View-Controller (MVC),
 106–7
 modifier key, 220
 modifiers attribute, 590
 modular programming, 621–22
 module, 622
 Mork, 626
 Mosaic browser, 3
 mouse gestures, 223–29
 advanced support, 229
 content selection, 224–29
 multiple, 226–27
 rubber band, 226
 drag and drop, 227–29
 window resizing, 229
 -moz-appearance style, 128,
 129, 258, 289
 Mozilla
 creating module, 708
 preferences in, 35–36
 product names in, 2–8
 example applications, 5–6
 platform versions, 4–5
 use of RDF, 364–65
 Mozilla 1.0, 3

Mozilla Browser, 3, 6
 Mozilla Mail, 3
 Mozilla Platform, 2, 11–23, 24,
 198, 618
 architecture, 11–17
 consequences, 21–23
 innovations, 17–21
 programming with, 2
 property names, 738
 in scheduling input and out-
 put, 199–204
 Web applications built on, 5–
 6
 Mozilla Profile Manager, 710
 Mozilla registries, 424, 726–28,
 727
 Mozilla SpiderMonkey enhance-
 ments, 162–63
 mozilla.org, 3
 Mozilla-specific events, 205–6
 multicolumn listbox, 439
 multifact queries, 505–7
 multifragment clipboard copying,
 21–22
 <multipanel> tag, 286
 <multipanelset> tag, 286, 287
 multiple panels, displaying,
 282–87
 multiple <rule> tags, 539
 multiple selection, 226–27
 multiple tags, displaying, 47–48
 multiplexed value semantics, 149
 multirecord (or record set) dis-
 plays, 436
 multiwindow development, 37–
 38
 <mytag> tag, 308

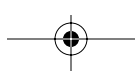
N

names, assigning, 712–14
 namespace, 386, 389
 namespace strings, 83
 naming conventions, 145, 619–21
 native installs, 708, 710, 711
 native objects, 156–57
 native themes, 126–28, 472
 <nativescrollbar> tag, 114,
 134, 275
 Nautilus, 7
 navigation, 265–93
 of forms, 243
 NoteTaker toolbars and tabs,
 289–92
 problems with, 292–93





- style options, 289
- systems for, 266–70
 - accessibility, 269–70
 - focus ring, 268–69
 - menu system, 269
 - recipe, 267
 - visual, 267
- navigation order, 268
- navigation widgets, 270–89
 - displaying multiple panels, 282–87
 - menu bars, 278–80
 - scrolling, 271–75
 - special-purpose, 288–89
 - statusbars, 280–81
 - style options, 289
 - title bars, 281
 - toolbars, 275–78
- Navigator*, 3, 4
- Navigator windows, versus
 - Chrome windows, 333
- near-zero validation, 76–77
- Necko, 8
- Netscape 7.0, 5
- Netscape Client Customization Kit, 711
- Netscape Communicator Web application suite, 3
- Netscape Navigator, 3
- Netscape Portable Runtime, 621
- network protocols, 633–36
- `newFileURI()` method, 629
- `newProxyInfo()` method, 634
- NGLayout, 8
- no-content XPI, shorthand for, 722
- Node object, 168
- nonattributes, 590–91
 - `applyauthorsheets`, 591
 - `applyauthorstyles`, 591
 - `applybindingsheet`, 591
 - `attachto` tag, 591
 - `charcode`, 591
 - `includes`, 591
 - `styleexplicitcontent`, 591
- nonstandard event behavior, 204
- nontags, 289, 344, 458
 - `<element>`, 591
 - `<fontpicker>`, 289
 - `<html>`, 344
 - `<menubutton>`, 120
 - `<object>`, 382
 - `<outliner>`, 458
 - `<package>`, 344
 - `<predicate>`, 382
 - `<script>`, 591
 - `<spinbuttons>`, 121
 - `<spinner>`, 120–21
 - `<Statement>`, 382
 - `<subject>`, 382
 - `<titlebar>`, 289
 - `<titlebutton>`, 120
 - `<treebody>`, 458
 - `<treecaption>`, 458
 - `<treecolpicker>`, 458
 - `<treefoot>`, 458
 - `<treehead>`, 458
 - `<treeicon>`, 458
 - `<treeindentation>`, 458
- nonattributes and, 590–91
- normalization, 370
- note-data sounds, 401
- `<notelacer>` tag, 598–613
 - adding XBL content, 602–5
 - adding XBL functionality, 605–10
 - designing widget interfaces, 599–601
 - integrating the binding, 611–13
- NoteTaker, 4, 34–35
 - adding text, 95–96
 - bundling up, 744–51
 - creating support files and scripts, 745–49
 - final bundling, 749–50
 - release preparation, 744–45
 - buttons and themes, 130–33
 - command design, 317–22
 - completing registration, 74–75
 - configuring and using
 - locales, 97–99
 - custom RDF tree views and data sources, 700
 - data made live, 546–59
 - coordinating and refreshing template content, 556–59
 - creating test data, 546–49
 - extended syntax template for listbox, 552–53
 - extended syntax template for tree, 553–56
 - simple templates for form elements, 549–52
 - data models, 398–407
 - dialogs, 349–56
 - dynamic content, 186–93
 - Edit dialog box, 186
 - events and forms, 258–61
 - keywords pane, 473–93
 - data-driven listboxes and trees, 481–93
 - laying out listbox and tree, 473–76
 - systematic use of event handlers, 476–80
 - layout design process, 71–74
 - `<notelacer>` tag, 598–613
 - adding XBL content, 602–5
 - adding XBL functionality, 605–10
 - designing widget interfaces, 599–601
 - integrating the binding, 611–13
 - reading string bundles via XPCOM, 190–93
 - saving and loading notes, 677–700
 - data source design, 677–78
 - data source setup, 678–80
 - dynamically allocating data sources to templates, 680–84
 - enhancing commands to process RDF content, 689–700
 - moving user-entered data into RDF, 688–89
 - scripted RDF queries using XPCOM interfaces, 685–88
 - scripting deck via the AOM and XBL, 189–90
 - scripting deck via the DOM, 186188
 - toolbars, 429–32
 - tabs and, 289–92
 - user input, 230–34
 - window, 133
- `notetaker-delete` command, 692
- `notetaker-save` command, 692
- NPA (not perfectly accurate) diagram, 2
- `nsICache` interface, 662
- `nsICmdLineHandler` interface, 624
- `nsIDirectoryService` interface, 663
- `nsIDirectoryServiceProvider` interface, 663





