

Index

Note: Page numbers preceded by BC refer to Bonus Chapters 36–58 on the CD-ROM.

Numbers

- 0 (zero), avoiding with `Date` objects, 115
- 1-12 `nodeType` property values, descriptions of, 235
- 32-bit Windows operating systems, using Internet Explorer with, 173

Symbols

- (hyphens), prohibition in JavaScript, 173
- ! (Not) Boolean operator, operand and result for, 1010
- != (does not equal) comparison operator, meaning of, 65, 1003
- !== (strictly does not equal) comparison operator, operand and result for, 1003
- # (hash mark) URL convention, property for, 498–500
- # placeholder, using with rollovers, 137
- \$ positional metacharacter, using in regular expressions, BC243
- \$1 . . . \$9 property, using with `RegExp` object, BC259–BC260
- % (modulo) connubial operator, operand and result for, 1005
- %= (modulo by value) assignment operator, example of, 1008
- && (bitwise And) operator, operands for, 1014
- &&& (And) Boolean operator, operand and result for, 1010, truth table for, 1011
- &|= (bitwise and by value) assignment operator, example of, 1008
- () (parentheses), using with regular expressions, BC244
- * (multiply) connubial operator, operand and result for, 1005
- * counting metacharacter, using in regular expressions, BC243
- *= (multiply by value) assignment operator, example of, 1008
- , (comma) operator, description of, 1018, evaluating left-to-right expressions with, 1018–1019

- . (dot) matching metacharacter, using in regular expressions, BC242
- // (comment) global statement, explanation of, 1070
- / (divide) connubial operator, operand and result for, 1005
- // (forward slashes), purpose of, 48–49, 147
- /*...*/ (comment) global statement, explanation of, 1070
- /= (divide by value) assignment operator, example of, 1008
- ? : operator, explanation of, 1019
- ? counting metacharacter, using in regular expressions, BC243
- @ (at) symbol, ensuring inclusion in e-mail addresses, 29–30
- (minus) connubial operator, operand and result for, 1005
- (minus) sign, checking for, BC266
- (decrement) connubial operator, operand and result for, 1005
- = (subtract by value) assignment operator, example of, 1008
- [...] matching metacharacter, using in regular expressions, BC242
- [] (square brackets), using in Internet Explorer, 202, using with repeat loops, 70
- [^...] matching metacharacter, using in regular expressions, BC242
- \ (backslash) pairs, examples of, 885
- ^ (bitwise XOR) operator, operands for, 1014
- ^ positional metacharacter, using in regular expressions, BC243
- ^= (bitwise XOR by value) assignment operator, example of, 1008
- _ (underscore), using with variable names, 59–60
- { } (curly braces), using, 74
- {n,} counting metacharacters, using in regular expressions, BC243
- | (bitwise Or) operator, operands for, 1014
- || (Or) Boolean operator, operand and result for, 1010, truth table for, 1011
- |= (bitwise or by value) assignment operator, example of, 1008

- ~ (bitwise Not) operator, operands for, 1014
- + (addition) operator, joining strings with, 110
- + (plus) connubial operator, operand and result for, 1005
- + (plus) operator, concatenating strings with, 64
- + (plus) sign
 - after version numbers, 157
 - joining strings with, 62
 - stringing together batches of text with, 25
- + counting metacharacter, using in regular expressions, BC243
- ++ (increment) connubial operator, operand and result for, 1005
- += (add by value) assignment operator
 - example of, 1008
 - joining strings with, 110
- +val (positive) connubial operator, operand and result for, 1005
- < (is less than) comparison operator, operand and result for, 1003
- < operator, meaning of, 65
- << (left shift) operator, operands for, 1014
- <<= (left shift by value) assignment operator, example of, 1008
- <= (is less than or equal to) comparison operator, meaning of, 65, 1003
- = (equals) operator, example of, 1008
- == (equals) comparison operator
 - meaning of, 65
 - operand and result for, 1003
 - using with parallel arrays, 77
- === (strictly equals) comparison operator, operand and result for, 1003
- > (is greater than) comparison operator, operand and result for, 1003
- > operator, meaning of, 65
- >= (is greater than or equal to) comparison operator, operand and result for, 1003
- >= (right shift by value) assignment operator, meaning of, 65, 1008
- >> (right shift) operator, operands for, 1014
- >>= (zero fill by value) assignment operator, example of, 1008
- >>> (zero fill right shift) operator, operands for, 1014
- >>>= (right shift by value) assignment operator, example of, 1008
- “ (quotes), including with element IDs, 24
- “ (quote pair), using with `String` objects, 110–113
- ‘‘ (quote pair), using with `String` objects, 110–113

A

- a element object
 - explanation of, 601
 - properties for, 600
- tag, explanation of, 527
- abbr property, using with `td` and `th` element objects, BC143
- abort () method for `XMLHttpRequest` object, explanation of, 878
- absbottom value of `align` property, description of, 486
- absmiddle value of `align` property, description of, 486
- accelerator property, explanation of, 853
- acceptCharset property for `form` object, explanation of, 654
- “Access disallowed from scripts at <URL> to documents at <URL>” error message text, explanation of, BC330
- “Access is denied” error message text, explanation of, B330–B331
- accessKey property
 - for HTML element objects, 199–201
 - using with `label` element object, BC14
- action property for `form` object, explanation of, 655
- activation and deactivation event handlers, using, 327
- Active Desktop, adding web site to, 389–390
- active elements, designating, 319
- activeElement property for `document` object, explanation of, 524
- ActiveX control, accessing Windows Media Player as, 425
- ActiveX objects, inclusion in Win32 versions of IE4, 216
- ActiveXObject global statement, explanation of, 1073–1074
- add by value (+=) assignment operator, example of, 1008
- Add () method for `Dictionary` object, description of, 1074
- add () method for `select` element object, explanation of, 730
- addBehavior () method for HTML element objects
 - description of, 259–260
 - examples of, 260–262
- add-by-value (+=) operator, joining strings with, 110
- addEventListener () method for HTML element objects
 - description of, 263–264
 - example of, 264–266
 - parameters of, 189
- addImport () method for `stylesheet` object, explanation of, 823

- addition (+) operator, joining strings with, 110
- `addRange()` method, using with `selection` object, BC47
- `addReadRequest()` method, using with `userProfile` object, BC193–BC194
- `addRule()` method for `styleSheet` object, explanation of, 823–824
- `addTotals()` method, using in order form application, BC414–BC415
- Adobe Reader, installing and running, 1110–1111
- ADO-related event object properties, descriptions of, 769
- `afterBegin` parameter
 - using with `getAdjacentText()` method, 287
 - using with `insertAdjacentElement()` method, 297
 - using with `replaceAdjacentText()` method, 314
- `afterEnd` parameter
 - using with `getAdjacentText()` method, 287
 - using with `insertAdjacentElement()` method, 297
 - using with `replaceAdjacentText()` method, 314
- Ajax (Asynchronous JavaScript and XML), significance of, 5
- AJAX outline
 - accumulating HTML for, BC431–BC434
 - OPML outliner prep for, BC427–BC428
 - setting scripted stage for, BC429–BC431
 - tooggling node expansion for, BC434–BC435
 - XML and HTML code for, BC428–BC429
 - XML specification related to, BC426
- alert dialog box
 - displaying, 416
 - example of, 85
 - explanation of, 415
- `alert()` method
 - displaying alert dialog box with, 42
 - using with `String` object, 110
- `alert()` method for window object
 - example of, 416
 - explanation of, 415–416
- `align` property
 - for `applet` object, BC227
 - for `embed` element object, BC237
 - for `h1...h6` element objects, BC9
 - for `hr` element object, BC10–BC12
 - for `iframe` element object, 486, 610–611
 - for object element object, BC231
 - for `table` element object, BC119
- `aLink` property for `body` element object, explanation of, 588
- `alinkColor` property for `document` object, explanation of, 524–527
- `all[]` property for HTML element objects, explanation of, 201–202
- `allowTransparency` property
 - for `frame` element object, 472
 - for `iframe` element object, 487
- `alpha()` filter, description and properties for, 856
- `Alpha()` filter, description and properties for, 862
- `alt` property
 - for `applet` object, BC228
 - for `area` element object, 629
 - for `img` element object, 612
 - for object element object, BC231
- `altHTML` property
 - for `applet` object, BC228
 - for object element object, BC231
- `altKey` property
 - for NN6+/Moz/Safari event object, 789–790
 - for of IE4+ event object, 767–768
- `altLeft` property for IE4+ event object, explanation of, 768
- `alwaysLowered` attribute of `open()` method, browsers for and description of, 431
- `alwaysRaised` attribute of `open()` method, browsers for and description of, 431
- `alwaysRaised` window, creating, BC358
- anchor element objects, explanation of, 600
- `anchorNode` property, using with `selection` object, BC45
- `anchorOffset` property, using with `selection` object, BC45
- `anchors[]` property for `document` object, explanation of, example of, 527–528
- And (&&) Boolean operator
 - operand and result for, 1010
 - truth table for, 1011
- API approach to plug-ins
 - building jukebox, BC313–BC317
 - initializing library for, BC311–BC312
 - invoking methods for, BC312
 - loading library for, BC311
 - sound player API objects, BC312
- API customization, using for DHTML compatibility, BC368–BC369

- API key, using with Google Maps, BC516–B517
- appName property, using with `clientInformation` and `navigator` objects, BC155
- appCore property for `window` object, explanation of, 380
- appendChild() method
 - exceptions thrown by, 991
 - explanation of, 266–268
 - role in W3C DOM Level 2, 183
- appendData() method, using with `Text` and `TextNode` objects, BC52–BC55
- applet element object, explanation of, BC226–BC227
- applet object checks, explanation of, BC349
- applets
 - calling HTML document with, BC309
 - controlling, BC290
 - data-type conversion concerns, BC302, BC305
 - faceless applets, BC298–BC302
 - limitations of, BC297
 - properties for, BC293
 - reading text files with, BC300–BC301
 - requirements for communication with scripts, BC303
 - scriptable methods for, BC293–BC297
 - source code for, BC299–BC300
 - stopping and starting, BC291–BC292
- applets[] property for `document` object, explanation of, 529
- applet-to-script communication
 - applet requirements for, BC303
 - HTML requirements for, BC304
 - role of `JSObject` class in, BC304
 - sample application, BC306–BC309
- apply() property for `function` object, explanation of, 1030–1031
- applyElement() method for HTML element objects, explanation of, 268–270
- appMinorVersion property, using with `clientInformation` and `navigator` objects, BC162–BC163
- appName property, using with `clientInformation` and `navigator` objects, BC155
- AppVersion property, using with `clientInformation` and `navigator` objects, BC155–BC158
- arc() method for `canvas` element object, explanation of, 640
- archive property
 - for `applet` object, BC228
 - using with `object` element object, BC232
- arcTo() method for `canvas` element object, explanation of, 640
- area element object, explanation of, 626–627
- areas[] property for `map` element object, explanation of, 631–633
- arguments, using with DOM object methods, 41
- arguments property for `function` object, explanation of, 1027
- arithmetic operators, using, 64
- array concatenation code sample, 957–958
- array data, accessing, 75–76
- array entities, deleting, 948
- array objects
 - methods, 955–968
 - properties of, 953–955
- array of objects, creating in planetary objects example, 1047–1048
- array syntax, using with `document.forms[]` property, 89
- `Array.join()` method code sample, 959
- `Array.reverse()` method code sample, 961–962
- arrays
 - for `area` element objects, 631
 - creating, 75
 - creating empty arrays, 946
 - creation enhancements made to, 947–948
 - definition of, 945
 - indexes in, 74
 - multidimensional arrays, 950–951
 - parallel arrays, 76–77, 948–951
 - populating, 947
 - returning number of items in, 232–233
 - use of, 946
 - using `document` objects in, 78
 - using in image precaching, 132
 - using variables with, 77
- assign() method for `location` object, explanation of, 510
- assignment operators, explanation of, 1008–1009
- assignment statement, using with `window.open()` method, 83
- Asynchronous JavaScript and XML (Ajax), significance of, 5
- at (@) symbol, ensuring inclusion in e-mail addresses, 29–30
- atEnd() method for `Enumerator` object, description of, 1075
- attachEvent() method for HTML element objects, explanation of, 270–271
- attribute HTML-related node type, description of, 180

attribute object, explanation of, 289

attribute properties for W3C DOM-compatible browsers,
list of, 289

attributes

- methods for, 564–565
- methods used with, 295–296
- for `open()` method for window object, 431
- propagating, 304–305
- removing, 310–312
- removing from elements, 275
- setting, 319–321

`attributes` attribute object property, description of, 203

`attributes` property for W3C DOM Level 2, description
of, 181

`attributes[]` property for HTML element objects,
explanation of, 203–204

aural properties, 854

authoring environment

- Mac OS X, 20–21
- reloading issues, 20
- setting up, 18–20
- window arrangement in Windows XP, 19
- Windows, 19
- workflow for, 18

`autocomplete` property for form object, explanation
of, 655

`availHeight` property, using with `screen` object,
BC188–BC189

`availLeft` property, using with `screen` object, BC189

`availTop` property, using with `screen` object, BC189

`availWidth` property, using with `screen` object,
BC188–BC189

`axis` property, using with `td` and `th` element
objects, BC143

B

`\b` matching metacharacter, using in regular
expressions, BC242

`\B` matching metacharacter, using in regular
expressions, BC242

`back()` method

- for `history` object, 515–517
- for window object, 417

`BackColor` command, parameter for and description
of, 575

background properties, 845–846

background property

- for `body` element object, 589
- using with `table` element object, BC120

backgrounds, changing for positioned elements,
BC198–BC201

backslash (\) pairs, examples of, 885

bar chart, including in Decision Helper application, BC479

`Barn()` filter, description and properties for, 862

base element object, explanation of, BC91–BC92

`BaseHref` property, using with `object` element
object, BC232

baseline value of `align` property, description of, 486

`baseURI` property

- for HTML element objects, 204
- for `object` element object, BC232

`baseURI[]` property for `document` object, explanation
of, 529

basic object model, significance of, 169–170

`BasicImage()` filter, description and properties for, 862

batch versus real-time validation, BC261–BC263,
BC272–BC273

BBedit text editor, downloading, 18

`beforeBegin` parameter

- for `getAdjacentText()` method, 287
- for `insertAdjacentElement()` method, 297
- for `replaceAdjacentText()` method, 314

`beforeEnd` parameter

- for `getAdjacentText()` method, 287
- for `insertAdjacentElement()` method, 297
- for `replaceAdjacentText()` method, 314

`beginPath()` method for `canvas` element object,
explanation of, 641

behavior property

- explanation of, 853
- using with `marquee` element object, BC16

`behaviorCookie` property for IE4+ event object,
explanation of, 768

`behaviorPart` property for IE4+ event object,
explanation of, 768

behaviors. *See also* Internet Explorer behaviors

- detaching from objects, 312
- importing and attaching to objects, 259–262

`behaviorUrns[]` property, using with HTML element
objects, 204

beta browsers, dealing with, 155

`bezierCurveTo()` method, using with `canvas` element
object, 640

- bgColor property
 - for body element object, 588
 - using with document object, 524–527
 - using with marquee element object, BC17–BC18
 - using with table element object, BC120
- bgProperties property, using with body element object, 589
- bidirectional event model, availability in W3C DOM, 188–189
- Bill of Rights example, using TDC objects with, 217–219
- binding events
 - cross-browser solution for, 757–758
 - methods for, 270–271, 382
 - through IE attachments, 756
 - through object properties, 755–756
 - through tag attributes, 754–755
 - through W3C listeners, 756–757
- bitwise operators, using, 1008, 1013–1014
- blendTrans () transition filter, description and properties for, 857
- Blinds () filter, description and properties for, 863
- blockquote element object, explanation of, BC4
- blur () filter, description and properties for, 856
- blur () method
 - for HTML element objects, 271–274
 - for text input object, 701
- Body
 - examining in Decision Helper application, 126
 - scripts in, 47–48
- Body and Head, scripts in, 47
- body element object
 - event handlers for, 587
 - explanation of, 587–588
 - methods of DOM objects, 587
 - properties for, 587
 - syntax for, 587
- body property for document object, explanation of, 529–530
- body text objects, features of, BC3
- book, support and updates for, 1105
- bookmarks property for IE4+ event object, explanation of, 769
- Boolean data type, example and description of, 58
- Boolean math, performing, 1010–1012
- Boolean object, explanation of, 926
- Boolean operators, using, 1009, 1012–1013
- border and edge properties, 847–851
 - border elements, returning thickness of, 213
 - border property
 - for frameset element object, 480
 - for img element object, 612
 - for object element object, BC232
 - for table element object, BC120–BC121
 - borderCollapse property, explanation of, 851
 - borderColor property
 - for frame element object, 472–473
 - for frameset element object, 480
 - for table element object, BC121
 - borderColorDark property, using with table element object, BC121
 - borderColorLight property, using with table element object, BC121
 - borderSpacing property, explanation of, 851
 - bottom positioning property, explanation of, 843
 - bottom property, using with TextRectangle object,, BC86–B88
 - bottom value of align property, description of, 486
 - bottomMargin property for body element object, explanation of, 589–590
 - bound data, sorting, 220–221
 - boundElements property for IE4+ event object, explanation of, 769
 - bounding rectangle, explanation of, 291
 - boundingHeight property, using with TextRange object, BC60–BC62
 - boundingLeft property, using with TextRange object, BC60–BC62
 - boundingTop property, using with TextRange object, BC60–BC62
 - boundingWidth property, using with TextRange object, BC60–BC62
 - br element object, explanation of, BC4–5
 - break statement, using with control structures, 77, 978
 - browser crashes, occurrence of, BC341
 - browser document objects
 - explanation of, 163–164
 - road map for, 165
 - browser information
 - accessing, BC357–BC358
 - getting for script1.html, 25
 - browser preferences, reading and writing, BC172–BC174
 - browser scripting, syntax for referencing objects in, 37
 - browser support, availability of, 168