nsIFile interface, 628, 629
 nsIFilePicker interface, 339
 nsIFilePicker object, 630
 nsIIOService interface, 645, 646
 nsIPipe interface, 632
 nsIProcess init () method, 636
 nsIProcess interface, 628
 nsIProgressEventSink interface, 657
 nsIPromptService interface, 338
 nsIProperties interface, 664
 nsIProperties method get (), 664
 nsIRDFDataSource interface, 647, 650
 nsIRDFLiteral interface, 647
 nsIRDFRemoteDatasource interface, 650
 nsIRDFResource interface, 647
 nsIRequestObserver interface, 657–58
 nsISOAPPParameter interface, 661
 nsIStringBundle, 191
 nsIStringBundleService, 191
 nsIThreadXPCOM interface, 632
 nsITreeBuilderObserver interface, 544
 nsIWebNavigation interface, 336
 nsIWebProgress interface, 657
 nsIWebProgressListener interface, 657
 nsIXULChromeRegistry interface, 420
 Null type, 144
 number literals, 144
 Number value, 144

O

object broker, 14
 object constructor, 154–55
 object directory, 14
 object discovery service, 14
 object methods, 153
 Object objects, 156
 <object> tag, 382
 Object type, 145
 Objective-C, 139
 object-oriented (OO) inheritance, 161

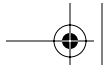
object-oriented (OO) libraries, 104
 object-oriented (OO) programming, 9
 objects, 152–53, 622
 built-in, 157–58
 creation alternatives, 178–80
 finding properties, for forms and menus, 263
 host, 155–56
 interface, 569
 name service, 14
 native, 156–57
 plain, 153–55
 observe () method, 209
 <observe> tag, 111
 Observer design pattern, 210
 observer.observe (), 209
 observers, 210–16, 217
 differences between listeners and, 210
 JavaScript, 214–15
 XPCOM, 215–16
 XUL, 211–14
 <observes> tag, 66, 211, 212, 213
 OEone, 6
 onclick handler, 66, 187, 298, 356
 oncommand attribute, 222, 308
 one frame, one tag rule, 52
 onevent, 66
 onEvent () method, 312
 open () method, 357, 647
 Open Source, 3, 24, 26
 example of, 138
 “open window” instruction, 37
 open-closed principle, 9–10
 OpenOffice, 91
 Opera Web browser, 16
 operator precedence, 158
 Optimoz project, 229
 Oracle’s SQL*Forms, 138
 ordinal attribute, 50
 ordinary menus, 327, 328
 ordinary <window> windows, 326–27
 orient attribute, 49
 out () method, 164
 <outliner> tag, 458
 overflow:scroll, 89
 overlay databases
 changing, 419–20
 reading, 417–18
 overlay merging system, 429

<overlay> tag, 413, 415
 overlayinfo database, 416, 418
 overlays, 326, 413–24
 discovery, 415–16
 bottom up method, 416, 417
 changing database, 419–20
 reading database, 417–18
 top down method, 416–17
 JavaScript, 413
 merging process, 420–24
 stylesheet, 413
 tags, 414
 <overlay>, 415
 <overlaytarget>, 415
 <?xul-overlay?>, 414
 tips for, 432–33
 <overlaytarget> tag, 415

P

pack attribute, 49
 packages, 412, 622
 name for, 713
 <package> tag, 344
 <page> tag, 342, 344
 panels
 defined, 340
 displaying multiple, 282–87
 panes. *See also* windows
 defined, 340
 embedding documents in, 340–44
 Keywords, 473–93
 <parameter> tag, 585, 586
 parseFloat () function, 158
 parseFloat () method, 160
 parseInt () function, 158
 parseInt () method, 160
 parser, 640
 parseType attribute, 388
 partial-fact problem, 379
 patchEvent () method, 310
 Penzilla, 6
 performInstall () method, 719, 721
 Perl, 24, 138, 178, 185, 332, 416
 persist attribute, 66
 persisting window state, 428–29
 Personal Information Managers (PIMs), 28
 Personal Toolbar, 515
 PHP language, 138, 139
 piggy-back install, 710
 placeDesktop () method, 607





placeNote() method, 607
 placePage() method, 607
 plain objects, 153–55
 platform configuration, 662–77
 cache control, 662–63
 file system directory, 663–70
 preferences, 670
 security, 670–76
 user profiles, 677
 platform install, 708
 platform-specific names, 712
 plugins, 180
 file system aliases, 669–70
 policed security, 673–76
 pool, 160
 popup content, 327–29
 context menus, 329
 defining, 327–28
 flyover help, 328
 tooltips, 328–29
 popup menus, 250
 popualign attribute, 329
 popuanchor attribute, 329
 <popupgroup> tag, 328
 <popupset> tag, 327–28
 porkjockeys, 8
 portability, 22
 position attribute, 47, 255, 423
 PostScript language, 565
 predicates="object" attribute, 527
 predicate/property tags, 388–90
 predicates, 369–72, 389
 <predicate> tag, 382
 pref() method, 185
 preferences, 670
 prefetching, 662
 prefix characters, 619–21
 prepare() function, 720
 primary binding, 570
 print_data() function, 201, 202
 PrintingPrompt, 340
 processes, 636–38
 produce(), 212
 Producer design pattern, 20
 profile, 121–29
 file system aliases, 668–69
 <progressmeter> tag, 288
 Prolog, 372, 403
 prompt(), 338, 719
 Prop, 675
 properties dialog boxes, 437
 <property> attribute, 606
 property files, 86–87

property names, built-in, 470–71
 property set, 535–36
 <property> tag, 583–85
 prototype, 622
 chains, 161–62
 development, 68
 object, 155
 pseudo-class, 54
 purposes, stylesheet, 413
 Python, 138, 332

Q

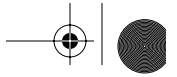
Qt widget set, 21, 104
 queries
 common patterns, 534–39
 containers, 534
 illegal, 539
 property set, 535–36
 single-fact, 535
 singleton solution, 536–37
 tree, 537–39
 unions, 539
 multifact, 505–7
 RDF, 501–12
 scripted, using XPCOM
 interfaces, 685–88
 recursive, 511
 single-fact, 503–5, 535
 stored, 511
 query lists, 511–12
 query solution strategy, 507–11
 QueryInterface() method, 272
 quick launch, 20

R

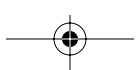
r=ada, 8
 <radio> tag, 116, 120, 246
 <radiogroup> tag, 245, 246
 Rapid application development (RAD) environment, 23–25
 COTS software, 25
 destructured testing, 25
 vertical solutions, 24–25
 visual prototypes, 24
 Rapid application development (RAD) projects, 27–28
 chrome directories and
 address book, 28–33
 command line options, 28, 29
 installation, 27–28

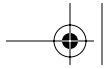
 NoteTaker preparation, 34–35
 RDF. *See* Resource Description Framework (RDF)
 <RDF> tag, 382, 385–86
 read_data() function, 201, 202
 read_info(), 201
 read-only documents, 138
 really_half_it() function, 300
 rebuild() method, 484, 545, 558, 683
 recipe navigation, 267
 recursive queries, 511
 redirection (>), 638–39
 ref property, 542
 Refresh() method, 541, 683
 refresh_dialog(), 559
 refresh_toolbar() function, 557, 558, 680, 682, 683
 RegExp object, 157
 registerChrome(), 749
 registries, 725
 registry application name, 713
 regular expressions, 152
 reification, 376–77, 395
 related AOM and XPCOM objects, 429
 related keywords, 399, 402
 relational data, 369
 remote deployment, 706, 712–25
 assigning names, 712–14
 assigning versions, 712–14
 remote install case, 711
 remotely sourced applications, 21
 removeBinding() method, 597
 removeChild() tag, 514
 removeelement, 423
 removeEventListener(), 207
 RENDER X11 extension, 91
 rendering strategies
 Mark I, 16
 Mark Ib, 16
 Mark II, 16
 Mark III, 16
 replace() method, 152, 261
 report() function, 194–95
 resizable, 335
 resizeafter, 283–84
 resizebefore, 283–84
 <resizer> tag, 118–19, 280
 <resolve> method, 557
 resource attribute, 390