- browser version branching, using object detection for, 151–153
 - browser versions, detecting, 149–154
 - browser window, DOM in, 32–33
 - `browserLanguage` property, using with `clientInformation` and navigator objects, BC163–BC164
 - browsers
 - basing on Mozilla engine, 157
 - choosing, 18
 - coding for nonscriptable browsers, 149–150
 - compatibility of, 9–10
 - dealing with beta browsers, 155
 - decrease in number of, 10
 - functions for examination of, BC159–BC162
 - handling non-JavaScript browsers, 48–49
 - hiding script statements from, 146–147
 - loading framesetting documents into, 120
 - modifying content for scriptable browsers, 153–154
 - placing in CSS standards-compatible mode, 192
 - plug-ins for, 6
 - scripting for, 150–154
 - switching from text editors in Mac OS, 19
 - brushes, setting, 638
 - `bubbles` property for NN6+/Moz/Safari event object, explanation of, 790
 - `bufferDepth` property, using with `screen` object, BC189
 - button element object
 - event handlers for, 669
 - explanation of, 670–671
 - `form` property for, 671–672
 - method for, 669
 - properties for, 669
 - syntax for, 670
 - button input element, creating, 168
 - button input object
 - event handlers for, 669
 - explanation of, 670–671
 - `form` property for, 671–672
 - method for, 669
 - properties for, 669
 - syntax for, 670
 - using with form controls as objects, 100
 - button property
 - for IE4+ event object, 770
 - for NN6+/Moz/Safari event object, 790
 - buttons
 - displaying with applet, BC306–BC309
 - using in forms, 653
 - buttons in radio groups, determining number of, 101–102
 - buttons sharing function code sample, 673–674
- ## C
- `calcBlockState()` function, using in outline-style TOC application, BC433
 - `calculate()` method
 - for Decision Helper application, BC477
 - for order form application, BC416
 - calculation, performing for resistor calculator, BC437–BC438
 - calculator
 - arrays for, BC440
 - calculation involved in, BC437–BC438
 - calculations and formatting for, BC440–BC441
 - changing images on the fly, BC442
 - creating `select` objects for, BC443–BC444
 - drawing initial images for, BC444–BC445
 - graphical user interface ideas for, BC438
 - preloading images for, BC441–BC442
 - calendar
 - generating via dynamic table, BC385–BC388
 - generating via static table, BC385–BC388
 - generating with dynamic HTML table, BC393–BC397
 - `call()` method for `function` object, explanation of, 1030–1031
 - `caller` property for `function` object, description of, 1028
 - Camino, relationship to other browsers, 157
 - `cancelable` property for NN6+/Moz/Safari event object, explanation of, 791
 - `cancelBubble` property
 - for IE4+ event object, 770–771
 - for NN6+/Moz/Safari event object, 791
 - `canHaveChildren` property for HTML element objects, explanation of, 204–206
 - `canHaveHTM` property for HTML element objects, explanation of, 206
 - canvas
 - drawing curves on, 640
 - drawing images to context in, 641
 - drawing lines and adjusting stroke location on, 642
 - drawing lines on, 639
 - filling and clearing areas of, 642
 - joining lines on, 639
 - managing clipping paths on, 641
 - managing paths on, 641
 - canvas content, establishing shadow around, 639–640
 - canvas element object, explanation of, 634–637

- caption element object, explanation of, BC135–BC136
- caption property, using with `table` element object, BC121–BC122
- captions, including with tables, BC104–BC105
- `captionSide` property, explanation of, 852
- `captureEvents()` method for document object, explanation of, 563
- carriage returns
 - appending to end of string, 581
 - inside `textareas`, 709–710
- Cascading Style Sheets (CSS)
 - availability in Internet Explorer 4+, 173
 - significance of, 5, 13–14
- case
 - changing for strings, 111
 - requirements for XHTML style, 23
- case sensitivity
 - of JavaScript, 37
 - of property names, 167
- `catch` block, example of, 992
- CDATA section node, generating, 565
- CDATA sections, encasing script statements in, 148
- CD-ROM for *JavaScript Bible*
 - contents of, 1110–1111
 - Customer Support for, 1111
 - system requirements for, 1109
 - troubleshooting, 1111
- `cellIndex` property, using with `td` and `th` element objects, BC143–BC144
- `cellPadding` property, using with `table` element object, BC122
- `cells` property
 - for `table` element object, BC122–BC123
 - using with `tr` element object, BC139–BC140
- `cellSpacing` property, using with `table` element object, BC122
- center dialog box feature in IE, description of, 455
- CGI programs, inspecting submitted forms with, 30
- CGI prototyping, using JavaScript for, 30
- `ch` property, using with `tbody`, `tfoot`, and `thead` element objects, BC135
- `channelMode` attribute of `open()` method, browsers for and description of, 431
- character conversions code sample, 892
- character set, revealing, 541
- characters
 - extracting copies of, 112–113
 - limitation on typing of, 696
- `characterSet` property for document object, explanation of, 530–531
- `charAt()` method, extracting copies of characters and substrings with, 112–113
- `charCode` property for NN6+/Moz/Safari event object, explanation of, 791–794
- `Charset` parameter for TDC, description of, 216
- `charset` property
 - for document object, 530
 - for a element object, 602
 - using with `link` element object, BC94
 - using with `meta` element object, BC97
- charts, using `canvas` element object with, 636–638
- checkbox input object
 - event handler for, 675
 - explanation of, 675–676
 - methods of DOM objects, 675
 - properties for, 675
 - syntax for, 675
 - using with form controls as objects, 100
- `checked` property
 - for checkbox input object, 676–677
 - for radio input object, 683–684
- `Checkerboard()` filter, description and properties for, 863
- child and parent nodes, roles in W3C DOM, 39
- child elements, removing, 312
- child frame, presence in parent window, 120–121
- child node objects, swapping, 315
- child nodes
 - explanation of, 223
 - inserting into existing elements, 300–301
- `childNodes` attribute object property, description of, 203
- `childNodes` property for W3C DOM Level 2, description of, 180
- `childNodes[]` property for HTML element objects, explanation of, 206–209
- children, inserting elements as, 268–269
- `children` property for HTML element objects, explanation of, 209–210
- child-to-child references, explanation of, 122–123
- child-to-parent references, explanation of, 122
- `chOff` property, using with `tbody`, `tfoot`, and `thead` element objects, BC135
- `chroma()` filter, description and properties for, 856
- `Chroma()` filter, description and properties for, 863
- `chrome` attribute of `open()` method, browsers for and description of, 431

- `circ` shape, coordinates for and example of, 629
- `circle` shape, coordinates for and example of, 629
- `cite` property
 - for HTML element objects, 210
 - using with `blockquote` and `q` element objects, BC4
- classes, scripting Java classes directly, BC320–BC321
- `classid` attribute, using with embedded objects
 - in IE, 216
- `classid` property, using with `object` element object, BC232–BC233
- `className` property for HTML element objects, explanation of, 211–212
- `clear()` method
 - using with `document` object, 563
 - using with `selection` object, BC47
- `clear` property
 - explanation of, 838
 - using with `br` element object, BC5
- `clearAttributes` method for HTML element objects, explanation of, 275
- `clearData()` method
 - for `clipboardData` object, 381
 - for `dataTransfer` object, 777
- `clearInterval()` method for `window` object, explanation of, 417
- `clearRect()` method for `canvas` element object, explanation of, 642
- `clearRequest()` method, using with `userProfile` object, BC194
- `clearTimeout()` method for `window` object, explanation of, 418–420
- `click()` method
 - of `button` element object, 672
 - of `Button` input object, 672
 - for `checkbox` input object, 679
 - for HTML element objects, 275–276
 - for `radio` input object, 686
 - for `Reset` input object, 672
 - for `Submit` input object, 672
- `clientHeight` and `clientWidth` properties for HTML element objects, explanations of, 212–213
- `clientInformation` object, explanation of explanation of, BC154–BC155
- `clientInformation` property for `window` object, explanation of, 380
- `clientLeft` and `clientTop` properties for HTML element objects, explanation of, 213
- `clientX` property
 - for IE4+ event object, 771–776
 - using with NN6+/Moz/Safari event object, 794–797
- `clientY` property
 - for IE4+ event object, 771–776
 - using with NN6+/Moz/Safari event object, 794–797
- `clip()` method, using with `canvas` element object, 641
- `clip` property, explanation of, 838
- `clipboardData` property for `window` object, explanation of, 380–381
- `clipBottom` property, explanation of, 838
- `clipLeft` property, explanation of, 838
- clipping rectangles, working with, BC201–BC207
- `clipRight` property, explanation of, 838
- `clipTop` property, explanation of, 838
- `cloneContents()` method, using with `Range` object, BC27
- `cloneNode()` method for HTML element objects, explanation of, 276–277
- `cloneNode(deep)` node object method, role in W3C DOM Level 2, 183
- `cloneRange()` method, using with `Range` object, BC27
- `close` attribute of `open()` method, explanation of, 431
- `close()` method
 - for `document` object, 563–564
 - for `window` object, 420
- `closed` property for `window` object, explanation of, 381–383
- `closePath()` method for `canvas` element object, explanation of, 641
- closures, relationship to functions, 1038–1039
- code, building incrementally, BC342
- code property
 - for `applet` object, BC228
 - using with `object` element object, BC233
- codebase principal, activating for digital certificates, BC351–B352
- `codeBase` property
 - for with `applet` object, BC228
 - for `object` element object, BC233
- `codeType` property, using with `object` element object, BC233
- `col` and `colgroup` element objects, explanations of, BC136–BC137
- `collapse()` method
 - using with `Range` object, BC28
 - using with `selection` object, BC47–BC48
 - using with `TextRange` object, BC63

- collapsed property, using with `Range` object, BC24
- `collapseToEnd()` method, using with `selection` object, BC48
- `collapseToStart()` method, using with `selection` object, BC48
- collapsible outlines, features of, BC426
- collections of objects, returning number of items in, 232–233
- color change, triggering from pop-up menu, 731–732
- color properties, using with `document` object, 524–526
- color property
 - explanation of, 833
 - using with `font` element object, BC6–BC7
 - using with `hr` element object, BC12
- `colorDepth` property, using with `screen` object, BC190
- colors, setting for layer backgrounds, BC200–BC201
- `cols` property
 - for `frameset` element object, 480–483
 - for `textarea` element object, 710
 - using with `table` element object, BC123
- `colSpan` property, using with `td` and `th` element objects, BC144
- columns in tables, modifying, BC114–BC117
- comma (,) operator, explanation of, 1018
- commands
 - converting to dot syntax, 164
 - executing, 574–576
 - methods as, 61
- comment (//) global statement, explanation of, 1070
- comment (/*...*/) global statement, explanation of, 1070
- comment HTML-related node type, description of, 180
- comment nodes, creating instances of, 565
- comment symbol, // (forward slashes) as, 48–49
- `compact` property, using with `dl`, `dt`, and `dd` element objects, BC151
- `compareBoundaryPoints()` method, using with `Range` object, BC28–BC32
- `compareDocumentPosition()` method for HTML element objects, explanation of, 277
- `compareEndpoints()` method, using with `TextRange` object, BC64–BC67
- `compareNode()` method, using with `Range` object, BC32
- `comparePoint()` method, using with `Range` object, BC32
- comparison operators, using, 64–65, 1002–1003
- compatibility
 - designing for, 154–157
 - striving for, BC363
- compatibility approaches
 - custom APIs, BC368–BC369
 - object detection, BC366–BC368
- `compatMode` property for `document` object, explanation of, 531
- `compile()` method, using with regular expression object, BC256
- `complete` property
 - for `image` input object, 689
 - for `img` element object, 612–613
- `complete` value for `readyState` property, description of, 247
- `componentFromPoint()` method for HTML element objects, explanation of, 277–280
- concatenating strings, 64, 110–111
- condition expressions, satisfying for `if` and `if...else` decisions, 970–971
- condition* item, using with repeat loops, 70
- conditional expressions
 - explanation of, 974
 - using with object detection, 153
- conditions, using with repeat loops, 975
- confirm dialog box, example of, 85–86
- `confirm()` method for `window` object, explanation of, 421–422
- connubial operators, using, 1002, 1005–1007
- `const` global statement, explanation of, 1071–1072
- `constructor` property for `string` object, explanation of, 888
- container, `form` object as, 96
- containment hierarchy, `document` object hierarchy as, 166
- containment versus inheritance, 374
- `contains()` method for HTML element objects, explanation of, 280
- `containsNode()` method, using with `selection` object, BC48
- content, separating from scripting, 190
- content page code sample, 509
- `content` property
 - explanation of, 839
 - for `meta` element object, BC98
 - for `window` object, 380
- content selection, event handler for, 366–367
- `contentDocument` property, 123
 - for `frame` element object, 473
 - for `iframe` element object, 487
 - for `object` element object, BC233

- contentEditable property for HTML element objects,
 - explanation of, 214–215
- contentType property for document object, explanation of, 531
- contentWindow property
 - for frame element object, 473–474
 - for iframe element object, 487
- context menu, event handler for, 334
- context-sensitive help
 - creating, 348–349
 - event handler for, 468
- continue statement, directing loop traffic with, 979
- control elements in forms, maintaining list of, 97
- control panel frame code sample, 483
- control structures
 - conditional expressions, 974
 - do-while loop, 980–981
 - for-in loop, 981–982
 - if and if...else decisions, 969–974
 - if constructions, 68–69
 - if...else constructions, 69
 - labeled statements, 983–985
 - for loops, 975–979
 - repeat loops, 975–979
 - with statement, 982–983
 - switch statement, 985–988
 - using break statement with, 77
 - while loop, 979–980
- controllers[] property for window object,
 - explanation of, 380
- controls
 - selecting for XML data transformation application, BC510
 - setting for Google Maps, BC517
- cookie data, retrieving, 535–540
- cookie file, explanation of, 532
- cookie property for document object, explanation of, 531
- cookie record, explanation of, 532–533
- cookieEnabled property, using with
 - clientInformation and navigator objects, BC164
- cookies
 - accessing, 533
 - concatenating subcookies, 541
 - domain of, 535
 - expiration dates for, 534
 - managing in flag-update application, BC447–BC448
 - name/data for, 534
 - path of, 535
 - saving, 533
 - SECURE parameter, for
- coords property
 - for a element object, 603
 - using with area element object, 629
- copyHistory attribute of open() method, browsers for and description of, 431
- core language objects. *See also* objects
 - Date, 114–115
 - Math, 113
 - String , 110–113
- core language standard
 - development of, 12
 - ECMAScript as, 144
- Count property for Dictionary object,
 - description of, 1074
- countdown timer code sample, 418–419
- counterIncrement property, explanation of, 839
- counterReset property, explanation of, 839
- cpuClass property, using with clientInformation and navigator objects, BC164–BC165
- createAttribute() method for document object,
 - explanation of, 564–565
- CreateBookmark command, parameter for and description of, 575
- createCaption() method, using with table element object, BC131
- createCDATASection() method for document object,
 - explanation of, 565
- createComment() method for document object,
 - explanation of, 565
- createContextualFragment() method, using with Range object, BC33
- createControlRange() method
 - for body element object, 593
 - for HTML element objects, 281
- createDocumentFragment() method for document object,
 - explanation of, 565–566
- createElement() method
 - using with document object, 566–568
 - using with nodes in W3C DOM, 184
- createElementNS() method for document object,
 - explanation of, 566–568
- createEvent() method for document object,
 - explanation of, 567

- `createEventObject()` method for document object,
 - explanation of, 567–568
 - `createLinearGradient()` method for canvas element object, explanation of, 641
 - CreateLink command, parameter for and description of, 575
 - `createNSResolver()` method for document object, explanation of, 568
 - `createPattern()` method for canvas element object, explanation of, 641
 - `createPopup()` method for window object, explanation of, 422
 - `createRadialGradient()` method for canvas element object, explanation of, 641
 - `createRange()` method
 - for document object, 568
 - for selection object, BC48
 - `createStyleSheet()` method for document object, explanation of, 568–570
 - `createTextNode()` method
 - using with document object, 570–571
 - using with nodes in W3C DOM, 184
 - `createTextRange()` method
 - using with body element object, 593
 - using with textarea element object, 711
 - `createTFOOT()` method, using with table element object, BC131–BC132
 - `createTHead()` method, using with table element object, BC131–BC132
 - `createTreeWalker()` method for document object, explanation of, 571–572
 - cross-confirmation validation code sample, BC286
 - cross-platform modifier key check example, 760–763
 - `crypto` property for window object, explanation of, 383
 - CSS (Cascading Style Sheets)
 - availability in Internet Explorer 4+, 173
 - significance of, 5, 13–14
 - CSS features, incompatible implementation of, 191
 - CSS image rollovers code sample, 135–137
 - CSS rules, associating with document elements, 210
 - CSS standards-compatible mode, placing browsers in, 192
 - CSS style object, accessing, 425
 - `cssFloat` property, explanation of, 839
 - CSS-Positioning, significance of, 14
 - `cssRule` object
 - explanation of, 825
 - properties for, 825
 - syntax for, 825
 - `cssRules` property, using with `styleSheet` object, 817
 - `cssText` property
 - for `cssRule` object, 825
 - explanation of, 853
 - for `styleSheet` object, 817–818
 - Ctrl key. *See* keyboard shortcuts
 - `ctrlKey` property
 - for IE4+ event object, 767–768
 - for NN6+/Moz/Safari event object, 789–790
 - `ctrlLeft` property for IE4+ event object, explanation of, 768
 - curly braces (`{}`), using, 74
 - `current` property for `history` object, explanation of, 514
 - `currentNode` method for `TreeWalker` object, explanation of, 596
 - `currentStyle` object
 - explanation of, 828
 - syntax for, 828
 - `currentStyle` property for HTML element objects, explanation of, 215
 - `currentTarget` property for NN6+/Moz/Safari event object, explanation of, 798–799
 - `cursor` property, explanation of, 839
 - cursors, restoring to normal, 564
 - curves, drawing on canvas, 640
 - custom objects
 - creating methods for, 1041–1042
 - creating properties for, 1041
 - custom user data, accessing, 295
 - Customer Support, contacting for *JavaScript Bible* CD-ROM, 1111
- ## D
- `\d` and `\D` matching metacharacters, using in regular expressions, BC242
 - data
 - forms of, 54
 - reflecting state of, 337
 - sorting and filtering, 219
 - data collections, displaying records in, 219
 - data entry validation, example of, 30
 - `data` property
 - for object element object, BC233
 - using with `Text` and `TextNode` objects, BC51
 - data types
 - converting, 62–64
 - descriptions of, 58
 - equality of disparate data types, 1003–1005