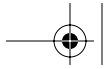
- Resource Description Framework (RDF), 2, 8–9, 11, 15, 20, 361–409
 - <Alt> tag, 382, 383, 390
 - <Bag> tag, 382, 383, 390
 - data models, 398–407
 - data sources, 641
 - <Description> tag, 382, 386–88
 - document in, 446
 - dumping, 407–8
 - enhancing commands to process content, 689–700
 - examples, 393–98
 - application areas, 397–98
 - Download Manager, 393–94, 396–97
 - MIME types, 396–97
 - using URNs for plain data, 394–96
 - facts
 - about facts, 375–78
 - concrete, 378–79
 - versus data structures, 366–69
 - ground, 378–79
 - ways to organize, 372–75
 - learning strategies for, 365–66
 - tag, 382, 383, 390
 - movement of user-entered data into, 688–89
 - Mozilla uses of, 364–65
 - <object> tag, 382
 - <predicate> tag, 382
 - predicate/property tags, 388–90
 - predicates and tuples, 369–72
 - queries, filters, and ground facts, 378–79
 - queries and unification, 501–12
 - <RDF> tag, 382, 385–86
 - scripted queries using XPCOM interfaces, 685–88
 - <Seq> tag, 382, 383, 390
 - <Statement> tag, 382
 - <subject> tag, 382
 - syntax, 379–81
 - containers, 383–84
 - identifiers, 384–85
 - tags, 381–82
 - terminology, 363–64, 382
 - resource property, 542
 - <resources> tag, 575–76
 - revision number, 714
 - rewindFocus() method, 316
 - Rhino, 141
 - root, 712
 - <row> tag, 62, 63–64
 - rowCountChanged() method, 492
 - rubber band selection, 226
 - <rule> tag, 526–29
 - extended syntax for, 529
 - simple syntax for, 527–29
 - <rules> tag, 517
 - run() function, 201, 636
 - run-time errors and exceptions, 150
- S**
- Same Origin test, 671
 - saving and loading Notetaker notes, 677–700
 - Sawfish, 18, 118
 - schedule() function, 201
 - schedule_folders() function, 720
 - scope, 159–60
 - scope chain, 159–60
 - script diagnosis, 193–95
 - <script src=> tags, 355
 - <script> tag, 68, 139, 140, 591
 - scriptable interfaces, 164–86
 - AOMs and BOMs, 173–75
 - interface origins, 165
 - island interfaces, 185–86
 - scripting libraries, 181–84
 - W3C DOM standards, 166–73
 - DOM 0, 166
 - DOM 1, 167–69
 - DOM 2, 169–71
 - DOM 3, 171–72
 - DOM compatibility ticks, 172–73
 - XBL bindings and, 184
 - XPCOM and XPCConnect, 175–81
 - scripting, 137–95, 203, 541–45, 716–21. *See also* ECMAScript Edition 3; scriptable interfaces
 - AOM template objects, 542–43
 - builders, 543–44
 - delegates, 544–45
 - views, 544
 - common scripting tasks, 545
 - debugging, 193–95
 - <deck> via the AOM and XBL, 189–90
 - general-purpose, 623–38
 - command line arguments, 623–24
 - data structures and algorithms, 624–25
 - databases, 625–27
 - environment, 628
 - files, 628–32
 - folders, 628–32
 - interrupts, 632
 - network protocols, 633–36
 - processes, 636–38
 - signals, 632
 - threads, 636–38
 - guidelines for dynamic templates, 541
 - language enhancements, 162–64
 - reading string bundles via XPCOM, 190–93
 - role of JavaScript as language, 139–40
 - standards, 140–41
 - styles via DOM, 188–89
 - Web, 624, 652, 655–62
 - downloading files, 657–58
 - file types, 658
 - MIME types, 658
 - uploading and posting files, 659
 - URIs, 656–57
 - URLs, 656–57
 - URNs, 656–57
 - Web protocol stack
 - objects, 659–62
 - XSLT batch processing, 662
 - XBL bindings and, 595–98
 - XPCOM template objects, 542–43
 - <scrollbar> tag, 274–75, 341
 - scrollbar thumb, 116
 - <scrollbarbutton> tag, 109, 114–15
 - scrollbars, 282
 - <scrollbox> tag, 89, 174, 254, 272–73, 341
 - scrolling widgets, 271–75
 - Seamonkey, 8
 - security, 23, 670–76
 - certified, 672–73
 - digital, 707





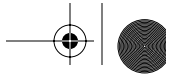
- install, 710
 - language enhancements, 163–64
 - policied, 673–76
 - restrictions, 676
 - trusted, 671
 - Web Safe, 671
 - select() kernel call, 199
 - SELECT statement, 502
 - selectedIndex property, 60–61, 190
 - selector, 43
 - seltype attribute, 443
 - seminal data, 462–63
 - send() method, 250
 - <separator> tag, 57
 - <Seq> tag, 75, 382, 383
 - serializer, 640
 - session, 639
 - setAttribute(), 274, 351
 - setAttribute() DOM property, 188
 - setCard() method, 190
 - setFromEvent() method, 601, 609, 610
 - setInterval(), 210, 637
 - setInterval() calls, 202
 - setInterval() method, 160, 164
 - setEditable() view method, 467
 - <setter> tag, 584, 585
 - setTimeout(), 208, 209, 210, 249, 250, 351, 637
 - setTimeout() calls, 202, 683
 - setTimeout() event, 208
 - setTimeout() method, 160, 693
 - shadowing, 161
 - signals, 632
 - simple merging, 420
 - simple template syntax, 517–18
 - simple template variables, 518–19
 - single tag, displaying, 46–47
 - single-fact queries, 503–5, 535
 - singleton solution, 536–37
 - sinks, 639–40
 - specialized, 640
 - sizemode, 326
 - skins, 88, 108, 121–29
 - diagnosing, 133–35
 - directories, 19, 125
 - stylesheet hierarchy, 125–26
 - <slider>, 121
 - <smiley> tag, 564
 - SOAP, 15, 655, 660–61
 - socket, creating, 634–35
 - SOCKS-enabled sockets, 635
 - software platform, 11
 - sort support, for <template> siblings, 525
 - sorting, 446
 - source id, 420
 - sources, 639–40
 - specialized, 640
 - <spacer> tag, 56–57, 285
 - spacers, 56
 - tag, 43, 89
 - tag, 352
 - special URIs and namespaces, 519–20
 - specialized sources and sinks, 640
 - special-purpose widgets, 288–89
 - special-purpose XPCOM dialog boxes, 339–40
 - SpiderMonkey, 140, 160, 185
 - enhancements, 162–63
 - <spinbuttons> tag, 121
 - <spinner> tag, 120–21
 - split(), 152
 - split widget design, 108–9
 - <splitter> tag, 116–17, 282–84, 287, 454, 460
 - spreadsheet, 437
 - spring, 78
 - <spring>, 58
 - SQL (Structured Query Language), 366, 398
 - three-table join in, 507
 - sr=ada, 8
 - stack, 160
 - <stack>, 58–60, 64
 - stacks and garbage collection, 160
 - statedataresource attribute, 524
 - <Statement> tag, 382
 - statement block, 159
 - static content, 81–101
 - adding text, 95–96
 - comparing XUL and HTML, 82–83
 - configuring and using locales, 97–99
 - debugging, 99–101
 - DOM Inspector in debugging, 99–101
 - font systems, 89–92
 - NoteTaker, 95–99
 - style options, 92–95
 - XUL content tags, 84–89
 - border decorations, 89
 - images, 87–89
 - text and strings, 84–87
 - <statusbar> tag, 280–81
 - <statusbarpanel> tag, 281
 - statusbars, 280–81
 - stdio libraries, 139
 - stored queries, 511
 - stream, 618, 638–44
 - stream content conversions, 643–44
 - stream objects, 618
 - string bundles, reading, via XPCOM, 190–93
 - <string> tag, 86–87
 - String type, 145
 - <stringbundle> tag, 86–87, 211
 - <stringbundleset> tag, 87, 211
 - structural options, 650
 - strut, 78
 - stub install, 711
 - style, 66
 - properties and extensions, 52–54
 - styleexplicitcontent attribute, 591
 - <stylesheet> tag, 575–76
 - stylesheets
 - diagnostic, 262
 - hierarchy, 125–26
 - overlays, 413
 - purposes, 413
 - theme, 108
 - <subject> tag, 382
 - sub-pixel accuracy, 90
 - support files and scripts, creating, 745–49
 - supportsCommand() method, 301, 304, 694
 - suppressonselect attribute, 443
 - SVG (vector graphics content), 8, 15
 - Mozilla support for, 17
 - Swing, 104
 - switch statement, 150
 - Sybase's PowerBuilder, 138
 - synchronous submission, 249
- ## T
- Tab key, 108, 268
 - <tab> tag, 244, 285–86





- <tabbox> tag, 58, 61, 284–85, 291, 352
 - <tabbrowser> AOM object, 336
 - <tabbrowser> tag, 329, 336, 343–44, 565
 - <tabpanel> tag, 284, 285, 286
 - <tabs> tag, 284, 285–86
 - tags. *See also specific tag names*
 - box layout, 54–64
 - displaying multiple, 47–48
 - displaying single, 46–47
 - names, 82, 108
 - Talkback technology, 5
 - target, 715–16
 - target document, 568
 - target sites, 227
 - targets attribute, 309
 - Tcl/Tk, 24, 138, 565
 - <template> siblings, sort support for, 525
 - <template> tag, 111–12, 501, 517, 526
 - templates, 497–561
 - building up, 560–61
 - common query patterns, 534–39
 - containers, 534
 - illegal, 539
 - property set, 535–36
 - single-fact, 535
 - singleton solution, 536–37
 - tree, 537–39
 - unions, 539
 - construction, 517–34
 - <action> tag, 532–34
 - base tag, 520–25
 - <bindings> tag, 531–32
 - <conditions> tag, 529–31
 - <coordinations> tag, 529–31
 - <member> tag, 531
 - <rule> tag, 526–29
 - special XUL names, 518–20
 - <template> tag, 526
 - data source issue, 516
 - dynamically allocating data sources to, 680–84
 - example, 499–501
 - extended syntax for <listbox>, 552–53
 - extended syntax for <tree>, 553–56
 - extended variables, 518
 - guidelines for dynamic, 541
 - JavaScript Access, 515–16
 - lifecycle, 539–40
 - NoteTaker data made live, 546–59
 - coordinating and refreshing template content, 556–59
 - creating test data, 547–49
 - extended syntax template for <listbox> tag, 552–53
 - extended syntax template for <tree>, 553–56
 - simple templates to form elements, 549–52
 - RDF queries and unification, 501–12
 - rebuilding, 514
 - scripting, 541–45
 - AOM and XPCOM template objects, 542–43
 - builders, 543–44
 - common tasks, 545
 - delegates, 544–45
 - guidelines for dynamic templates, 541
 - views, 544
 - simple, for form elements, 549–52
 - simple variables, 518–19
 - style options, 546
 - survival guide, 559–61
 - XUL, 13, 650
 - XUL content generation, 512–15
- term, 367
 - test-debug-fix loop, 76
 - <text>, 96
 - text, boilerplate, 95
 - text grids, 436–38
 - text input tags, 436
 - text name, 712–13
 - <text> tag, 84–85
 - <textbook> tag, 437, 680
 - <textbox> tag, 246–47, 256, 437, 682
 - TextEvents, 207
 - <textnode> tag, 534
 - The tree, 8
 - theme engine, 121–22
 - theme stylesheet, 108
 - themes, 19, 88, 104, 121–29
 - native, 126–28, 472
 - NoteTaker, 130–33
 - third-generation procedural languages, 139, 151
 - threads, 636–38
 - <thumb> tag, 109, 114
 - Thunderbird, 3
 - timed events, 208–10, 217
 - title attribute, 331
 - title bars, 281
 - <titlebar> tag, 289, 335
 - <titledbutton> tag, 120
 - toggle() method, 583
 - toggleOpenState() method, 490–91
 - toggleOpenView() method, 492
 - tonumber() method, 158
 - toolbar action() function, 689
 - <toolbar> tag, 277
 - <toolbarbutton> tag, 109, 112–13, 186–87, 268, 277–78, 356
 - <toolbargrippy> tag, 117–18, 277
 - <toolbaritem> tag, 277–78
 - toolbars, 112, 275–78
 - locked, 275–76
 - NoteTaker, 289–92
 - <toolbarseparator> tag, 57, 277–78
 - <toolbox> tag, 276–77
 - toolkit.jar, 19, 47
 - <tooltip> tag, 112, 328
 - tooltips, 326, 328–29
 - top down method, 413, 416–17
 - toString() method, 158, 159, 215–16
 - traditional variable scoping, 159
 - transports, 639, 644–45
 - tree queries, 537–39
 - <tree> tag, 50, 61, 64, 226, 228, 436, 454
 - attributes specific to, 524
 - debugging, 493–94
 - extended syntax template, 553–56
 - laying out, 473–76
 - making work, 493–94
 - style options for, 469–72
 - <tree>-based templates, content for, 533–34
 - <treebody> tag, 458
 - <treecaption> tag, 458
 - <treecell> tag, 457–58
 - <treecildren> tag, 456
 - <treecol> tag, 455–56
 - <treecolgroup> tag, 458
 - <treecolpicker> tag, 458