- data validation
 - controlling in Decision Helper application, BC462
 - of date entries in forms, 941–944
 - of dates and times, BC269–BC273
 - designing filters for, BC263–BC264
 - event handlers for, BC286
 - example of presubmission validation, 30
 - of forms, 106–108
 - real-time versus batch validation, BC261–BC263
 - of serverless database, BC404–BC405
- database, sorting for XML data transformation application, BC508. *See also* serverless database
- `dataFld` property
 - using, 219
 - using with HTML element objects, 216–217
 - using with IE4+ event object, 769
- `dataFormatAs` property, using with HTML element objects, 216–217
- `dataPageSize` property, using with `table` element object, BC123
- `dataSrc` property for HTML element objects, explanation of, 216–217
- `dataTransfer` property for IE4+ event object, explanation of, 776–778
- `DataURL` parameter for TDC, description of, 216
- date and time arithmetic, performing, 936–937
- date bugs, occurrence in early browsers, 940–941
- date calculations, performing, 115–117
- date entries, validating in forms, 941–944
- date formats, using with older browsers, 935–936
- `Date` object
 - creating, 929–930
 - explanation of, 929
 - methods, 931–934
 - native object properties and methods for, 931
 - syntax for, 929
 - using, 114–115
- date validation sample, BC279–BC285
- dates, treating as strings, 934–935
- dates and times, validating, BC269–BC273
- `dateTime` property for HTML element objects, explanation of, 215. *See also* time
- days, counting, 938–939
- `dd` element object, explanation of, BC150–BC151
- deactivation and activation event handlers, using, 327
- debugging tools. *See also* troubleshooting techniques
 - embeddable Evaluator, BC336–BC337
 - emergency evaluation, BC337–BC338
 - Firefox's FireBug Debugger, BC336
 - Mozilla's Venkman Debugger, BC335
 - Safari's Drosera Debugger, BC336
 - WinIE Script Debugger, BC335
- decimal point, testing for, BC266
- decimal-to-hexadecimal converter function code
 - sample, 916–917
- Decision Helper application
 - controlling multiple frames in, 124–126
 - description of, BC457–BC458
 - design of, BC458–BC459
 - `dh1.htm` data-entry field in, BC467–BC468
 - `dh2.htm` data-entry screen in, BC468–BC470
 - `dh3.htm` data-entry screen in, BC470–BC473
 - `dh4.htm` data-entry fields in, BC473–BC476
 - `dh5.htm` decision support calculations in, BC476–BC479
 - `dhHelp.htm` code in, BC480
 - files for, BC459
 - `index.htm` code for, BC460–BC464
 - navigation bar in, BC465–BC467
- `declare` property, using with object element object, BC234
- `decodeURI` global function, explanation of, 1062–1063
- `decodeURIComponent()` function, explanation of, 912, 1062–1063
- decrement (`--`) connubial operator, operand and result for, 1005
- `defaultCharset` property for document object, explanation of, 541
- `defaultChecked` property
 - for checkbox input object, explanation of, 677–678
 - for radio input object, explanation of, 685
- `defaultStatus` property for window object, explanation of, 383–384
- `defaultValue` property for text input object, explanation of, 695
- `defaultView` property for document object, explanation of, 541–542
- `defer` property, using with `script` element object, BC99
- deferred scripts, running, 50–52
- `_defineGetter_()` method, using, 187
- `_defineSetter_()` method, using, 187
- `delete` object operator, explanation of, 1015
- `deleteCaption()` method, using with `table` element object, BC131
- `deleteCell()` method, using with `tr` element object, BC141

- `deleteConstants()` method, using with `Range` object, BC34–BC35
- `deleteData()` method, using with `Text` and `TextNode` objects, BC52–BC55
- `deleteFromDocument()` method, using with `selection` object, BC48
- `deleteRow()` method, using with `table` element object, BC132
- `deleteRule()` method for `styleSheet` object, explanation of, 824
- `deleteTFoot()` method, using with `table` element object, BC131–BC132
- `deleteTHead()` method, using with `table` element object, BC131–BC132
- dependent attribute of `open()` method, browsers for and description of, 431
- `description` property
 - for `error` object, 998
 - using with `contentType` object, BC175
 - using with `plugin` object, BC179
- descriptions of images, property for, 617
- `designMode` property for `document` object, explanation of, 542
- `detach()` method, using with `Range` object, BC35
- `detachEvent()` method for HTML element objects, explanation of, 270–271
- `detail` property for NN6+/Moz/Safari event object, explanation of, 800
- DHTML (Dynamic HTML). *See also* puzzle map game
 - DHTML application
 - handling non-DHTML browsers, BC369–BC370
 - significance of, 14, BC361–BC362
 - DHTML API, example of, BC370–BC372
 - DHTML behavior
 - availability in Internet Explorer 5+, 174
 - exploiting in W3C DOM, 177
 - DHTML compatibility tricks
 - inline branching, BC364–BC365
 - platform equivalency, BC365–BC366
 - DHTML interactivity, using JavaScript for, 30
 - DHTML techniques
 - changing stylesheet settings, 138
 - dynamic content via W3C DOM nodes, 138–139
- dialog boxes, 439–441. *See also* modal dialog box document
 - code sample
 - advisory about, 456
 - displaying, 436–437
 - displaying with window properties and methods, 85–87
 - `dialogArguments` property for window object, explanation of, 384–385
 - dialog-box data, retrieving with `showModalDialog()` method, 454–455
 - dialog-box features, availability in Internet Explorer, 455
 - `dialogHeight` dialog box feature in IE, description of, 455
 - `dialogHeight` property for window object, explanation of, 386
 - `dialogLeft` dialog box feature in IE, description of, 455
 - `dialogLeft` property for window object, explanation of, 386–387
 - `dialogTop` dialog box feature in IE, description of, 455
 - `dialogTop` property for window object, explanation of, 386–387
 - `dialogWidth` dialog box feature in IE, description of, 455
 - `dialogWidth` property for window object, explanation of, 386
 - `Dictionary` object, properties and methods for, 1074
 - digital certificates
 - activating codebase principal for, BC351–B352
 - obtaining, BC351
 - `dimensions()` method for `VArray` object, description of, 1076
 - `dir` element object, explanation of, BC151
 - `dir` property for HTML element objects, explanation of, 221
 - `direction` property
 - explanation of, 840
 - using with `marquee` element object, BC18
 - directive element, explanation of, BC89
 - `directories` attribute of `open()` method, browsers for and description of, 431
 - `directories` property for window object, explanation of, 387–388
 - `disabled` property
 - for HTML element objects, 221–222
 - for `link` element object, BC94
 - for `styleSheet` object, 818
 - `disabled` status, determining, 230
 - dispatch lookup table for validation, creating, BC274
 - `dispatchEvent()` method for HTML element objects, explanation of, 281–283
 - `display` property, explanation of, 840
 - `div` element in `script1.html`, referring to, 24
 - `<div>` tag in `script1.html`, usage of, 23
 - `divide (/)` connumeral operator, operand and result for, 1005
 - `divide by value (/=)` assignment operator, example of, 1008
 - `d1` element object, explanation of, BC150–BC151

- doctype property for document object, explanation of, 542–543
- DOCTYPE switching, significance of, 191–192
- <!DOCTYPE> element, using with CSS standards-compatible mode, 192
- document fragments, creating instances of, 565–566
- document HTML-related node type, description of, 180
- document location, returning, 543
- document object
 - automatic existence of, 178
 - capabilities of, 88
 - event handlers for, 520–521
 - explanation of, 522–523
 - methods of DOM objects, 520–521
 - properties for, 520–522
 - role in browser-window hierarchy, 33
 - syntax for, 522
 - using in arrays, 78
- document object checks, explanation of, BC349
- document object hierarchy, considering as containment hierarchy, 166
- document object model (DOM). *See also* W3C DOM
 - in browser window, 32–33
 - hierarchy of, 520
 - relationship to HTML structure, 31–32
 - standard negotiated for, 13
- document objects. *See also* objects
 - establishment of, 166
 - event handlers for, 168
 - methods of, 167–168
 - properties for, 166–167
 - separating from language, 11
- document property
 - for HTML element objects, 222
 - using with popup object, 491
 - using with window object, 388
- document tree, navigating, 571–572
- document writing code samples, 583–584
- document.all array, inclusion in Internet Explorer 4+, 172
- document.close() method versus document.write(), 90
- document.createElement() method, using, 92
- document.createTextNode() method, using, 92
- document.domain, setting, BC348
- documentElement property for document object
 - explanation of, 543
- document.forms[] property, using, 88–89
- document.getElementById() method, using, 92
- document.images[] property, using, 89
- document-loading process, 33–36
- documents
 - adding paragraph element to, 34
 - adding paragraph text to, 35
 - loading into windows or frames, 430
 - making new elements in, 35
 - opening, 578–579
 - placement of scripts in, 45–49
- document.selection object code sample, BC46–BC47
- DocumentType HTML-related node type, description of, 180
- documentURI property for document object,
 - explanation of, 543
- document.write() method
 - using, 89–91
 - using with Date object calculation, 116
- document.write() method, parameter required for, 61
- does not equal (!=) comparison operator, operand and result for, 1003
- does not equal operator, symbol for, 65
- DOM (document object model). *See also* W3C DOM
 - in browser window, 32–33
 - hierarchy of, 520
 - relationship to HTML structure, 31–32
 - standard negotiated for, 13
- DOM concepts in W3C DOM, element referencing, 177–178
- DOM Level 0, script references to objects in, 177
- DOM nodes, creating dynamic content with, 138–139
- DOM objects. *See* objects
- domain property for document object,
 - explanation of, 543–544
- done() function, running relative to onload event handler, 51
- doReadRequest() method, using with userProfile object, BC194–BC195
- Dortch, Bill, cookie functions, 536–540
- doScroll() method
 - for body element object, 593–594
 - for HTML element objects, 283–284
- dot syntax
 - converting command to, 164
 - use of, 37, 40–41
- dotted rectangle, appearance of, 226
- double-click sequence, event handler for, 337–338
- do-while loop, explanation of, 980–981
- drag-and-drop operation, events and targets during, 339–340

- equals (==) comparison operator, operand and result for, 1003
- error message text, interpreting, BC324–BC331
- error object exception, throwing, 995
- `error` object, explanation of, 997–998
- error trapping techniques, dependence on JavaScript version, 160
- error-checking mode for documents, revealing, 561
- errors
 - occurrence in scripts, 398
 - relationship to exceptions, 988–989
 - syntax versus runtime errors, BC323–BC324
 - viewing script errors, 52–54
- escape codes, using with strings, 912
- `escape` global function, explanation of, 1064
- `EscapeChar` parameter for TDC, description of, 216
- escaped characters, examples of, 885
- `eval` global function, explanation of, 1064–1065
- `evaluate()` method for `document` object, explanation of, 574
- Evaluator, debugging with, BC336–BC337
- Evaluator Jr., testing expression evaluation with, 61
- The Evaluator Sr., features of, 155–156
- event binding
 - cross-browser solution for, 757–758
 - methods for, 270–271, 382
 - through IE attachments, 756
 - through object properties, 755–756
 - through tag attributes, 754–755
 - through W3C listeners, 756–757
- event bubbling
 - availability in Internet Explorer 4+, 174
 - preventing in IE, 745
- event capture
 - disengaging, 307
 - event handler for, 355–356
- event handlers
 - assigning functions to, 168
 - assignment for IE behaviors, BC375–BC376
 - for `body` element object, 587
 - for `body` text objects, BC20
 - for `button` element object, 669
 - for `Button` input object, 669
 - for `checkbox` input object, 675
 - designing for text fields, 693
 - for `document` object, 520–521
 - for `document` objects, 168
 - for `file` input element object, 736
 - for `form` object, 646
 - for generic HTML element objects, 196–198
 - for `img` element objects, 608
 - of `map` element object, 630
 - in older browsers, 763–764
 - for `Reset` input object, 669
 - for `select` element object, 714
 - for `Submit` input object, 669
 - for `text` input object, 692
 - for `textarea` element object, 708
 - trends related to, 191
 - for validation, BC286
 - for `window` object, 376–377
 - for `XMLHttpRequest` object, 871
- event listeners
 - availability in W3C event model, 188
 - modifying, 189
 - types of, 263
- event models
 - cross-platform key capture, 761–763
 - cross-platform modifier key check, 760–761
- event object
 - compatibility of, 758–760
 - explanation of, 767
 - properties and methods of, 759
 - referencing, 753
 - syntax for, 767
- event object properties related to ADO, descriptions of, 769
- event objects, creating, 567–568
- event propagation
 - in IE4+, 743–747
 - in NN4, 741–743
 - in W3C, 748–753
- event property
 - for `script` element object, BC100
 - for `window` object, 388–389
- event types, in IE4+ and NN6+/W3C, 764–765
- `eventPhase` property for NN6+/Moz/Safari event object, explanation of, 800
- event-related tasks, using `componentFromPoint()` method for, 277–280
- events
 - allowing firing of, 281–283
 - bidirectional event model in W3C DOM, 188–189
 - binding, 382
 - binding to scripts, 146
 - explanation of, 740
 - firing, 284–286
 - redirecting in IE4+, 745–746
 - static `Event` object, 741

- events of DOM objects, explanation of, 42
- exception handling
 - exception mechanism, 989–990
 - exceptions and errors, 988–989
- exceptions
 - error object exception, 995
 - object exception, 996–997
 - raising, 991
 - string exceptions, 994
 - throwing, 991, 993–997
- exec() method, using with regular expression object, BC256–BC257
- execCommand() method
 - for document object, explanation of, 574–576
 - for TextRange object, BC68–BC70
- execScript() method for window object, explanation of, 423
- Exists() method for Dictionary object, description of, 1074
- expand() method, using with TextRange object, BC71
- expandentityReference method for TreeWalker object, explanation of, 597
- expando property for document object, explanation of, 544–545
- “Expected }” error message text, explanation of, BC328–BC329
- “Expected <something>” error message text, explanation of, BC327
- expression evaluation
 - explanation of, 60
 - testing, BC342
- expression text, returning, 295
- expressions
 - evaluating XPath expressions, 574
 - removing, 312–313
 - setting, 321–324
 - using in scripts, 60–62
 - using with variables, 62
- extend() method, using with selection object, BC49
- extendRow() method, using in order form application, BC413–BC414
- external property for window object, explanation of, 389–390
- extractContents() method, using with Range object, BC35–BC36

F

- face property, using with font element object, BC8
- faceless applets, using, BC298–BC302
- Fade() filter, description and properties for, 863
- FAQs, consulting, 1106
- fgColor property for document object, explanation of, 524–527
- fieldDelim parameter for TDC
 - description of, 217
 - using, 219
- fieldset element object
 - explanation of, 664
 - properties for, 663
 - syntax for, 664
- file information for web page code sample, 546
- file input element object, event handler for, 736–737
- fileCreatedDate property
 - for document object, 545–546
 - for img element object, 614–615
- fileModifiedDate property
 - for document object, 545–546
 - for img element object, 614–615
- fileName property for error object, explanation of, 999
- filename property, using with plugin object, BC179
- fileSize property
 - for document object, 545–546
 - for img element object, 614–615
- fileUpdatedDate property for img element object, explanation of, 614–615
- fill() method for canvas element object, explanation of, 642
- fillRect() method for canvas element object, explanation of, 642
- fillStyle property for canvas element object, explanation of, 638
- filter, using tags() method as, 325
- filter method for TreeWalker object, explanation of, 597
- filter object, explanation of, 855
- filter property, explanation of, 840
- filter switching, event handler for, 345–346
- filter syntax changes in WinIE4.5+, 861–865
- filter-function libraries, building for data-validation, BC264–BC267
- filtering data, 219

- filters
 - designing for data validation, BC263–BC264
 - IE4+ transition filters, 857–858
 - IE4-compatible static filter types, 856–857
- `filters[]` property for HTML element objects, explanation of, 222–223
- finally block, example of, 992
- `find()` method for window object, explanation of, 424
- `findText()` method, using with `TextRange` object, BC72–BC76
- `fireEvent()` method for HTML element objects, explanation of, 284–285
- Firefox, relationship to other browsers, 157
- Firefox's FireBug Debugger, features of, BC336
- `firstChild` attribute object property, description of, 203
- `firstChild()` method for `TreeWalker` object, explanation of, 597
- `firstChild` property
 - for HTML element objects, 223–225
 - for W3C DOM Level 2, 180, 183
- `firstPage()` method, using with `table` element object, BC132
- flag-update application
 - `<body>` tags in, BC454
 - code for, BC451–BC453
 - date comparison aspect of, BC454
 - description of, BC449–BC450
 - managing cookies in, BC447–BC448
 - time tracking scheme for, BC448–BC449
- `flipH()` filter, description and properties for, 856
- `flipV()` filter, description and properties for, 856
- floating-point numbers versus integers, 63, 913
- focus
 - determining, 524
 - event handlers for, 347
- `focus()` method
 - for HTML element objects, 271–274
 - for text input object, 701
- `focusNode` property, using with `selection` object, BC45
- `focusOffset` property, using with `selection` object, BC45
- font and properties, 833–838
- font element object
 - explanation of, BC5–BC6
 - properties for, BC6–BC9
- font properties, changing dynamically, BC7
- font property, explanation of, 833
- `FontColor` command, parameter for and description of, 575
- `fontFamily` property, explanation of, 833
- `FontName` command, parameter for and description of, 575
- `FontSize` command, parameter for and description of, 575
- `fontSize` property, explanation of, 833
- `fontSizeAdjust` property, explanation of, 833
- `fontSmoothingEnabled` property, using with `screen` object, BC190
- `fontStretch` property, explanation of, 833
- `fontStyle` property, explanation of, 834
- `fontVariant` property, explanation of, 834
- `fontWeight` property, explanation of, 834
- for loop. *See also* loops
 - in puzzle map game DHTML application, BC492
 - using, 70, 975–979
 - using with `form.elements[]` property, 97
 - using with parallel arrays, 77
- for-in loop, explanation of, 981–982
- form attributes, changing, 652
- form code sample, 548–549
- form control elements
 - references to, 647
 - using `blur()` and `focus()` methods with, 272–273
- form controls as objects
 - button input object, 100
 - checkbox input object, 100
 - considering, 97–98
 - radio input object, 101–102
 - `select` object, 102–104
 - text-related input objects, 98–100
- form data and elements, passing to functions, 104–106
- form element arrays, using, 653–654
- form object
 - event handlers for, 646
 - explanation of, 646–647
 - methods of DOM objects, 646
 - as object and container, 96
 - passing as parameter, 648–649
 - properties for, 646
 - syntax for, 646
 - using, 96–97
- form object checks, explanation of, BC349
- form objects in documents, determining number of, 88–89