`<treecols>` tag, 454
`<treefoot>` tag, 458
`<treehead>` tag, 458
`<treeicon>` tag, 458
`<treeindentation>` tag, 458
`<treeitem>` tag, 456, 530
tree-pseudo-classes, 469–70
`<treerow>` tag, 457, 458
trees, 224, 449–68

- AOM and XPCOM interfaces, 460–62
 - builders, 463–64
 - seminal data, 462–63
 - views, 464–68
- construction, 451–53
- custom views, 484–93
- data-driven, 481–83
- native theme support, 472
- nontags, 458
- RDF and sorting, 458
- `<tree>` tag, 454
- `<treecell>` tag, 457–58
- `<treechildren>` tag, 456
- `<treecol>` tag, 455–56
- `<treecolpicker>` tag, 458
- `<treecols>` tag, 454
- `<treeitem>` tag, 456
- `<treerow>` tag, 457, 458
- `<treeseparator>` tag, 457
- user interactions, 458–60
- visual appearance, 450–51

`<treeseparator>` tag, 58, 457
`<tree>` tag, 174
trigger script, 716
`<triple>` tag, 530
trivial protocols, 647
trusted security, 671
try block, 150, 151
tuples, 367, 368–72, 504–5
type attribute, 108, 591
type convention, 158–59
Type Null, 143
Type Object, 143
type-ahead find, 20, 223
typeof operator, 143

U

UDDI, 655
Unassert() method, 694, 701
Undefined type, 143
Unicode characters, 145
Unicode standard, 10
Unicode string, 712
unification, 502, 503

uninstall() method, 718, 721
unions, 539
UNIX X11/GTK behavior, 37–38
update() function, 312–13
update_all() function, 309
UpdateCommands() method, 306, 308, 309, 310, 313–14
`<updater>` tag, 312, 314
URI (Universal Resource Identifier), 385, 395, 656–57
uri attribute, 66, 530
url(), 123
URL (Uniform Resource Locator), 2, 395, 656–57, 717

- chrome, 19–20
- using, 631–32

URN (Uniform Resource Name), 395–97, 656–57

- examples, MIME types, 396–97
- using, for plain data, 394–96

user input, 600–601
user interactions, 446–47
user profiles, 677
user-entered data, movement with RDF, 688–89
user-interface design, 82–83
user_pref method, 185
UTF-8-encoded characters, 142

V

value attribute, 445
var keyword, 146
variable interpolation, 519
variables, 145–46, 153, 503

- names for, 145–46

`<vbox>` tag, 54, 78–79, 105–6, 234, 422, 501
Venkman, 35
Venkman Chrome, 713
version information, 99
versions, assigning, 712–14
vertical `<box>`, 48
views, 464–68, 544
visibility, 47
Visual Basic, 138
Visual Basic for Applications (VBA), 140
visual formatting model, 45
visual interface, 599
visual navigation, 267
VNC server, 91

W

W3C, 17
W3C standards, 20, 568

- DOM, 12, 166–73
- RDF, 365
- XML, 362

WAPScript language, 140
warnings, 76
watch() method, 194
watchpoints, 194–95
Web browsers, 9, 436, 652
Web protocol stack objects, 659–62
Web Safe security, 671
Web scripting, 624, 652, 655–62