- form property
 - accessing, 96–97
 - for button element object, 671–672
 - for Button input object, 671–672
 - for label element object, BC14
 - for object element object, BC234
 - for Reset input object, 671–672
 - for Submit input object, 671–672
 - for text input object, 696
- form values, clearing from Reload/Refresh actions, 510–511
- <form> tag, elements nested in, 97
- form.elements array code sample, 657
- form.elements[] property, using, 97
- form.reset() and form.submit() methods code sample, 660–661
- forms
 - buttons in, 653
 - e-mailing, 651–652
 - inspecting with CGI programs, 30
 - passing to functions, 648–651
 - redirecting after submission, 653
 - resetting, 660–661
 - sending reset event to, 662
 - submitting, 661
 - submitting and prevalidating, 106–108
 - using <input> element objects with, 654
 - using onsubmit event handler with, 106–107
 - validating date entries in, 941–944
- forms[] property for document object, explanation of, 547–549
- forward() method
 - for history object, 515–517
 - for window object, 417
- forward slashes (/), purpose of, 48–49, 147
- fragment HTML-related node type, description of, 180
- frame content table of contents code sample, 482–483
- frame element object, explanation of, 471–472
- frame object model, explanation of, 370–371
- frame property, using with table element object, BC124–BC126
- frame scripting, tips for, 123
- frame source code, viewing, 375
- frame versions, switching back to, 374
- frameBorder property
 - for frame element object, 474
 - for frameset element object, 483
- frameElement property for window object, explanation of, 390
- frames. *See also* iframe element object
 - controlling with navigation bars, 124–126
 - creating, 370
 - versus frame element objects, 375
 - hiding and showing, 481–482
 - loading documents into, 430
 - naming in Decision Helper application, 126
 - referencing, 372
 - synchronizing, 374–375
 - using blank frames, 375
- frames property
 - for document object, 549
 - for window object, 390–393
- frameset definition, example of, 120
- frameset element objects, explanation of, 478
- frameset for property picker code sample, 500
- frameset for URL example, 554–555
- frameset objects, properties for, 478
- <frameset> tag, specifying onload event handler in, 123
- framesets
 - code sample, 508
 - for hiding/showing frame, 481–482
 - for navigation lab, 516
 - visualizing as window objects, 120
- framesetting documents, loading into browsers, 120
- frameSpacing property
 - for frameset element object, 484
 - for iframe element object, 487
- framing
 - ensuring, 373–374
 - preventing, 373
- fromElement property for IE4+ event object, explanation of, 778–779
- fullscreen attribute of open() method, browsers for and description of, 431
- fullScreen() function, invoking window object methods with, 167
- fullScreen property for window object, explanation of, 393
- function data type, example and description of, 58
- “Function does not always return value” error message text, explanation of, BC330
- function object, explanation of, 1023–1024
- function parameters, passing with setInterval() function, 447–448
- function workbenches, building, BC342

functions. *See also* global functions and statements

- assigning to custom properties for objects, 187
- assigning to event handlers, 168
- capabilities of, 70–71, 160
- creating, 1024–1025
- for examining browsers, BC159–BC162
- global and local variables, 1033–1035
- invoking, 1032
- making generalizable, 1037–1038
- naming parameters for, 104
- nesting, 1025–1026
- parameter variables, 1035–1036
- parameters for, 1026–1027
- parameters of, 71–72
- passing form data and elements to, 104–106
- passing forms to, 648–651
- recursion in, 1036
- relationship to closures, 1038–1039
- relationship to deferred scripts, 50
- turning into libraries, 1037–1038
- using `this` keyword with, 104
- using variables with, 72

G

`getckoActiveXObject()` method for window object, explanation of, 425

generic HTML element objects. *See also* HTML element objects; objects

- contextual objects, 196
- event handlers for, 196–198
- formatting objects, 196
- methods of, 196–198
- properties for, 196–198

`getAdjacentText()` method for HTML element objects, explanation of, 287–288

`getAllResponseHeaders()` method for XMLHttpRequest object, explanation of, 878

`getAttribute()` method

- using with HTML element objects, 288
- using with `userProfile` object, BC196

`getAttributeNode()` method for HTML element objects, explanation of, 289–290

`getAttributeNodeNS()` method for HTML element objects, explanation of, 290

`getAttributeNS()` method for HTML element objects, explanation of, 290

`getBookmark()` method, using with `TextRange` object, BC77

`getBoundingClientRect()` method for HTML element objects, explanation of, 291–293

`getClientRects()` method for HTML element objects, explanation of, 293–294

`getComputedStyle()` method for window object, explanation of, 425

`getContext()` method for canvas element object, explanation of, 642

`getData()` method for `clipboardData` object, description of, 381

`getData()` method for `dataTransfer` object, description of, 777

`getElementById()` command, using to reference objects, 37

`getElementById()` method for document object, explanation of, 576–577

`getElementsByName()` method for document object, explanation of, 577

`getElementsByTagName()` method for HTML element objects, explanation of, 294

`getElementsByTagNameNS()` method for HTML element objects, explanation of, 294

`getExpression()` method for HTML element objects, explanation of, 295

`getFeature()` method for HTML element objects, explanation of, 295

`getItem()` method for `VBArray` object, description of, 1076

`getMonth()` method, use of zero (0) with, 115

`getRangeAt()` method, using with `selection` object, BC49

`getResponseHeader()` method for XMLHttpRequest object, explanation of, 878

`getSelection()` method for window object, explanation of, 425

getters and setters, defining for object properties, 1050–1051

`getUserData()` method for HTML element objects, explanation of, 295

global functions and statements, list of, 1062. *See also* functions

global property, using with regular expression object, BC255

global scope, defining in JavaScript, 159

- global variables
 - defining `newWindow` variable as, 83
 - explanation of, 72
 - relationship to functions, 1033–1035
 - using with clipping rectangles, BC204
- `globalAlpha` property for canvas element object, explanation of, 638
- `globalCompositeOperation` property for canvas element object, explanation of, 638
- Globally Unique Identifier (GUID), using in IE, 216
- `glow()` filter, description and properties for, 856
- `Glowl()` filter, description and properties for, 864
- GMT (Greenwich Mean Time), relationship to time zones, 927–928
- `go()` method for `history` object, explanation of, 517–518
- `goNext()` function, using in Decision Helper application, 126
- Google Maps application
 - creating markers on, BC518
 - designing mashup for, BC519
 - displaying customization of, BC521–BC524
 - getting API key for, BC516–B517
 - haunted place customization example, BC519–BC521
 - local search feature of, BC515
 - markers in, BC514
 - mashups in, BC514
 - obtaining geocoordinates of locations in, BC518–BC519
 - opening, reading, and processing XML document for, BC523–BC524
 - resources for, BC516
 - RideFinder feature of, BC515
 - search matches in, BC514
 - setting control for, BC517
 - setting default area for, BC517–BC518
 - style sheet for, BC525
 - styling custom information window in, BC524–BC525
 - testing, BC525–BC527
 - using `GMap()` function with, BC517
- graphic context, saving and restoring state of, 643
- graphical user interface, creating for resistor calculator, BC438
- `gray()` filter, description and properties for, 856
- greater than (`>`) comparison operator, operand and result for, 1003
- greater than or equal to (`>=`) comparison operator, operand and result for, 1003

- Greenwich Mean Time (GMT), relationship to time zones, 927–928
- GUID (Globally Unique Identifier), using in IE, 216

H

- `h1...h6` element objects, explanation of, BC9
- `hasAttribute()` method for HTML element objects, explanation of, 295
- `hasAttributeNS()` method for HTML element objects, explanation of, 296
- `hasAttributes()` method for HTML element objects, explanation of, 296
- `hasChildNodes()` method for HTML element objects, explanation of, 296–297
- `hasChildNodes()` node object method, role in W3C DOM Level 2, 183
- `hash` property for a element object, explanation of, 603
- `hash` property for `location` object, explanation of, 498–500
- hash table, simulating, 952–953
- `hasOwnProperty()` method for `object` object, explanation of, 1057
- Haunted Places map, testing, BC525–BC527
- `hauntmap.htm` web page code sample, BC521–BC522
- Head, scripts in, 46–47
- Head and Body, scripts in, 47
- `head` element object, explanation of, BC91
- `headers` property, using with `td` and `th` element objects, BC143
- height and width of elements, reporting, 237–238
- height and width settings for windows code sample, 395–396
- `height` attribute of `open()` method, browsers for and description of, 431
- height of windows, adjusting, 394–396
- `height` property
 - for `applet` object, BC229
 - for `document` object, 550
 - for `embed` element object, BC237
 - explanation of, 844
 - for `frame` element object, 474–475
 - for HTML element objects, 225–226
 - for `iframe` element object, 488
 - for `img` element object, 615
 - for `marquee` element object, BC18
 - for `object` element object, BC234
 - for `screen` object, BC188–BC189
 - for `table` element object, BC126–BC127

- for `td` and `th` element objects, BC145
- for `tr` element object, BC140
- help
 - creating context-sensitive help, 348–349
 - opening Winhelp window, 453
- help dialog box feature in IE, description of, 455
- helper applications, using with browsers, 6
- hexadecimal and octal integers, using, 916–917
- hidden input object, explanation of, 707–708
- hidden property, using with `embed` element object, BC237
- `hide()` method for `popup` object, explanation of, 492–494
- `hideFocus` property for HTML element objects, explanation of, 226
- `hideTip()` function, using in Decision Helper application, BC464
- hiding scripts, explanation of, 49
- hierarchy of object model, significance of, 163–165
- `HIERARCHY_REQUEST_ERR` exception, throwing, 991
- history list, counting items in, 514–515
- history object
 - explanation of, 513
 - methods of DOM objects, 513
 - properties for, 513
 - role in browser-window hierarchy, 33
 - syntax for, 513
- `home()` method for `window` object, explanation of, 427
- host property
 - for `a` element object, 603
 - for `location` object, 500–503
- hostname property
 - for `a` element object, 603
 - for `location` object, 503–504
- `hotkeys` attribute of `open()` method, browsers for and description of, 431
- `hr` element object, explanation of, BC10
- `hr` object properties, controlling, BC11–BC12
- `href` attributes, applying `javascript:pseudo-URL` to, 137
- `href` property
 - for `base` element object, BC92
 - for `a` element object, 603
 - for `link` element object, BC94
 - for `location` object, 87, 504–505
 - for `styleSheet` object, 818
- `hreflang` property for a `element` object, explanation of, 604
- `hrefLang` property, using with `link` element object, BC94
- `hspace` property
 - for `applet` object, BC229
 - for `forth` object element object, BC234
 - for `iframe` element object, 488
 - for `img` element object, 616
 - for `marquee` element object, BC18
- HTAs (HTML applications), availability
 - in Internet Explorer 5+, 174
- `.htc` behavior, explanation of, BC375
- HTML (Hypertext Markup Language)
 - examining in `script1.htm`, 23
 - loading external HTML into layers, BC213–BC214
 - purpose of, 4
 - requirements for applet-to-script communication, BC304
 - tags in, 4
 - transforming XML data into, BC524–BC525
 - treatment in W3C DOM, 177
- HTML applications (HTAs), availability
 - in Internet Explorer 5+, 174
- HTML content, determining acceptability of, 206
- HTML directive objects, explanation of, BC89
- HTML documents
 - calling with applets, BC309
 - code sample, 178
 - embedding scripts in, 145–149
 - minimum elements in, 32
 - nodes associated with, 179–180
- `html` element object, explanation of, BC90
- HTML element objects. *See also* generic HTML element objects
 - adding read-only prototype property to, 188
 - inclusion in Internet Explorer 4+, 172
 - returning references to, 543
- HTML forms. *See* forms
- HTML structure, relationship to DOM, 31–33
- HTML tags
 - adding `id` attributes to, 36–37
 - use of, 31
- HTML text, inserting, 299–300
- `htmlFor` property
 - for `label` element object, BC15, 666
 - using with `script` element object, BC100
- `htmlText` property, using with `TextRange` object, BC62–BC63
- HTTP status codes for `status` property, 876–877
- `httpEquiv` property, using with `meta` element object, BC98

- Hypertext Markup Language (HTML)
 - examining in `script1.htm`, 23
 - purpose of, 4
 - tags in, 4
 - transforming XML data into, BC524–BC525
- hyphens (-), prohibition in JavaScript, 173
- I**
- `ibound()` method for `VBArray` object,
 - description of, 1076
- `id` attributes
 - adding to HTML tags, 36–37
 - getting elements by, 576–577
 - versus `name` attributes, 37
 - treatment in W3C DOM, 177
- `id` property
 - generating, 257–259
 - for HTML element objects, 226–227
 - for `styleSheet` object, 819
- IDs
 - assigning, 37
 - rules related to, 37
- IE (Internet Explorer). *See also* Internet Explorer (IE)
 - `all[]` property for, 201–202
 - `behaviorUrns` property for, 204
 - data-binding facilities in, 216–221
 - dialog-box features in, 455
 - GUID (Globally Unique Identifier) in, 216
 - idiosyncrasies related to `open()` method, 436
 - preventing event bubbling in, 745
 - references in, 216
 - using with 32-bit Windows operating systems, 173
 - using `classid` attribute in, 216
- IE attachments, binding events through, 756
- IE browser versions, problems with, 155
- IE query commands, descriptions of, 579
- IE3, event types in, 763–764
- IE4 extensions, browser support for, 168
- IE4+
 - event object properties and methods for, 759
 - event propagation in, 743–747
 - event types in, 765–766
 - modifying select options in, 716–721
 - table modification methods for, BC109
- IE4+ `clientInformation` object, explanation of, BC154–BC155
- IE4+ event object, explanation of, 767
- IE4+ transition filters, 857–858
- IE4-compatible static filter types, 856–857
- IE5 extensions, browser support for, 168
- IE5+, viewing script errors in, 53
- `if` and `if...else` decisions, explanation of, 969–974
- `if` constructions
 - indentation of, 973
 - structure of, 68–69
 - using with `document.images[]` property, 89
 - using with parallel arrays, 77
- `if...else` constructions
 - code sample, 973
 - nesting, 972–974
 - structure of, 69
- `iframe` element, explanation of, 124
- `iframe` element object, explanation of, 484–485.
 - See also* frames
- `ignoreCase` property, using with regular expression object, BC255
- image arrays, maintaining collection of, 89
- `image` input object, explanation of, 688–689
- image object and rotating images code sample, 620–622
- image object checks, explanation of, BC349
- image objects
 - constructing in memory, 130
 - storage of references for, 129
- image `onload` event handler code sample, 625–626
- image precaching, 130–132, 619. *See also* images
- image rollovers, creating, 132–135. *See also* rollover states
- image size, control of, 130
- `image.complete` code sample, 613
- `imageOff()` function, invoking with `onmouseout` event handler, 133–134
- `imageOn()` function, invoking with `onmouseover` event handler, 133–134
- images. *See also* precaching images
 - drawing for resistor calculator, BC444–BC445
 - flagging, BC447
 - interchanging, 129–130
- `images[]` property for `document` object, explanation of, 550–551
- `imeMode` property, explanation of, 853
- `img align` property code sample, 611
- `img` element objects
 - event handlers for, 608
 - properties for, 608
 - syntax for, 608
- `img` objects, explanation of, 609–610

- implementation property for document object, explanation of, 551
- @import selector, using with stylesheets, 813
- importNode() method for document object, explanation of, 578
- imports property for styleSheet object, explanation of, 819
- in object operator, explanation of, 1016
- increment (++) connumbial operator, operand and result for, 1005
- Indent command, parameter for and description of, 575
- index value, setting in select object, 724
- indexes, role in arrays, 74
- index.htm code, using in Decision Helper application, BC460–BC464
- inheritance in OOP, relationship to prototypes, 1053–1054
- inheritance versus containment, 374
- init() function, using in puzzle map game DHTML application, BC494
- initEvent() method for NN6+/Moz/Safari event object, explanation of, 807–808
- initial expression item, using with repeat loops, 70
- initialize() function, using in flag-update application, BC453
- initKeyEvent() method for NN6+/Moz/Safari event object, explanation of, 807–808
- initMouseEvent() method for NN6+/Moz/Safari event object, explanation of, 807–808
- initUIEvent() method for NN6+/Moz/Safari event object, explanation of, 807–808
- inline branching, using for DHTML compatibility, BC364–BC365
- inline characters, examples of, 885
- inline display and layout properties, 838–843
- innerHeight attribute of open() method, browsers for and description of, 431
- innerHeight property for window object, explanation of, 394–396
- innerHTML property
 - advisory about use of, 154
 - dynamic content through, 139
 - explanation of, 227–229
 - implementation of, 186–187
- innerText property for HTML element objects, explanation of, 227–229
- innerWidth attribute of open() method, explanation of, 431
- innerWidth property for window object, explanation of, 394–396
- input element, using with form controls as objects, 100–102
- input property, using with RegExp object, BC258
- <input> element objects, using with forms, 654
- inputEncoding property for document object, explanation of, 552
- inRange() method, using with TextRange object, BC77
- insertAdjacentElement() method for HTML element objects, explanation of, 297–299
- insertAdjacentHTML() method for HTML element objects, explanation of, 299–300
- insertAdjacentText() method for HTML element objects, explanation of, 299–300
- insertBefore() method
 - for HTML element objects, 300–301
 - role in W3C DOM Level 2, 183
- insertCell() method, using with tr element object, BC141
- insertData() method, using with Text and TextNode objects, BC52–BC55
- insertNode() method, using with Range object, BC36–BC38
- insertRow() method, using with table element object, BC132
- insertRule() method for styleSheet object
 - description and example of, 824
 - description of, 824
- instanceof object operator, explanation of, 1016
- integers
 - versus float-point numbers, 63, 913–916
 - hexadecimal and octal integers, 916–917
- interactive value for readyState property, description of, 247
- interCap format, using with variable names, 60
- international characters, avoiding in signed scripts, BC360
- Internet Explorer (IE). *See also* IE (Internet Explorer)
 - a11[] property for, 201–202
 - behaviorUrns property for, 204
 - data-binding facilities in, 216–221
 - dialog-box features in, 455
 - GUID (Globally Unique Identifier) in, 216
 - idiosyncrasies related to open() method, 436
 - preventing event bubbling in, 745
 - references in, 216
 - using with 32-bit Windows operating systems, 173
 - using classid attribute in, 216

- Internet Explorer 3
 - dominance of, 12
 - scripting included in, 12
 - Internet Explorer 4+ extensions
 - CSS (Cascading Style Sheets), 173
 - element containment hierarchy, 172
 - event bubbling, 174
 - HTML element objects, 172
 - phantom page syndrome in, 173
 - Internet Explorer 5+ extensions
 - DHTML behaviors, 174
 - HTAs (HTML applications), 174
 - Internet Explorer behavior examples
 - element dragging, BC377–BC380
 - text rollover, BC380–BC383
 - Internet Explorer behaviors. *See also* behaviors
 - component structure, BC375–BC376
 - enabling and disabling, BC374
 - linking in behavior component, BC374
 - resources for, BC383
 - stylesheets for scripts, BC373–BC374
 - Internet Explorer (IE)
 - GUID (Globally Unique Identifier) in, 216
 - preventing event bubbling in, 745
 - references in, 41
 - using `classid` attribute in, 216
 - `intersectsNode()` method, using with `Range` object, BC38
 - interval loop action, turning off, 417
 - intervals, setting, 446–450
 - `invert()` filter, description and properties for, 856
 - `Iris()` filter, description and properties for, 864
 - is greater than (>) comparison operator, operand and result for, 1003
 - is greater than operator, symbol for, 65
 - is greater than or equal to (>=) comparison operator, operand and result for, 1003
 - is greater than or equal to operator, symbol for, 65
 - is less than (<) comparison operator, operand and result for, 1003
 - is less than operator, symbol for, 65
 - is less than or equal to (<=) comparison operator, operand and result for, 1003
 - is less than or equal to operator, symbol for, 65
 - `isChar` property for NN6+/Moz/Safari event object, explanation of, 800
 - `isCollapsed` property, using with `selection` object, BC45
 - `isContentEditable` property for HTML element objects, explanation of, 229–230
 - `isDefaultNamespace()` method for HTML element objects, explanation of, 302
 - `isDisabled` property for HTML element objects, explanation of, 230
 - `isEmpty()` function, using in data validation, BC264–BC265
 - `isEqual()` method, using with `TextRange` object, BC78
 - `isEqualNode()` method for HTML element objects, explanation of, 302
 - `isFinite()` global function, explanation of, 1065
 - `isInteger()` function, using in data validation, BC265–BC266
 - `isMap` property for `img` element object, explanation of, 616
 - `isMultiLine` property for HTML element objects, explanation of, 230–231
 - `isNaN()` global function, explanation of, 1065–1066
 - `isNumber()` function, using in data validation, BC266
 - `isOpen` property for `popup` object, explanation of, 491–492
 - `isPointInRange()` method, using with `Range` object, BC38
 - `isPrototypeOf()` method for `object` object, explanation of, 1057
 - `isSameNode()` method for HTML element objects, explanation of, 302
 - `isSupported()` method
 - for HTML element objects, 302
 - role in W3C DOM Level 2, 183
 - `isTextEditable` property for HTML element objects, explanation of, 231
 - `isTrusted` property for NN6+/Moz/Safari event object, explanation of, 800
 - `item()` method
 - for `Enumerator` object, 1075
 - for HTML element objects, 302–303
 - for `select` element object, explanation of, 730–731
 - `Item()` property for `Dictionary` object, description of, 1074
 - `Items()` method for `Dictionary` object, description of, 1074
- ## J
- Jasob obfuscator, downloading, 147
 - Java
 - building block classes in, BC290
 - methods in, BC291–BC293

Java classes, scripting directly, BC320–BC321
 Java fields, accessing, BC293
 Java language, derivation of, 6
 Java sandbox, explanation of, BC346–BC347
`javaEnabled()` method, using with `clientInformation`
 and `navigator` objects, BC171

JavaScript

built-in objects, 158
 case sensitivity of, 37
 control structures in, 160
 global scope defined in, 159
 guidelines for use of, 30–31
 limitations of, 8
 loosely typed feature of, 159
 object-based feature of, 158
 as scripting language, 158
 solutions offered by, 8
 transition from LiveScript, 7
 treatment of functions in, 160
 versions of, 12, 143–144

JavaScript Bible

contents of CD-ROM for, 1110–1111
 support and updates for, 1105

JavaScript expression evaluation, testing, 61

JavaScript Obfuscator, downloading, 147

`javascript:` pseudo-URL

using, 137

using `void` operator with, 1020

JavaScript resources, consulting, 1106–1107

`.js` files, using, 148–149

`JSObject` class, role in applet-to-script communication,
 BC304

jukebox, scripting, BC313–BC317

`justify` commands, parameters for and
 descriptions of, 575

K

key presses

checking events for, 762–763

preventing from becoming typed characters, 351

`Key()` property for `Dictionary` object,
 description of, 1074

keyboard events

scripting, 349–350

typical tasks for, 351–355

keyboard shortcuts

reloading saved source files, 19

save-switch-reload sequence steps, 19

switching between text editor and browser in Mac OS, 19
 using `accessKey` property for HTML element objects
 with, 199

`keyCode` property

for IE4+ event object, 779–782

for NN6+/Moz/Safari event object, 791–794

`Keys()` method for `Dictionary` object,

description of, 1074

L

`label` element object, explanation of, BC13–BC14, 665–666

`label` property

for `optgroup` element object, 734–736

for `option` element object, 733–734

labeled statements, explanation of, 983–985. *See also*
 statements

`lang` property for HTML element objects,
 explanation of, 231

language, separating from document objects, 11

language attribute of `<script>` tag, use of, 46, 145–146

`Language` parameter for TDC, description of, 217

language property for HTML element objects

explanation of, 232

using with `clientInformation` and `navigator`
 objects, BC165

`lastChild` attribute object property, description
 of, 203

`lastChild()` method for `TreeWalker` object,
 explanation of, 597

`lastChild` property for HTML element objects, explanation
 of, 223–225

`lastChild` property for W3C DOM Level 2, significance of,
 180, 183

`lastIndex` property, using with regular expression
 object, BC255

`lastMatch` property, using with `RegExp` object, BC259

`lastModified` property for `document` object, 552
 description of, 552

example of, 552–553

`lastPage()` method, using with `table` element
 object, BC132

`lastParen` property, using with `RegExp` object, BC259

layer, definition of, BC197

layer and clip location properties, comparing, BC205–BC207

layer backgrounds

setting, BC198–BC199

setting colors for, BC200–BC201

layer clipping, implementing, BC201–BC207

- layer element, inclusion in Navigator 4 object model, 171
- layer source content, setting, BC214
- layer stacking order, scripting, BC216–BC218
- layers
 - dragging, BC219–BC221
 - loading external HTML into, BC213–BC214
 - resizing, BC221–BC224
 - scripting nested layers, BC207–BC213
 - visibility relationships of nested layers, BC215–BC216
- layers property for document object, explanation of, 553
- layerX property for NN6+/Moz/Safari event object, explanation of, 794–797
- layerY property for NN6+/Moz/Safari event object, explanation of, 794–797
- layout and inline display properties, 838–843
- layoutGrid property, explanation of, 840
- layoutGridChar property, explanation of, 840
- layoutGridLine property, explanation of, 841
- layoutGridMode property, explanation of, 841
- layoutGridType property, explanation of, 841
- layout-operation completion, event handler for, 355
 - try-catch routine, 992
- left attribute of open() method, browsers for and description of, 431
- left property
 - explanation of, 844
 - using with TextRectangle object., BC86–B88
- left shift (<<) operator, operands for, 1014
- left shift by value (<<=) assignment operator, example of, 1008
- left value of align property, description of, 486
- leftContext property, using with RegExp object, BC259
- leftMargin property for body element object, explanation of, 589–590
- legend element object
 - explanation of, 664
 - properties for, 663
 - syntax for, 664
- length property
 - for array object, 953
 - for form object, 659
 - for function object, 1029–1030
 - for history object, 514–515
 - for HTML element objects, 232–233
 - for plugin object, BC179
 - for radio input object, 685
 - for select element object, 723
 - for string object, 888–889
- less than (<) comparison operator, operand and result for, 1003
- less than or equal to (<=) comparison operator, operand and result for, 1003
- letterSpacing property, explanation of, 834
- Level 0 of DOM, script references to objects in, 177
- Level 2 of W3C DOM, node object properties for, 180
- Levels 0-3 in W3C DOM, explanations of, 175–176
- li element
 - explanation of, BC149
 - using with rollovers, 135–137
- libraries, turning functions into, 1037–1038
- light () filter, description and properties for, 857
- Light () filter, description and properties for, 864
- line terminator, treatment as statement delimiter, 161
- lineBreak property, explanation of, 834
- lineCap property for canvas element object, explanation of, 639
- lineHeight property, explanation of, 834
- lineJoin property for canvas element object, explanation of, 639
- lineNumber property for error object, explanation of, 999
- lines, drawing and joining on canvas, 639
- lines of text, determining display of, 230–231
- lineTo () method for canvas element object, explanation of, 642
- lineWidth property for canvas element object, explanation of, 639
- link element object, explanation of, BC92–BC94, 600
- link property for body element object, explanation of, 588
- linkColor property for document object, explanation of, 524–527
- linked script library object checks, explanation of, BC349
- links [] property for document object, explanation of, 553–554
- list properties, 850–851
- Listings
 - accessKey property for HTML element objects, 200–201
 - addBehavior () and removeBehavior (), 261–262
 - addBehavior () makeHot.htc behavior component, 260
 - alert dialog box, 416
 - anchors for navigating page, 527–528
 - anchors in document, 499
 - appendChild () child methods, 267–268

- `applyElement()` method, 269–270
- area elements modified on the fly, 631–633
- array concatenation, 957–958
- array generation and population, 947
- `Array.join()` method, 959
- `Array.reverse()` method, 961–962
- `Array.sort()` method, 965–966
- binding data to page, 218–219
- `blur()` and `focus()` methods, 273–274
- Boolean operators, 1012–1013
- bubble and capture prevention in W3C, 750–752
- button with event handler, 42
- buttons sharing function, 673–674
- `canHaveChildren` property for HTML element objects, 205–206
- canvas containing simple chart, 636–637
- canvas skeleton, 635
- character conversions, 892
- `charCode` and `keyCode` property values, 792–793
- checkbox and `onclick` event handler, 680–681
- checkbox object's `checked` property, 100–101
- checked property as conditional, 677
- child elements, 209–210
- child nodes, 207–209
- `className` property for HTML element objects, 211–212
- `clearTimeout()` method of window object, 418–419
- `clientHeight` and `clientWidth` properties for HTML element objects, 212–213
- client-side image map, 628
- `closed` property of window object, 382–383
- color change triggered from pop-up menu, 731–732
- color tweaked for page elements, 525–526
- `componentFromPoint()` method, 279–280
- confirm dialog box, 421–422
- `const` keyword, 1071–1072
- content for scriptable and nonscriptable browsers, 153–154
- content page, 509
- `contentEditable` property for HTML element objects, 214–215
- context-sensitive help, 348–349
- control panel frame, 483
- cookie functions, 536–540
- countdown timer, 418–419
- counting days until Christmas, 938
- CSS image rollovers, 135–137
- `currentTarget` and `eventPhase` properties, 798–799
- cutting and pasting under script control, 336–337
- data validation via `onchange` event handler, 706
- date string for older browser, 935
- date validation in form, 941–944
- decimal-to-hexadecimal converter function, 916–917
- `defaultStatus` property of window object, 384
- directory of current document extracted, 505
- `dispatchEvent()` method, 282–283
- document writing based on user input, 583–584
- document writing example, 583
- document writing example placeholder page, 584
- `document.write()` method on current window, 90
- `document.write()` method used on other window, 91
- DOM content added and replaced, 186
- drag-related event handlers, 341–343
- dynamic properties, 322–323
- error object exception, 995
- event bubbling, 743–744
- event cancellation and redirection in IE4.5+, 746–747
- event cancellation and redirection in W3C DOM, 752–753
- Event Lab in W3C, 265–266
- `eventPhase` and `currentTarget` properties, 798–799
- events checked for key and mouse button pressed, 762–763
- events checked for modifier keys, 760–761
- field selection, 702–703
- file information for web page, 546
- `file` input element object, 738
- `fireEvent()` method for HTML element objects, 285–286
- `firstChild` and `lastChild` properties, 224–225
- `focus()` and `blur()` methods, 273–274
- form, 548–549
- form object and form element passed to function, 105–106
- form object passed as parameter, 648–649
- `form.elements` array, 657
- `form.reset()` and `form.submit()` methods, 660–661
- frame, 375
- frame content table of contents, 482–483
- `frame.scrolling` property, 477
- frameset, 508
- frameset and script for hiding/showing frame, 481–482

Listings (*continued*)

- frameset for property picker, 500
- frameset for URL example, 554–555
- frameset forced to load, 374
- framesetting document, 392
- framesetting document for two-frame window in
 - planetary objects example, 1042–1043
- framing prevented, 373
- function called from event handler, 71
- generalizable function, 1037
- `getBoundingClientRect()` method for HTML
 - element objects, 292–293
- global and local variable scope, 73
- graphical navigation bar, 125
- `hash` property of `location` object, 499
- height and width settings for windows, 395–396
- HTML document, 178
- HTML page with immediate script statements, 50
- IE4+ event coordinate properties, 773–775
- image object and rotating images, 620–622
- image `onload` event handler, 625–626
- image rollovers, 132
- `image.complete`, 613
- `img align` property, 611
- `innerHTML` property for HTML element objects,
 - 228–229
- `innerText` property for HTML element objects,
 - 228–229
- `insertBefore()` method for HTML element
 - objects, 301
- keyboard event handler laboratory, 353–355
- `keyCode` property values, 780–781
- labeled statements, 984–985
- `lastChild` and `firstChild` properties, 224–225
- links with custom Status-bar messages, 412
- local and global variable scope, 73
- location information for different controls, 555–556
- `location.replace()` method, 512
 - for loop, 978
- looping array lookup, 950–951
- main window document generating second window, 402
- `mergeAttributes()` method for HTML element
 - objects, 304–305
- modal dialog box document, 457–460
- modeless dialog box document, 462–464
- mouse tracked while passing over elements, 572–573
- `moveBy()` method of window object, 428–429
- navigation bar, 125
- navigation lab control panel, 516
- navigation lab frameset, 516
- navigation to history item, 519
- nested `if...else` constructions, 973
- nested string methods, 911–912
- NN6+/Moz/Safari event coordinate properties, 796–797
- node-related methods, 316–318
- `<noscript>` tag, 150
- number-formatting routine, 915
- object exception, 996–997
- object property, 1056
- object-oriented planetary data presentation, 1043–1044
- `offsetParent` property for HTML element objects,
 - 240–241
- `onbeforecopy` event handler, 328
- `onbeforepaste` and `onpaste` event handlers,
 - 361–362
- `onbeforeunload()` event handler for window
 - object, 467
- `onblur` and `onfocus` event handlers, 331
- `onchange` event handler for text input object, 706
- `onclick` and `ondblclick` event handlers, 333–334
- `onclick` event handler for radio buttons, 687–688
- `oncopy` and `oncut` event handlers, 336–337
- `ondragenter` and `ondragleave` event handlers, 344
- `onfilterchange` event handler, 346
- `onfocus` and `onblur` event handlers, 331
- `onfocus` event handler for text input object, 704–705
- `onmousedown` and `onmouseup` event handlers,
 - 356–357
- `onmouseover` and `onmouseout` event handlers, 359
- `onpropertychange` property, 363–364
- `onreset` and `onsubmit` event handlers, 662–663
- `onselectstart` event handler for HTML element
 - objects, 366–367
- `onstop` event handler, 586–587
- `open()` method new window laboratory, 433–434,
 - 436–437
- `opener` property of window object, 402–403
- `options[].text` property of select element
 - object, 726
- `options[].value` property of select element
 - object, 726
- `outerHTML` and `outerText` properties, 242–243
- page functionality exercise, 55, 66
- parallel array lookup, 949–950

- parent property of window object, 406–407
- parentTextEdit property for HTML element objects, 245–246
- placeholder page for URL example, 556
- pop-up shown and hidden, 493–4
- precaching images, 131
- prevalidation of form, 107
- print example frameset, 438
- printing control, 438–439
- prompt dialog box, 440
- Property Inspector function, 981–982
- property picker, 501–503
- prototype property for array object, 954
- radio input object, 684
- radio objects, 102
- read-only prototype property added to HTML element objects, 188
- recordNumber property data binding, 248–249
- recursion in function, 1036
- referrer page, 559
- regular expression match workshop, 896
- relatedTarget property of NN6+/Moz/Safari event object, 801–803
- release() method for HTML element objects, 308–310
- reloading, 511
- resizing windows, 442
- reveal transition between images, 859
- reveal transitions chosen between images, 860–861
- script errors, 399–400
- script in Body, 47
- script in Head and Body, 47
- script run from onLoad event handler, 51
- script run from user action, 52
- script1.html source code, 22
- scripts hidden from most old browsers, 48–49
- scripts in Head, 46–47
- scroll values, 591–592
- scrollBy() method of window object, 444, 445–446
- scrolling banner, 413–414
- scrolling of image, 444–445
- scrolling prevention, 470
- scrolling property of frame element object, 477
- select() method of text input object, 702–703
- select object used for navigation, 103–104
- select options modified, 717–719
- select options modified in IE4+, 720–721
- select options modified in W3C DOM, 722
- selected text retrieval, 426
- selectedIndex property of select element object, 728
- selection list, 725
- self property of window object, 411
- server submission action adjustment, 678–679
- setCapture() method for HTML element objects, 308–310
- setExpression() method for HTML element objects, 322–323
- setInterval() method of window object, 448–450
- setTimeout() method of window object, 452
- showModalDialog() main page, 456–457
- showModelessDialog() main page, 460–461
- slicing a string, 901–902
- sorting bound data, 220–221
- srcElement property, 785–786
- status message changes, 413
- status property of window object, 412
- stopping script using onstop event handler, 586–587
- string exceptions, 994
- string handlers, 908–909
- string methods, 910–911
- string object prototype, 889
- string portion, 904–907
- string.replace() and string.search(), 899–900
- style sheets, 569–570
- submission action adjustment, 678–679
- summer games countdown, 939–940
- switch construction, 987–988
- tabIndex property for HTML element objects, 254–255
- table of contents, 509
- target property of NN6+/Moz/Safari event object, 804–805
- text object passed to function, 700
- text objects reset to default values, 695
- text object's value property, 99–100
- text object's value property, 699
- time display, 452
- time stamp on page, 552–553
- timeStamp property of NN6+/Moz/Safari event object, 805–806
- title property for HTML element objects, 257
- toElement and fromElement properties, 778–779
- toString() global function custom method, 1069