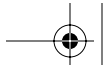
- downloading files, 657–58
- file types, 658
- MIME types, 658
- uploading and posting files, 659
- URIs, 656–57
- URLs, 656–57
- URNs, 656–57
- Web protocol stack objects, 659–62
- XSLT batch processing, 662

Web standards, support for, 16–17
Web-based systems, 437–38
Web-side objects, 731–32
whitespace, 407
widgets

- defined, 104
- designing interfaces, 599–601
- navigation, 270–89
- origins of XUL, 107–9
- picking apart complex, 262–63
- reflection of command state, 312–14
- scrolling, 271–75
- special-purpose, 288–89
- split design, 108–9

Win32, 21, 104, 107
WinAmp, 122
Window object, 174
`<window>` tag, 344
window.Components.Constructor() method, 182
window.open() method, 326, 327, 331, 333–35
window.openDialog() method, 335–36





windows, 325–59
 diagnostic, 357–59
 dialog boxes, 329–32
 embedding documents in panes, 340–44
 JavaScript creation of, 332–40
 managing existing, 346–47
 mixing documents, 344–45
 Navigator versus Chrome, 333
 NoteTaker dialogs, 349–56
 ordinary `<window>`, 326–27
 popup content, 327–29
 resizing, 229
 style options, 347–49
 Windows registry key, 712
`window.setTimeout()`, 637
`windowtype`, 326–27
`WinProfile` object, 741
`WinReg` object, 741
`WinZip`, 28
 with statement, 151
`<wizard>` tag, 282, 728, 729–30
`<wizardpage>` tag, 728, 730–31
 wizards, XUL, 728–31
`<wizards>` tag, 58, 61, 332
 World Wide Web Consortium, 8
 standards for, 165
`write()` method, 357
 WSDL (Web Services Description Language), 15, 655, 661, 670
 standards for, 567
 WSH (Windows Scripting Host) technology, 567

X

x, 185
 X11 Protocol, 91
 XBL (XML Binding Language), 10, 17
 adding content, 602–605
 adding functionality, 605–10
 definition, 108
 documents, 15
 keyboard maps, 220–21
 scripting `<decks>` via AOM and, 189–90
 XBL bindings, 13, 109, 112, 165, 184, 563–615, 676
 combining multiple, 591–94
 inner content bindings, 594
 input façade bindings, 593–94
 simple inheritance with `extends=`, 591–92
 zero content bindings, 592–93
 component framework, 572
 constructing, 572–91
`<binding>` tag, 574–75
`<bindings>` tag, 574
 bound tags and `-moz-binding`, 573–74
`<content>`, 576–81
`<handlers>` tag, 588–90
`<implementation>` tag, 581–88
 nontags and nonattributes, 590–91
`<resources>`, 575–76
 default actions, 568–69
 diagnosis, 614–15
 example, 565–67
 object-like features, 569–72
 processing, 594–95
 relationship with DOM hierarchies, 568–69
 scripting, 595–98
 standards, 567–68
 style options, 598
`<notelacer>` tag, 598–613
 XBL markup languages, 165
`xbl:inherits=`, merging attributes with, 576–78
 XHTML, 8–9, 15
 XLink, 15, 17
 XML, 15
 interface, 601
 namespace, 15, 33–34, 326
 schema, 15, 655
 software support for standards, 15
 syntax, 365
 technologies, 2
 XML attribute, 206
 XML documents, 9, 10, 15–16
 mixing types, 344–45
 underlying format of, 9
 XML environment, 8–11
 hierarchical thinking, 10–11
 open-closed principle, 9–10
 underlying format, 9
 universality, 10
 XML Query, 378
`XMLHttpRequest` object, 249–50
 XML-like technologies, 2
`xmlns` attribute, 415

XML-RPC, 655
 XP, 7
 XPath, 378
 XPCOM (Cross Platform Component Object Model), 11, 14, 165, 175–81, 198, 208–9
 creation of dialog windows from JavaScript using, 332
 direct access to, 12
 interfaces, 447–49, 460–62
 Java, 180–81
 object creation alternatives, 178–80
 plugs and DOM, 180
 reading string bundles via, 190–93
 XPCOM broadcasters, 215–16
 XPCOM components, 13–14, 176–78, 332
 commands and, 314–16
 finding, 176–78
 using, 178
 XPCOM file system aliases, 665–67
 XPCOM interfaces, scripted RDF queries using, 685–88
 XPCOM objects, 23, 429, 617–703, 743
 data transfer, 638–52
 channels, 645–47
 content processing, 638–69
 sinks, 638–40
 sources, 638–40, 647–52
 streams, 640–44
 transports, 644–45
 foreign-type systems, 622–23
 general-purpose scripting, 623–38
 command line arguments, 623–24
 data structures and algorithms, 624–25
 databases, 625–27
 environment, 628
 files, 628–32
 folders, 628–32
 interrupts, 632
 network protocols, 633–36
 processes, 636–38
 signals, 632
 threads, 636–38



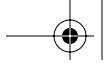


- modular programming, 621–22
- naming conventions, 619–21
- platform configuration, 662–77
 - cache control, 662–63
 - file system directory, 663–70
 - preferences, 670
 - security, 670–76
 - user profiles, 677
- saving and loading Notetaker notes, 677–700
- Web scripting, 652, 655–62
 - downloading files, 657–58
 - file types, 658
 - MIME types, 658
 - uploading and posting files, 659
 - URIs, 656–57
 - URLs, 656–57
 - URNs, 656–57
 - Web protocol stack
 - objects, 659–62
 - XSLT batch processing, 662
- XPCOM observers, 215–16
- XPCOM template objects, 542–43
- XPCoconnect, 12, 165, 175–76, 198
- xpcshe11, 12, 185
- XPI files, 721–22
- XPIDL (Cross Platform IDL), 14
 - files in, 619
- XPInstall, 32, 419, 706–7, 712
 - technology, 21
- XPInstall File object, 739–41
- XPInstall Install Object, 733–36
- XPInstall WinReg object, 742
- XPInstall-side objects, 732–42
- Xresources, 129
- X-server, 91
- XSL-FO (XSL Formatting Objects), 15
- XSLT (Extensible Stylesheet Language Transformations), 15
 - batch processing, 662
- XUL (XML User-interface Language), 10, 11, 33, 42, 242–43, 416
 - applications, 307–10
 - basic structure, 42
 - broadcasters, 211–14
 - buttons, 109–21
 - <autorepeatbutton> tag, 113–14
 - <button> tag, 110–12
 - <dropmarker> tag, 115–16
 - <gripper> tag, 119
 - <grippy> tag, 116–17
 - grippys, 116–19
 - labels and label, 119–20
 - <menubutton> tag, 120
 - <menupopup> tag, 115
 - <resizer> tag, 118–19
 - <scrollbarbutton> tag, 114–15
 - <slider> tag, 121
 - <spinbuttons> tag, 121
 - <spinner> tag, 120–21
 - <thumb> tag, 114
 - <titledbutton> tag, 120
 - <toolbarbutton> tag, 113–14
 - <toolbargrippy> tag, 117–18
 - variations, 115–16
 - comparison of HTML and, 82–83
 - content generation, 512–15
 - content tags, 84–89
 - border decorations, 89
 - images and <image>, 87–89
 - text and strings, 84–87
 - detecting bad, 75–79
 - difference between HTML and, 46
 - displaying multiple tags in, 47–48
 - documents, 15
 - mixing, 345
 - form tags, 245–47
 - forms
 - common properties of, 243–44
 - comparison with HTML forms, 240–42
 - HTML submission of, 248
 - general-purpose attributes, 65–66
 - good coding practices for, 66–69
 - layout, 41–79
 - box layout tags, 54–64
 - support for event handlers, 207–8
 - template, 13, 650
 - user interfaces, 296
 - using commands via, 307–10
 - widgets, 269, 436
 - description language for, 22
 - origins of, 107–9
 - window object, 174–75
 - wizards, 728–31
- XUL accesskey attribute, 223
- XUL <command> tag, 303
- XUL hidden attribute, 189
- XUL layout, 41–79
 - box improvement, 64–65
 - box layout tags, 54–64
 - boxes, 54–55
 - flex and spacers, 55–58
 - grids, 61–64
 - stacks and decks, 58–61
 - boxes in, 43–44, 54–55
 - <caption>, 64–65
 - detecting bad XUL, 75–79
 - general-purpose attributes, 65–66
 - good coding practices, 66–69
 - <groupbox>, 64–65
 - layout design process, 71–74
 - principles of, 44–54
 - common box layout attributes, 49–51
 - displaying multiple tags, 47–48
 - displaying single tag, 46–47
 - frame and style extensions concepts, 51–54
 - style options, 69–71
- XUL observers, 211–14
- XUL tags, 45–46, 83, 207–8, 257
 - changing appearance of, 104
 - displaying single, 46
- xulcontentgenerated attribute, 521–22
- <?xul-overlay?> tag, 414, 417
- X-Window, 90–91

Z

- Zarro Boogs, 8
- Zawinski, Jamie, 2
- zero content bindings, 592–93
- ZIP archives, working on, 632
- Zope application server, 138



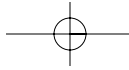


About the Author



Nigel McFarlane is a science and technology writer, analyst, and programmer. He is the author of many articles on Web, XML, JavaScript, and other technologies, and his work has appeared in periodicals such as *DevX* and *The Sydney Morning Herald*, and on *www.builder.com*. Nigel is also the author of *Instant JavaScript* and *Professional JavaScript*. He lives in Melbourne, Australia.





!"

!* ##\$% %%%#

! " " # %&' s() *

\$% & ' ' !"# \$ \$ "# # % & ' % (\$

) !" ##\$ && #&

+ " * ") * ; " +

*) \$\$ # # + % !"# \$

) !" ##\$ \$\$#

" + " + . / + " ./+ "

\$ (& ' ' & * , " - " . / (* / ! 0 / /) &

+ 0 !" ##\$ \$ #

" 0 1 & 1 0 2 + 3 4)+55 2.0 #

\$()

+ 0 -) - () * + # %\$ (,) - # \$ # . \$\$%# + +) *

) * * +

(((' () ' / + # !" ##\$ % # #

" . 6 , . 6

& *++ ,

-- # \$ # # ' . "

) 1 - + & % !" ##\$ \$ #