- Listings (*continued*)
 - `toString()` global function with `radix` values, 1068
 - variable scope workbench page, 1033
 - W3C event capture and bubble, 748–749
 - W3C Event Lab, 265–266
 - Welcome message, 937
 - `while` loop, 980
 - window chrome, 387–388
 - window object references for, 84
 - window properties, 392
 - window-related properties, 406–407
 - XML data reading script, 874
 - literal notation, using with arrays, 947–948
 - LiveScript, transition to JavaScript, 7
 - load event, using with window object, 87
 - `loadCached()` function, using with precaching images, 131–132
 - loaded value for `readyState` property, description of, 247
 - loading process, event handler for, 468–469
 - loading value for `readyState` property, description of, 247
 - `loadXMLDoc()` function, using in outline-style TOC application, BC430–BC431
 - local scope of variables, explanation, 73
 - local variables
 - explanation of, 72
 - relationship to functions, 1033–1035
 - `localName` attribute object property, description of, 203
 - `localName` property
 - for HTML element objects, 233
 - for W3C DOM Level 2, description of, 181
 - `location` attribute of `open()` method, browsers for and description of, 431
 - location information, displaying for different controls, 555–556
 - `location` object
 - explanation of, 496–498
 - methods of, 496
 - properties for, 496–497
 - references in two-frame browser window, 496
 - role in browser-window hierarchy, 33
 - syntax of, 496
 - using, 87
 - `location` object checks, explanation of, BC348
 - `locationbar` property for window object, explanation of, 387–388
 - `location.replace()` method, invoking, 512
 - `longDesc` property
 - description of, 475
 - for `frame` element object, 475
 - for `iframe` element object, 488
 - for `img` element object, 616
 - lookup table. *See* serverless database
 - `lookupNamespaceURI()` method for HTML element objects, explanation of, 303
 - `lookupPrefix()` method for HTML element objects, explanation of, 303
 - loop counter, using, 976–977
 - `loop` property
 - for `img` element object, 616–617
 - for `marquee` element object, BC18
 - loop traffic, directing with `continue` statement, 979
 - loops. *See also* `for` loop
 - breaking out of, 978
 - `do-while`, 980–981
 - `for`, 975–979
 - `for-in`, 981–982
 - using, 69–70
 - `while`, 979–980
 - lowercase
 - converting strings to, 111
 - using with XHTML style, 23
 - `lowsrc` property for `img` element object, explanation of, 617
 - `lowSrc` property for `img` element object, explanation of, 617
- ## M
- Mac HTML text editors, suggestions for, 18
 - Mac OS authoring environment, setting up, 19–20
 - `mailto` URL, unreliability of, 651–652
 - `makeNewWindow()` function, invoking for
 - `document.write()` method, 91
 - `makeTitleRow()` method, using in order form application, BC416–BC417
 - `map` element object
 - explanation of, 630
 - event handlers for, 630
 - explanation of, 631
 - properties for, 630
 - syntax for, 630
 - `margin` properties, explanation of, 848–849
 - `marginHeight` property
 - for `frame` element object, 475
 - for `iframe` element object, 488

- margins, properties for, 591
- marginWidth property
 - for frame element object, 475
 - for iframe element object, 488
- markerOffset property, explanation of, 841
- marks property, explanation of, 841
- marquee element object, explanation of, BC15
- marquee object properties, controlling, BC16–BC17
- mashups
 - designing, BC519
 - using in Google Maps application, BC514
- mask () filter, description and properties for, 857
- MaskFilter () filter, description and properties for, 864
- match found array object properties, descriptions of, BC256
- matches
 - getting information about, BC250
 - searching with regular expressions, BC249
- Math object
 - explanation of, 919
 - methods for, 920–921
 - properties for, 919–920
 - shortcut for, 921–922
 - syntax for, 919
 - using, 113
- MAX_VALUE property for Number object, explanation of, 923
- maxHeight property, explanation of, 841
- maxLength property for text input object, explanation of, 696
- maxValue placeholder, using with repeat loops, 70
- maxWidth property, explanation of, 841
- media property
 - for document object, 557
 - for link element object, BC95
 - for style element object, 815
 - for styleSheet object, 819
- memory management, handling of, 160
- menu element object, explanation of, BC151
- menubar attribute of open () method, browsers for and description of, 431
- menubar property for window object, explanation of, 387–388
- mergeAttributes () method for HTML element objects, explanation of, 303–305
- message property for error object, explanation of, 999
- meta element object, explanation of, BC96–BC97
- metaKey property for NN6+/Moz/Safari event object, explanation of, 789–790
- method property for form object, explanation of, 659
- methods. *See also* string parsing methods
 - accessing for elements, 199
 - accessing for windows, 82–83
 - adding to working objects, 159
 - for body element object, 587
 - for button element object, 669
 - for Button input object, 669
 - calling directly, 167
 - for canvas element object, 634
 - as commands, 61
 - for custom objects, 1041–1042
 - for dataTransfer object, 777
 - for Date object, 115, 931–934
 - for Dictionary object, 1074
 - for document object, 520–521
 - for document objects, 167–168
 - for Enumerator object, 1075
 - for error object, 997
 - for event object, 759
 - for file input element object, 736
 - for function object, 1023
 - for generic HTML element objects, 196–198
 - for history object, 513
 - invoking for DOM objects, 41
 - for location object, 496
 - for Math object, 920–921
 - for object object, 1055
 - for popup object, 490
 - for Reset input object, 669
 - for select element object, 714
 - for string object, 886
 - for Submit input object, 669
 - testing availability of, 152
 - for text input object, 692
 - for TreeWalker object, 595
 - for VbArray object, 1076
 - for window object, 376–377
 - for XML element reading, 868–869
 - for XMLHttpRequest object, 871
- Methods property for a element object, explanation of, 604
- Microsoft, implementation of JavaScript by, 7
- middle value of align property, description of, 486
- MIME type, specifying for JavaScript, 145
- mime types, verifying, BC181–BC183
- mimeTypes object
 - explanation of, BC174–BC175
 - using, BC180–BC181

- mimeType property
 - for a element object, 604
 - for document object, 557
 - for img element object, 617
 - mimeType property, using with clientInformation and navigator objects, BC165–BC166
 - MIN_VALUE property for Number object, explanation of, 923
 - minHeight property, explanation of, 841
 - minimizable attribute of open() method, browsers for and description of, 431
 - minus (–) connubial operator, operand and result for, 1005
 - minus sign (–) sign, checking for, BC266
 - minWidth property, explanation of, 841
 - “Missing } after function body” error message text, explanation of, BC328
 - miterLimit property for canvas element object, explanation of, 639
 - modal attribute of open() method, browsers for and description of, 431
 - modal dialog box document code sample, 457–460. *See also* dialog boxes
 - modeless dialog box document code sample, 462–464
 - modifier keys, checking events for, 760–761
 - modulo (%) connubial operator, operand and result for, 1005
 - modulo by value (%=) assignment operator, example of, 1008
 - month lengths, checking, BC284–BC285
 - MotionBlur() filter, description and properties for, 864
 - mouse, tracking while passing over elements, 572–573
 - mouse actions, event handlers for, 356–360
 - mouse button presses, checking events for, 762–763
 - mouse button, using onclick event handler with, 332–334
 - mouse events, capturing, 306
 - mouse rollovers, relationship to DOM, 13
 - mouse wheel, event handler for, 360
 - move() method, using with TextRange object, BC78–BC79
 - moveBy() method for window object, explanation of, 427–430
 - moveEnd() method, using with TextRange object, BC79–BC80
 - moveFirst() method for Enumerator object, description of, 1075
 - movement of elements, event handlers for, 360
 - moveNext() method for Enumerator object, description of, 1075
 - moveRow() method, using with table element object, BC132–BC133
 - moveStart() method, using with TextRange object, BC79–BC80
 - moveTo() method for canvas element object, explanation of, 642
 - moveToBookmark() method, using with TextRange object, BC80–BC81
 - moveToPoint() method, using with TextRange object, BC81–BC82
 - Moz1 DOM, description of, 187
 - Moz1.4 browsers, geckoActiveXObject() method in, 425
 - mozBorder properties, explanation of, 849
 - Mozilla, relationship to other browsers, 157
 - Mozilla console windows, error message notification in, BC324
 - Mozilla engine
 - attributes property for, 202
 - browsers based on, 157
 - Mozilla-based browsers
 - viewing script errors in, 53
 - web resource for, 380
 - Mozilla.org, formation of, 13
 - Mozilla’s console, warnings in, BC331
 - Mozilla’s SignTool, downloading, BC352
 - Mozilla’s Venkman Debugger, features of, BC335
 - Moz/NN6+/Safari event object
 - explanation of, 788–789
 - methods of DOM objects, 787–788
 - properties for, 787–788
 - syntax for, 788
 - mozOpacity property, explanation of, 842
 - multidimensional arrays, using, 950–951
 - multiline property, using with RegExp object, BC258
 - multiline property, using with regular expression object, BC255
 - multiple property for select element object, explanation of, 723
 - multiply (*) connubial operator, operand and result for, 1005
 - multiply by value (*=) assignment operator, example of, 1008
 - myAge variable, creating, 59

N

- name attribute, getting elements by, 577
- name attribute object property, description of, 203
- name property
 - for applet object, BC229
 - for button element object, 672

- for `Button` input object, 672
- for `a` element object, 604
- for `embed` element object, BC237
- for `error` object, 999–1000
- for `form` object, 659
- for `frame` element object, 475
- for `iframe` element object, 489
- for `img` element object, 618
- for `meta` element object, BC98
- for `object` element object, BC234
- for `plugin` object, BC179
- for `radio` input object, 685–686
- for `Reset` input object, 672
- for `Submit` input object, 672
- for `text` input object, 697
- for `window` object, 396
- name versus `id` attributes, 37
- `namedItem()` method for `select` element object,
 - explanation of, 730–731
- `nameProp` property
 - for `document` object, 557
 - for `a` element object, 605
 - for `img` element object, 618
- namespaces
 - checking matching of, 302
 - resolving, 568
- `namespaces[]` property for `document` object,
 - explanation of, 557
- `namespaceURI` attribute object property, description of, 203
- `namespaceURI` property
 - for HTML element objects, 233
 - for W3C DOM Level 2, 181
- `NaN` (Not a Number) value
 - checking for, 1065
 - significance of, 918–919
- `NaN` property for `Number` object, explanation of, 924
- `naturalHeight` property for `img` element object,
 - explanation of, 618–619
- `naturalWidth` property for `img` element object,
 - explanation of, 618–619
- `navigate()` method for `window` object, explanation of, 430
- navigation bars, controlling frames with, 124–126
- navigation lab
 - control panel code sample, 516
 - frameset code sample, 516
- navigation to history item code sample, 519
- `Navigator` 3
 - dominance of, 12, 171
 - release of, 169
- `Navigator` 4-only extensions
 - event capture model in, 171
 - layers in, 171
- `navigator` object
 - explanation of, BC154–BC155
 - fetching browser information with, 25
 - role in browser-window hierarchy, 33
 - using, 88
- `navigator` property for `window` object, explanation of, 397
- negation (`val`) connullial operator, operand and result for, 1005
- `NEGATIVE_INFINITY` property for `Number` object,
 - explanation of, 923
- nested `if . . . else` constructions code sample, 973
- nested layers
 - scripting, BC207–BC213
 - visibility relationships of, BC215–BC216
- Netscape
 - relationship to other browsers, 157
 - specifying targets in, BC354–BC355
- Netscape Navigator 2, DOM implemented by, 169
- Netscape Plugin Application Programming Interface (NPAPI),
 - significance of, BC289–BC290
- `netscape` property for `window` object, explanation of, 397
- Netscape signed script policy, explanation of, BC349–BC351
- Netscape/Mozilla-only signed scripts, using `open()` method with, 432–434
- network-oriented information, using `location` object
 - properties with, 497
- `new` object operator, explanation of, 1017
- newsgroups, consulting, 1106
- `newWindow` function, generating windows with, 127
- `newWindow` variable, defining as global variable, 83, 91
- `next` property for `history` object, explanation of, 514
- `nextNode()` method for `TreeWalker` object, explanation of, 597–598
- `nextPage()` method, using with `table` element object, BC133
- `nextPage` property for IE4+ event object, explanation of, 782
- `nextSibling` attribute object property, description of, 203
- `nextSibling()` method for `TreeWalker` object,
 - explanation of, 597

- nextSibling property
 - for HTML element objects, 233–234
 - for W3C DOM Level 2, 181
- NN2, event types in, 763–764
- NN3, event types in, 763–764
- NN3+, modifying select options in, 716–720
- NN4
 - event propagation in, 741–743
 - event types in, 763–764
- NN4 extensions, browser support for, 168
- NN6, event types in, 765–766
- NN6+, table modification methods for, BC109
- NN6+/Moz/Safari event object
 - explanation of, 788–789
 - methods of DOM objects, 787–788
 - properties for, 787–788
 - syntax for, 788
- NO_MODIFICATION_ALLOWED_FOR exception,
 - throwing, 991
- node content
 - generating in W3C DOM, 184
 - replacing in W3C DOM, 185–186
- node hierarchy in W3C DOM, significance of, 178–180
- node maps, use of, 223
- node methods in W3C DOM, explanation of, 183
- Node object property constants, support for, 235
- node objects, copying, 276–277
- node properties in W3C DOM, explanation of, 180–183
- nodeName attribute object property, description of, 203
- nodeName property
 - for HTML element objects, 234
 - for W3C DOM Level 2, 180
- nodes
 - adding user data to, 324
 - checking equality and sameness of, 302
 - child and parent nodes, 39
 - creating dynamic content with, 138–139
 - exchanging positions of, 324–325
 - importing, 578
 - inserting into ranges, BC37–BC38
 - relationship to elements, 868
 - relationship to HTML documents, 179–180
 - removing, 313–314
 - replacing, 315–318
 - role in W3C DOM, 38–39
 - significance of, 179
 - types of, 38–39, 179–180
- nodeType attribute object property, description of, 203
- nodeType property
 - for HTML element objects, 234–236
 - for W3C DOM Level 2, 180, 182
- nodeValue attribute object property, description of, 203
- nodeValue property
 - for HTML element objects, 236–237
 - for W3C DOM Level 2, description of, 180
- noHref property for area element object, explanation of, 630
- non-JavaScript browsers, handling, 48–49
- noResize property for frame element object, explanation of, 476
- normalize() method for HTML element objects,
 - explanation of, 305–306
- <noscript>...</noscript> tags, inclusion in JavaScript-capable browsers, 149
- noShade property, using with hr element object,
 - BC12–BC13
- Not (!) Boolean operator, operand and result for, 1010
- Not a Number (NaN) value
 - checking for, 1065
 - significance of, 918–919
- nowrap property
 - for body element object, 590
 - for td and th element objects, BC145
- NPAPI (Netscape Plugin Application Programming Interface),
 - significance of, BC289–BC290
- null and empty entries, filtering, BC265
- null data type, example and description of, 58
- null value, interpretation relative to window object, 83
- number data type, example and description of, 58
- Number() global function, explanation of, 1066–1067
- Number object, explanation of, 922–923
- number property for error object, explanation of, 1000
- number-formatting routine code sample, 915
- numbers
 - converting strings to, 63, 917–918
 - converting to strings, 62, 63–64, 918
- number.toExponential() method for Number object,
 - explanation of, 924–925
- number.toFixed() method for Number object,
 - explanation of, 924–925
- number.toLocaleString() method for Number object,
 - explanation of, 925

`number.toPrecision()` method for `Number` object, explanation of, 924–925
`number.toString()` method for `Number` object, explanation of, 925–926

O

obfuscators, using, 147

object data type, example and description of, 58

object detection

- using for browser version branching, 151–153
- using for DHTML compatibility, BC366–BC368

“Object doesn’t support this property or method” error message text, explanation of, BC328

object element object, explanation of, BC230–BC231

object exception code sample, 996–997

“Object expected” error message text, explanation of, BC327

object IDs

- assigning, 37
- rules related to, 37

object model families. *See also* W3C DOM

- basic object model, 169–170
- basic object model plus images, 170–171
- Internet Explorer 4+ extensions, 171–174
- Internet Explorer 5+ extensions, 174

object model, hierarchy of, 163–165

object naming, importance in puzzle map DHTML application, BC496

object object

- explanation of, 1055–1057
- methods for, 1055
- properties for, 1055
- syntax, 1055

object operators

- `delete`, 1015
- `in`, 1016
- `instanceof`, 1016
- `new`, 1017
- `this`, 1017–1018

object properties, binding events through, 755–756

object property

- defining getters and setters for, 1050–1051
- using with `applet` object, BC229
- using with `object` element object, BC235

object signing, explanation of, BC350

object-oriented concepts

- adding prototypes, 1052
- prototype inheritance, 1053

object-oriented programming (OOP), explanation of, 1040.

- See also* planetary objects example

object-oriented W3C DOM, significance of, 182

objects. *See also* core language objects; document objects;

- DOM objects; generic HTML element objects

- adding properties and methods to, 159

- assigning functions to custom properties for, 187

- attaching behaviors to, 259–262

- browser document objects, 163–164

- built-in objects for JavaScript, 158

- creating, 1049–1050

- creating array of, 1047–1048

- customizing, 1041–1042

- detaching behaviors from, 312

- embedding in planetary objects example, 1048–1049
- events of, 42

- explanation of, 39, 1040–1041

- methods of, 40–41

- naming, 36–37

- nesting relative to prototype inheritance, 1053–1054

- properties for, 40

- prototype-based inheritance employed by, 159–160

- referencing, 37

- relationships between, BC244–BC248

- separating from language, 11

- using VBScript with, BC186

- `void`, 1020

octal and hexadecimal integers, using, 916–917

`offscreenBuffering` property for `window` object, explanation of, 397

offset properties, availability in Internet Explorer 4+, 172

`offsetHeight` property for HTML element objects, explanation of, 237–238

`offsetLeft` property for HTML element objects, explanation of, 238–239

`offsetParent` property for HTML element objects, explanation of, 239–241

`offsetTop` property for HTML element objects, explanation of, 238–239

`offsetWidth` property for HTML element objects, explanation of, 237–238

`offsetX` property for IE4+ event object, description of, 771–776

`o1` element object, explanation of, BC146

`onabort` event handler for `img` element object, explanation of, 624

`onactivate` event handler for HTML element objects, explanation of, 326–327

- `onafterprint()` event handler for window object, explanation of, 466
- `onafterupdate` event handler
 - for HTML element objects, 327–328
 - for text input object, 703
- `onbeforecopy` event handler code sample, 328
- `onbeforecut` event handler for HTML element objects, explanation of, 328–329
- `onbeforeeditfocus` event handler for HTML element objects, explanation of, 329
- `onbeforepaste` event handler for HTML element objects, explanation of, 329–330
- `onbeforeprint()` event handler for window object, explanation of, 466
- `onbeforeunload()` event handler for window object, explanation of, 467
- `onbeforeupdate` event handler
 - for HTML element objects, 327–328
 - for text input object, 703
- `onblur` event handler
 - for HTML element objects, 330–331
 - for text input object, 703–704
- `oncellchange` event handler for HTML element objects, explanation of, 332
- `onchange` event handler
 - for select element object, 103, 731–732
 - for text input object, 705–706
- `onclick` event handler
 - assigning, 168
 - for button element object, 673–674
 - for Button input object, 673–674
 - for checkbox input object, 679–681
 - for HTML element objects, 332–334
 - for radio input object, 686–688
 - for Reset input object, 673–674
 - for Submit input object, 673–674
- `oncontextmenu` event handler for HTML element objects, explanation of, 334
- `oncontrolselect` event handler for HTML element objects, explanation of, 335
- `oncopy` event handler for HTML element objects, explanation of, 335–337
- `oncut` event handler for HTML element objects, explanation of, 335–337
- `ondataavailable` event handler for HTML element objects, explanation of, 337
- `ondatachanged` event handler for HTML element objects, explanation of, 337
- `ondatasetcomplete` event handler for HTML element objects, explanation of, 337
- `ondblclick` event handler for HTML element objects, explanation of, 337–338
- `ondeactivate` event handler for HTML element objects, explanation of, 326–327
- `ondrag` event handler for HTML element objects, explanation of, 338–343
- `ondragend` event handler for HTML element objects, explanation of, 338–343
- `ondragenter` event handler for HTML element objects, explanation of, 343–344
- `ondragleave` event handler for HTML element objects, explanation of, 343–344
- `ondragover` event handler for HTML element objects, explanation of, 343–344
- `ondragstart` event handler for HTML element objects, explanation of, 338–343
- `ondrop` event handler for HTML element objects, explanation of, 345
- `onerror` event handler
 - for img element object, 624
 - using with runtime errors, 988
 - for window object, 467
- `onerror` property for window object, explanation of, 398–401
- `onerrorupdate` event handler
 - for HTML element objects, 345
 - for text input object, 703
- `onfilterchange` event handler for HTML element objects, explanation of, 345–346
- `onfocus` event handler
 - for HTML element objects, 347
 - for text input object, 703–705
- `onfocusin` event handler for HTML element objects, explanation of, 347
- `onfocusout` event handler for HTML element objects, explanation of, 347
- `onhelp` event handler
 - for HTML element objects, 348–349
 - for window object, 468
- `onkeydown` event handler for HTML element objects, explanation of, 349–351
- `onkeypress` event handler for HTML element objects, explanation of, 349–351
- `onkeyup` event handler for HTML element objects, explanation of, 349–351

- `onlayoutcomplete` event handler for HTML element objects, explanation of, 355
- online documentation, consulting, 1107
- `onLine` property, using with `clientInformation` and `navigator` objects, BC166–BC167
- `onload` event handler
 - advisory about using with frameset frames, 374
 - for `document.write()` method, 91
 - for `img` element object, 625–626
 - for `link` element object, BC96
 - relationship to phantom page syndrome, 173
 - specifying in `<frameset>` tag, 123
 - for `window` object, 468–469
- `onload` property for `window` object, relationship to deferred scripts, 50–51
- `onlosecapture` event handler for HTML element objects, explanation of, 355–356
- `onmousedown` event handler
 - for `button` element object, 674
 - for `Button` input object, 674
 - for `Reset` input object, 674
 - for `Submit` input object, 674
- `onmousedown` event handler for HTML element objects, explanation of, 356–357
- `onmouseenter` event handler for HTML element objects, explanation of, 357–358
- `onmouseleave` event handler for HTML element objects, explanation of, 357–358
- `onmousemove` event handler for HTML element objects, explanation of, 358
- `onmouseout` event handler
 - for HTML element objects, 358–359
 - for `imageOff()` function with, 133–134
- `onmouseover` event handler
 - for HTML element objects, 358–359
 - invoking `imageOn()` function with, 133–134
- `onmouseup` event handler
 - for `button` element object, 674
 - for `Button` input object, 674
 - for HTML element objects, 356–357
 - for `Reset` input object, 674
 - for `Submit` input object, 674
- `onmousewheel` event handler for HTML element objects, explanation of, 360
- `onmove` event handler for HTML element objects, explanation of, 360
- `onmoveend` event handler for HTML element objects, explanation of, 360
- `onmovestart` event handler for HTML element objects, explanation of, 360
- `onpaste` event handler for HTML element objects, explanation of, 360–362
- `onpropertychange` event handler for HTML element objects, explanation of, 363–364
- `onreadystatechange` event handler for HTML element objects, explanation of, 364
- `onreset` event handler `form` object, explanation of, 662–663
- `onresize` event handler
 - for HTML element objects, 364
 - for `window` object, 469
- `onresizeend` event handler for HTML element objects, explanation of, 365
- `onresizestart` event handler for HTML element objects, explanation of, 365
- `onrowenter` event handler for HTML element objects, explanation of, 365
- `onrowexit` event handler for HTML element objects, explanation of, 365
- `onrowsdelete` event handler for HTML element objects, explanation of, 365
- `onrowsinserted` event handler for HTML element objects, explanation of, 365
- `onscroll` event handler
 - for HTML element objects, 365–366
 - for `window` object, 470
- `onselect` event handler for `text` input object, explanation of, 703–704
- `onselectionchange()` event handler for `document` object, explanation of, 585
- `onselectstart` event handler for HTML element objects, explanation of, 366–367
- `onstop()` event handler for `document` object, explanation of, 585–586
- `onsubmit` event handler `form` object, explanation of, 106–107, 663
- `onunload()` event handler for `window` object, explanation of, 470–471
- OOP (object-oriented programming), explanation of, 1040.
See also planetary objects example
- `opacity` property, explanation of, 842
- `open()` method
 - for `document` object, 578–579
 - for `window` object, 430–434, 436–437
 - for `XMLHttpRequest` object, 878–879

- `openDialog()` method for window object, explanation of, 436–437
- `opener` property for window object, explanation of, 126, 401–404
- operand, performing equality comparisons on, 1004
- operators
 - , (comma), 1018
 - ?:, 1019
 - arithmetic, 64
 - assignment, 1002, 1008–1009
 - bitwise, 1002, 1013–1014
 - Boolean, 1002, 1009–1013
 - comparison, 64–65, 1002
 - connubial, 1002, 1005–1007
 - object, 1014–1018
 - objects, 1002
 - precedence of, 1020–1022
 - typeof, 1019
- OPML (Outline Processor Markup Language), using with
 - AJAX outline, BC426
- OPML outliner, preparing, BC427–BC428
- `optgroup` element object, explanation of, 734
- option element object, explanation of, 732–733
- `options[]` property for `select` element object, explanation of, 723–724
- `options[] .defaultSelected` property for `select` element object, explanation of, 724
- `options[] .index` property for `select` element object, explanation of, 724
- `options[] .selected` property for `select` element object, explanation of, 724–725
- `options[] .text` property for `select` element object, explanation of, 726
- `options[] .value` property for `select` element object, explanation of, 726
- `options.add()` method for `select` element object, explanation of, 730
- `options.remove()` method for `select` element object, explanation of, 730
- Or (|) Boolean operator
 - operand and result for, 1010
 - truth table for, 1011
- order form application
 - defining data-entry fields in, BC416–BC417
 - design of, BC412
 - global adjustments in, BC413–BC414
 - HTML aspect of, BC415–BC416
 - math aspect of, BC415
 - writing to screen, BC417–BC418
- origin checks, types of, BC348–BC349
- `originalTarget` property for NN6+/Moz/Safari event object, explanation of, 801
- `orphans` property, explanation of, 852
- `oscpu` property, using with `clientInformation` and `navigator` objects, BC167
- `Outdent` command, parameter for and description of, 575
- `outerHeight` attribute of `open()` method, browsers for and description of, 431
- `outerHeight` property for window object, explanation of, 394–396
- `outerHTML` property for HTML element objects, explanation of, 241–243
- `outerText` property for HTML element objects, explanation of, 241–243
- `outerWidth` attribute of `open()` method, browsers for and description of, 431
- `outerWidth` property for window object, explanation of, 394–396
- Outline Processor Markup Language (OPML), using with
 - AJAX outline, BC426
- outline properties, explanation of, 849–850
- outline-style TOC application
 - assembling outline content for, BC423–BC426
 - implementation plan for, BC419–BC420
 - outline code for, BC421–BC423
 - setting frames for, BC421
- output local variable, purpose of, 73
- `overflow` property, explanation of, 842
- `overflowX` property, explanation of, 842
- `overflowY` property, explanation of, 842
- `ownerDocument` attribute object property, description of, 203
- `ownerDocument` property
 - for HTML element objects, 243
 - for W3C DOM Level 2, 181
- `ownerElement` attribute object property, description of, 203
- `ownerNode` property for `styleSheet` object, explanation of, 820
- `ownerRule` property for `styleSheet` object, explanation of, 820
- `owningElement` property for `styleSheet` object, explanation of, 820

P

- <p> tags, including in documents, 34
- padding properties, explanation of, 850
- page and printing properties, 852–853
- page property, explanation of, 852
- pageBreakAfter property, explanation of, 852
- pageBreakBefore property, explanation of, 852–853
- pageBreakInside property, explanation of, 853
- pages, testing, BC342–BC343. *See also* URLs
- pages property for `styleSheet` object, explanation of, 821
- pageX property for NN6+/Moz/Safari event object, explanation of, 794–797
- pageXOffset property for window object, explanation of, 404–405
- pageY property for NN6+/Moz/Safari event object, explanation of, 794–797
- pageYOffset property for window object, explanation of, 404–405
- paragraph element, adding to document, 34
- paragraph text, adding to documents, 35
- parallel arrays, using, 76–77, 948–951
- param element, explanation of, BC238
- parameter variables, relationship to functions, 1035–1036
- parameters
 - for `document.write()` method, 61
 - for `doScroll()` method for `body` element object, 594
 - of functions, 71–72
 - for functions, 1026–1027
 - naming for functions, 104
 - for TDC (Tabular Data Control), 216
 - using with DOM object methods, 41
- parent and child nodes, roles in W3C DOM, 39
- parent container, use of, 223
- parent object, role in frame object model, 371
- parent property for window object, explanation of, 405–407
- parent reference, using with frames, 373
- parent window, loading framesetting document into, 120–121
- `parentElement()` method, using with `TextRange` object, BC82
- `parentElement` property for HTML element objects, *m* explanation of, 243–244
- parentheses (), using with regular expressions, BC244
- `parentNode` attribute object property, description of, 203
- `parentNode()` method for `TreeWalker` object, explanation of, 597
- `parentNode` property
 - for HTML element objects, 244
 - for W3C DOM Level 2, 180, 183
- parents, inserting elements as, 268–269
- `parentStyleSheet` property
 - for `cssRule` object, 826
 - for `styleSheet` object, 821
- `parentTextEdit` property for HTML element objects, explanation of, 245–246
- parent-to-child references, explanation of, 122
- `parentWindow` property for document object, explanation of, 557–558
- `parse()` method, using with `Date` object, 936
- `parseFloat()` function
 - explanation of, 1066–1067
 - using in data type conversions, 63
 - using in order form application, BC415
- `parseInt()` function
 - description of, 1066–1067
 - using in data type conversions, 63
 - using in serverless database, BC404
 - using with Decision Helper application, BC478
 - using with string conversions, 917
- parsing methods for strings
 - `string.charAt(index)`, 890–892
 - `string.charCodeAt()`, 890–892
 - `string.concat()`, 893
 - `string.fromCharCode()`, 890
 - `string.indexOf()`, 894
 - `string.lastIndexOf()`, 894–895
 - `string.localCompare()`, 895
 - `string.match()`, 895–897
 - `string.replace()`, 897–900
 - `string.search()`, 900
 - `string.slice()`, 900–902
 - `string.split()`, 903
 - `string.substr()`, 903–904
 - `string.substring()`, 905–906
 - `string.toLocaleLowerCase()`, 907
 - `string.toLocaleUpperCase()`, 907
 - `string.toLowerCase()`, 907
 - `string.toStringig()`, 908
 - `string.toUpperCase()`, 907
 - `string.valueOf()`, 908
- password input object, explanation of, 707

- paste edit action, event handler for, 360–362
- pasteHTML () method, using with TextRange object, BC83
- pathname property
 - for a element object, 603
 - for location object, 505
- paths
 - adding rectangles to, 642
 - managing in canvas, 641
- personalBar attribute of open () method, browsers for and description of, 431
- personalbar property for window object, explanation of, 387–388
- phantom page syndrome, occurrence in Internet Explorer 4, 173
- pixel height and width, revealing, 212
- pixel width, controlling, 616
- pixelBottom positioning property, explanation of, 844
- pixelDepth property, using with screen object, BC190
- pixelHeight positioning property, explanation of, 844
- pixelLeft positioning property, explanation of, 844
- pixelRight positioning property, explanation of, 844
- pixelTop positioning property, explanation of, 844
- pixelWidth positioning property, explanation of, 844
- pkcs11 property for window object, explanation of, 383
- placeholders
 - for URL example, 556
 - using # with rollovers, 137
 - using with repeat loops, 70
- planet web page exercise, 79
- planetary objects example. *See also* object-oriented programming (OOP)
 - creating array of objects in, 1047–1048
 - data presentation for, 1043–1044
 - encapsulation in, 1046–1047
 - framesetting document for two-frame window in, 1042–1043
 - using embedded objects in, 1048–1049
- platform equivalency, using for DHTML compatibility, BC365–BC366
- platform property, using with clientInformation and navigator objects, BC167–BC168
- plug-in installation, managing, BC183
- plugin object
 - explanation of, BC177–BC178
 - using, BC180–BC181
- plug-ins
 - API approach toward, BC311–BC313
 - detecting in WinIE, BC183–BC186
 - embedding multiple sounds, BC317–BC320
 - HTML side of, BC310
 - scripting, BC309–BC310
 - using with browsers, 6
 - verifying, BC181–BC183
- plugins property, using with clientInformation and navigator objects, BC168
- plugins [] property for document object, explanation of, 558
- pluginspage property, using with embed element object, BC237
- plus (+) connubial operator, operand and result for, 1005
- plus (+) operator, concatenating strings with, 64
- plus (+) sign
 - after version numbers, 157
 - joining strings with, 62
 - stringing together batches of text with, 25
- poly shape, coordinates for and example of, 629
- polygon shape, coordinates for and example of, 629
- popup object
 - explanation of, 490–491
 - methods of, 490
 - properties for, 490
 - syntax for, 490
- pop-up window, creating, 421–422
- port property for a element object, explanation of, 603
- port property for location object, explanation of, 506
- posBottom positioning property, explanation of, 844–845
- posHeight positioning property, explanation of, 844–845
- position positioning property, explanation of, 845
- positioned elements
 - changing backgrounds of, BC198–BC201
 - layers as, BC197
 - visibility behavior of, BC215–BC216
- positioning properties, 843–845
- positive (+val) connubial operator, operand and result for, 1005
- positive integers, testing for, BC265
- POSITIVE_INFINITY property for Number object, explanation of, 923
- posLeft positioning property, explanation of, 844–845
- posRight positioning property, explanation of, 844–845
- posTop positioning property, explanation of, 844–845
- posWidth positioning property, explanation of, 844–845
- precaching images, using src property for img element object with, 130–132, 619. *See also* images
- precedence of operators, levels of, 1021–1022

- `preference()` method, using with `clientInformation` and `navigator` objects, BC171–BC174
- `prefix` attribute object property, description of, 203
- `prefix` property for HTML element objects, explanation of, 233
- `Prefix` property for W3C DOM Level 2, description of, 181
- `preventDefault()` method for NN6+/Moz/Safari event object, explanation of, 808
- `previous` property for `history` object, explanation of, 514
- `previousNode()` method for `TreeWalker` object, explanation of, 597–598
- `previousPage()` method, using with `table` element object, BC133
- `previousSibling` attribute object property, description of, 203
- `previousSibling()` method for `TreeWalker` object, explanation of, 597
- `previousSibling` property
 - for HTML element objects, 233–234
 - for W3C DOM Level 2, description of, 180
- `print()` method for `window` object, explanation of, 438–439
- printing, event handlers for, 466
- printing and page properties, 852–853
- privilege targets, descriptions of, BC355–BC356
- `PrivilegeManager` errors, handling, BC359
- privileges
 - blending into scripts, BC356–BC357
 - gaining, BC352
- problems, preventing, BC341–BC342. *See also* troubleshooting techniques
- `processData()` function, using with `form` object, 650
- `product` property, using with `clientInformation` and `navigator` objects, BC169
- `productSub` property, using with `clientInformation` and `navigator` objects, BC169
- `profile` property, using with `head` element object, BC91
- programming versus scripting, 54
- prompt dialog box, example of, 86–87
- `prompt()` method for `window` object, explanation of, 439–441
- `prompter` property for `window` object, explanation of, 380
- properties
 - accessing for elements, 199
 - accessing for `form` object, 96–97
 - accessing for windows, 82–83
 - adding to working objects, 159
 - for `anchor` element objects, 600
 - for `area` element object, 626
 - for array objects, 953–955
 - aural, 854
 - for background, 845–846
 - for `body` element object, 587
 - for borders and edges properties, 847–851
 - for `button` element object, 669
 - for `Button` input object, 669
 - for `canvas` element object, 634
 - for `checkbox` input object, 675
 - creating for custom objects, 1041
 - for `cssRule` object, 825
 - customizations for, 545–546
 - for `Dictionary` object, 1074
 - for `doctype` object in NN6+/Moz, 542
 - for `document` object, 520–522
 - for `document` objects, 166–167
 - for edges and borders, 847–851
 - for `a` element objects, 600
 - for `Enumerator` object, 1075
 - for `error` object, 997
 - event handler for, 363–364
 - for `event` object, 759
 - for `fieldset` element object, 663
 - for `file` input element object, 736
 - font and text properties, 833–838
 - for `form` object, 646
 - for `frame` element objects, 471, 478
 - for `function` object, 1023
 - for generic HTML element objects, 196–198
 - for `history` object, 513
 - for IE4+ `event` object, 766–767
 - for IE4+ transition filters, 857–858
 - for IE4-compatible static filter types, 856–857
 - for IE5.5 `DXImageTransform.Microsoft` filters, 862–865
 - for `iframe` element objects, 484
 - for `image` input object, 688
 - for `img` element objects, 608
 - for inline display and layout, 838–843
 - for `label` element object, 665
 - for layout and inline display, 838–843
 - for `legend` element object, 663
 - for `link` element objects, 600
 - for lists, 850–851
 - for `location` object, 496–497
 - looping through, 981–982
 - for `map` element object, 630

- properties (*continued*)
 - for `Math` object, 113, 919–920
 - for NN6+/Moz/Safari event object, 787–788
 - for object object, 1055
 - for `optgroup` element object, 734
 - for `option` element object, 732
 - for pages and printing, 852–853
 - for `popup` object, 490
 - for positioning, 843–845
 - for printing and pages, 852–853
 - for `Reset` input object, 669
 - for scroll bars, 851
 - for `select` element object, 714
 - for `string` object, 886
 - for `style` element object, 814
 - for `styleSheet` object, 816
 - for `Submit` input object, 669
 - for table properties, 851–852
 - for text and font properties, 833–838
 - for `text` input object, 692
 - for `textarea` element object, 708
 - tracking values for, 1056–1057
 - for `TreeWalker` object, 595
 - for `window` object, 376–378
 - for `xml` element object, 869
 - for XML element reading, 868–869
 - for `XMLHttpRequest` object, 871
 - properties and methods, exposing relative to IE behaviors, BC376
 - properties of DOM objects, accessing, 40
 - properties of elements, assigning text to, 25
 - property names, case sensitivity of, 167
 - property picker code sample, 501–503
 - `propertyIsEnumerable()` method for object object, explanation of, 1058
 - `propertyName` property for IE4+ event object, explanation of, 782
 - protected properties and methods, accessing, BC354–BC356
 - `protocol` property
 - for `document` object, 558
 - for `a` element object, 603
 - for `img` element object, 619
 - for `location` object, 506
 - prototype inheritance
 - explanation of, 1053
 - relationship to nested objects, 1053–1054
 - `prototype` property
 - for array object, 954
 - for `RegExp` object, BC259
 - for `string` object, 889–890
 - prototype-based inheritance, use by JavaScript objects, 159–160
 - prototypes, adding in OOP (object-oriented programming), 1052
 - puzzle map game DHTML application. *See also* DHTML (Dynamic HTML)
 - calculating data for x and y coordinates in, BC487
 - `clearInterval()` function in, BC493–BC494
 - custom API for, BC485
 - design of, BC483–BC484
 - `dragIt()` function in, BC490
 - `engage()` function in, BC489
 - `init()` function in, BC494
 - loading `.js` library file for, BC486
 - `for` loop in, BC492
 - main program for, BC485
 - main script for, BC486–BC487
 - `moveHelp()` function in, BC493
 - `onTarget()` function in, BC491, BC495
 - `release()` function in, BC490–BC491
 - `setSelectedMap()` function in, BC488–BC489
 - setting label layer's background color in, BC489
 - `shiftTo()` API function in, BC491
 - `showHelp()` function in, BC492
 - `state` object for, BC488
 - structure of, BC485–BC486
- ## Q
- `q` element object, explanation of, BC4
 - `quadraticCurveTo()` method for canvas element object, explanation of, 640
 - `qualifier` property for IE4+ event object, explanation of, 769
 - query commands, using, 579–580
 - `queryCommandCommandState()` method for `document` object, explanation of, 579–580
 - `queryCommandEnabled()` method
 - for `document` object, 579–580
 - for `TextRange` object, BC84
 - `queryCommandIndeterm()` method
 - for `document` object, 579–580
 - for `TextRange` object, BC84

`queryCommandState()` method, using with `TextRange` object, BC84

`queryCommandSupported()` method
for `document` object, 579–580
for `TextRange` object, BC84

`queryCommandText()` method
for `document` object, 579–580
for `TextRange` object, BC84

`queryCommandValue()` method
for `document` object, 579–580
for `TextRange` object, BC84

quote pair (“ ”), using with `String` objects, 110–113

quote pair (‘ ’), using with `String` objects, 110–113

quotes (“”), including with element IDs, 24

quotes property, explanation of, 834

R

`radio` input object
explanation of, 682–683
using with form controls as objects, 101–102

`radix` values, using with `toString()` function, 1068

random numbers, using `Math` object with, 113, 921

`RandomBars()` filter, description and properties for, 865

`RandomDissolve()` filter, description and properties for, 865

`Range` object
creating, 568
explanation of, BC20–22
versus `TextRange` object, BC59

`rangeCount` property, using with `selection` object, BC45

`ranges`
inserting nodes into, BC37–BC38
working with text ranges, BC23, BC58–BC59

`readOnly` property
for `cssRule` object, 826
for `styleSheet` object, 821
for text input object, 697–698

read-only prototype, adding to HTML element objects, 188

read-only `style` object, returning, 215

read/write access, availability in Internet Explorer 4+, 172

read/write status, determining for `document` object properties, 167

ready state, event handler for, 364

`readyState` property
for HTML element objects, 246–247
for `XMLHttpRequest` object, 875

real-time versus batch validation, BC261–BC263

`reason` property for IE4+ event object, explanation of, 769

`recalc()` method for `document` object, explanation of, 580

`recordNumber` property for HTML element objects, explanation of, 247–249

records, displaying in data collections, 219

`recordset` property for IE4+ event object, explanation of, 769

`rect()` method for `canvas` element object, explanation of, 642

`rect` shape, coordinates for and example of, 629

`rectangle` shape, coordinates for and example of, 629

recursion of functions, explanation of, 1036

references
child-to-child, 122–123
child-to-parent, 122
for control objects in W3C DOM, 177
to form control elements, 647
for form controls as objects, 97–98
for `form` object, 95
for `frame` element object, 123
for frames, 372
for image objects, 129
for `location` object in two-frame browser window, 496
for multiple windows, 126–128
parent-to-child, 122
passing to form elements and forms, 105–106
to `style` element objects, 561
for `window` object, 83
for `window` objects, 84
for `window.alert()` method, 85

`referrer` property for `document` object, explanation of, 558–559

`Refresh` command, parameter for and description of, 575

`refresh()` method,
for `plugin` object, BC179–BC180
for using with `table` element object, BC133

`RegExp` constructor, using with of, BC245

`RegExp` object, explanation of, BC257–BC258

regular expression match workshop code sample, 896

regular expression object, explanation of, BC254

regular expressions
counting metacharacters for, BC243
grouping and backreferencing, BC244
matching metacharacters for, BC242
patterns related to, BC239–BC240
performing matches with, BC248–BC249

- regular expressions (*continued*)
 - positional metacharacters for, BC243
 - relationship to objects, BC244–BC248
 - simple patterns related to, BC241
 - source property for, BC245
 - special characters used with, BC241–BC244
 - using, BC248–BC253
 - using to replace strings, BC252–BC253
- rel property
 - for a element object, 605
 - for link element object, BC95
- relatedTarget property for NN6+/Moz/Safari event object, explanation of, 801–803
- releaseCapture() method for HTML element objects, explanation of, 306
- reload() method for location object, explanation of, 510
- Remove() method for Dictionary object, description of, 1074
- remove() method for select element object, explanation of, 730
- RemoveAll() method for Dictionary object, description of, 1074
- removeAllRanges() method, using with selection object, BC49
- removeAttribute() method for HTML element objects, explanation of, 310
- removeAttributeNode() method for HTML element objects, explanation of, 310–311
- removeAttributeNS() method for HTML element objects, explanation of, 311–312
- removeBehavior() method for HTML element objects
 - code sample, 261–262
 - explanation of, 312
- removeChild() method for HTML element objects
 - explanation of, 312
 - role in W3C DOM Level 2, 183
- removeEventListener() method for HTML element objects, explanation of, 263–266
- removeExpression() method for HTML element objects, explanation of, 312–313
- RemoveFormat command, parameter for and description of, 575
- removeNode() method for HTML element objects, explanation of, 313–314
- removeRange() method, using with selection object, BC49
- repeat loops
 - explanation of, 975–979
 - using, 69–70
- repeat property for IE4+ event object, explanation of, 783
- replace() method for location object, explanation of, 512
- replaceAdjacentText() method for HTML element objects, explanation of, 314–315
- replaceChild() method
 - for HTML element objects, 315
 - for W3C DOM Level 2, 183, 185
- replaceData() method, using with Text and TextNode objects, BC52–BC55
- replaceNode() method for HTML element objects, explanation of, 315–318
- reset event, sending to forms, 662
- Reset input object
 - event handlers for, 669
 - explanation of, 670–671
 - form property for, 671–672
 - method for, 669
 - properties for, 669
 - syntax for, 670
- reset() method for form object, explanation of, 660–661
- resistor calculator
 - arrays for, BC440
 - calculation involved in, BC437–BC438
 - calculations and formatting for, BC440–BC441
 - changing images on the fly, BC442
 - creating select objects for, BC443–BC444
 - drawing initial images for, BC444–BC445
 - graphical user interface ideas for, BC438
 - preloading images for, BC441–BC442
- resizable attribute of open() method, , explanation of, 431
- resizable dialog box feature in IE, description of, 455
- resizeBy() method for window object, explanation of, 441–442
- resized objects, event handlers for, 364–365
- resizeTo() method for window object, explanation of, 441–442
- resizing windows, event handler for, 469
- responseText property for XMLHttpRequest object, explanation of, 875
- responseXML property for XMLHttpRequest object, explanation of, 875

`restore()` method for `canvas` element object, explanation of, 642–643

Return key, using with text boxes, 694

`returnValue` property

- for IE4+ event object, 783–784
- for window object, 408

`rev` property

- for a element object, 605
- for link element object, BC95

`revealTrans()` transition filter, description and properties for, 858

`right` property

- explanation of, 843
- using with `TextRectangle` object, BC86–B88

right shift (>>) operator, operands for, 1014

right shift by value (>=) assignment operator, example of, 1008

right value of `align` property, description of, 486

`rightContext` property, using with `RegExp` object, BC259

`rightMargin` property for `body` element object, explanation of, 589–590

rollover dynamism, accomplishing without scripts, 135–137

rollover IE behavior, example of, BC380–BC383

rollover states, image sizes used with, 130. *See also* image rollovers

rollovers, using `li` elements with, 135–137

`root` method for `TreeWalker` object, explanation of, 597

`rotate()` method for `canvas` element object, explanation of, 643

rotating images and image object code sample, 620–622

`row` property for `frameset` element object, explanation of, 480–483

`RowDelim` parameter for TDC

- description of, 217
- using, 219

`rowIndex` property, using with `tr` element object, BC140

rows. *See also* tables

- event handlers for, 365
- modifying, BC108–BC114
- placing at tops of table columns, BC104

`rows` property

- for `textarea` element object, 710
- for `table` element object, BC127

`rowSpan` property, using with `td` and `th` element objects, BC144

`rubyAlign` property, explanation of, 835

`rubyOverhang` property, explanation of, 835

`rubyPosition` property, explanation of, 835

`rules` property

- for `styleSheet` object, 822
- for `table` element object, BC127–BC130

runtime error, explanation of, 988

runtime versus errors, BC323–BC324

`runtimeStyle` property for HTML element objects, explanation of, 249–250

S

`\S` matching metacharacter, using in regular expressions, BC242

Safari, recording script errors in, 54

Safari errors, notification of, BC324–BC325

Safari/Moz/NN6+ event object

- explanation of, 788–789
- methods of DOM objects, 787–788
- properties for, 787–788
- syntax for, 788

Safari's Drosera Debugger, features of, BC336

`salesrpt.xml` XML report

- constructing table for, BC508–BC510
- converting data for, BC502–BC504
- implementation plan for, BC500
- initialization sequence for, BC501–BC502
- `select` elements in, BC499
- selecting controls for, BC510
- sorting JavaScript database for, BC504–BC505
- structure of, BC498
- `styleSheet` object for, BC500–BC501

same origin policy, explanation of, BC347–BC348

sandbox in Java, explanation of, BC346–BC347

`save()` method for `canvas` element object, explanation of, 642–643

`saveCurrentVisit()` function, using in flag-update application, BC452

saved source files, reloading, 19

save-switch-reload sequence

- considering in authoring environment, 18
- keyboard shortcuts for, 19

`saveType` property for IE4+ event object, explanation of, 784

`scale()` method for `canvas` element object, explanation of, 643

scope of variables, significance of, 72–74

`scopeName` property for HTML element objects, explanation of, 250

- screen object
 - explanation of, BC187
 - role in browser-window hierarchy, 33
- screenLeft property for window object, explanation of, 408–409
- screenTop property for window object, explanation of, 408–409
- screenX attribute of open () method, explanation of, 431
- screenX property
 - for IE4+ event object, 771–776
 - for NN6+/Moz/Safari event object, 794–797
 - for window object, 409
- screenY attribute of open () method, browsers for and description of, 431
- screenY property
 - for IE4+ event object, 771–776
 - for NN6+/Moz/Safari event object, 794–797
 - for window object, 409
- script element object, explanation of, BC98–BC99
- script errors, viewing, 52–54
- <script for> tags, using, 146
- script libraries, using, 148–149
- script references, naming objects for, 36–37
- script statements. *See also* statements
 - encasing in CDATA sections, 148
 - executing, 423
 - execution of, 49–52
 - hiding from older browsers, 146–147
 - relationship to IE behaviors, BC375
- script structure, refining, BC341–BC342
- <script> tag
 - attributes of, 45–46
 - including closing tags with, 45
 - positions of, 46–48
 - using, 23, 45–46
 - using with HTML documents, 145–146
- <script>...</script> tags, inclusion in JavaScript-capable browsers, 149
- script1.html
 - <div> tag in, 23
 - getting browser information, 25
 - HTML in, 23
 - inserting text in, 24–25
 - modifying, 25
 - <script> tag in, 23
 - source code, 22
 - trigger used for running of, 24
 - troubleshooting, 25
- scriptable browsers, modifying content for, 153–154
- ScriptableClock.java/* code sample, BC294
- scripted clock code sample, BC295–BC297
- scripting
 - exposing elements to, 241
 - separating content from, 190
- scripting language, JavaScript as, 158
- scripting strategy, development of, 14–15
- scripting trends
 - handling events, 191
 - separating content from scripting, 190
 - using W3C DOM where possible, 190–191
- scripting versus programming, 54
- scripts. *See also* signed scripts
 - behavior of first script, 21
 - binding events to, 146
 - communication with applets, BC303
 - embedding in HTML documents, 145–149
 - entering and previewing first script, 21–23
 - errors in, 398
 - establishing goals of, 150–151
 - hiding entirely, 147
 - hiding from XHTML validators, 148
 - placement in documents, 45–49
 - running deferred scripts, 50–52
 - running in response to user actions, 52
 - signing, BC352–BC354
 - stopping, 586–587
 - using expressions in, 60–62
 - using repeat loops with, 69–70
- scripts[] property for document object, explanation of, 560
- scroll bar properties, 851
- scroll box, event handler for, 365–366
- scroll () method for window object, explanation of, 443
- scroll property for body element object, explanation of, 591
- scroll values, determining, 591–592
- scrollAmount property, using with marquee element object, BC19
- scrollbars attribute of open () method, browsers for and description of, 431
- scrollbars property for window object, explanation of, 387–388
- scrollBy () method for window object, explanation of, 443–446
- scrollByLines () method for window object, explanation of, 446

- `scrollByPages()` method for window object, explanation of, 446
- `scrollDelay` property, using with marquee element object, BC19
- `scrollHeight` property for HTML element objects, explanation of, 250–251
- scrolling
 - activating unwanted scrolling, 443
 - event handler for, 470
- scrolling banner, creating, 413–414
- scrolling elements, 283–284
- scrolling methods, using with body element object, 593–594
- `scrolling` property
 - for frame element object, 476–477
 - for iframe element object, 489
- `scrollIntoView()` method for HTML element objects, explanation of, 318–319
- `scrollLeft` property
 - for body element object, 591–592
 - for HTML element objects, 251–252
- `scrollMaxX` property for window object, explanation of, 410
- `scrollMaxY` property for window object, explanation of, 410
- `scrollTo()` method for window object, explanation of, 443–444
- `scrollTop` property
 - for body element object, 591–592
 - for HTML element objects, 251–252
- `scrollWidth` property for HTML element objects, explanation of, 250–251
- `scrollX` property for window object, explanation of, 410
- `scrollY` property for window object, explanation of, 410
- search property
 - for a element object, 603
 - for location object, 506–510
- search-and-replace approaches with Undo, BC72–BC76
- `sectionRowIndex` property, using with `tr` element object, BC140–BC141
- security policies, types of, BC347
- `security[]` property for document object, explanation of, 560
- `securityPolicy` property, using with `clientInformation` and `navigator` objects, BC169
- select element
 - explanation of, 714–715
 - using in XML data transformation application, BC499
 - using with precaching images, 131–132
- `select()` method
 - for text input object, 701–703
 - for `TextRange` object, BC84
- select object
 - creating for resistor calculator, BC443–BC444
 - using with form controls as objects, 102–104
- select options
 - modifying in IE4+, 720–721
 - modifying in NN3+ and IE4+, 716–720
 - modifying in W3C DOM, 721–722
- SelectAll command, parameter for and description of, 575
- `selectAllChildren()` method, using with `selection` object, BC50
- `selectedIndex` property for `select` element object, explanation of, 102–104, 728
- `selection` object, explanation of, BC43–BC44
- `selection` property for document object, explanation of, 561
- `selectNode()` method, using with `Range` object, BC38
- `selectNodeContents()` method, using with `Range` object, BC38–BC39
- `selectorText` property for `cssRule` object, explanation of, 826
- `self` property for window object
 - description of, 410–411
 - example of, 411
- `send()` method for `XMLHttpRequest` object, explanation of, 879
- server programming, varieties of, 5–6
- serverless database. *See also* database
 - arrays for, BC402–BC403
 - Body part of, BC407
 - data validation functions for, BC404–BC405
 - developing, BC399–BC400
 - document title for, BC401
 - implementation plan for, BC400
 - searching, BC406
- servers, offloading busy servers with JavaScript, 30
- server-side image map, property for, 617
- `setActive()` method for HTML element objects, explanation of, 319
- `setAttribute()` method for HTML element objects, explanation of, 319–320
- `setAttributeNode()` method for HTML element objects, explanation of, 310–311
- `setAttributeNS()` method for HTML element objects, explanation of, 320–321

- setCapture() method for HTML element objects,
 - explanation of, 306–310
- setCookie() function, calling in Decision Helper application, BC461
- setCookieData() function, using in flag-update application, BC452
- setData() method
 - for clipboardData object, 381
 - for dataTransfer object, 777
- setEnd() method, using with Range object, BC39–BC40
- setEndAfter() method, using with Range object, BC41
- setEndBefore() method, using with Range object, BC41
- setEndPoint() method, using with TextRange object, BC84–BC85
- setExpression() method for HTML element objects,
 - explanation of, 321–324
- setInterval() method for window object, explanation of, 446–450
- setMonth() method, use of zero (0) with, 115
- setMsg() function, using with image rollovers, 134
- setRequestHeader() method for XMLHttpRequest object, explanation of, 879
- setStart() method, using with Range object, BC39–BC40
- setStartAfter() method, using with Range object, BC41
- setStartBefore() method, using with Range object, BC41
- setters and getters, defining for object properties, 1050–1051
- setTimeout() method for window object, explanation of, 450–453
- setUserData() method for HTML element objects,
 - explanation of, 324
- shadow() filter, description and properties for, 857
- Shadow() filter, description and properties for, 865
- shadowBlur property for canvas element object,
 - explanation of, 639–640
- shadowColor property for canvas element object,
 - explanation of, 639–640
- shadowOffsetX property for canvas element object,
 - explanation of, 639–640
- shadowOffsetY property for canvas element object,
 - explanation of, 639–640
- shape property
 - for area element object, 629
 - for a element object, 603
- sheet property, using with link element object, BC95
- shiftKey property
 - for IE4+ event object, 767–768
 - for NN6+/Moz/Safari event object, 789–790
- shiftLeft property for IE4+ event object, explanation of, 768
- show() method for popup object, explanation of, 492–494
- showHelp() method for window object, explanation of, 453
- showModalDialog() main page code sample, 456–457
- showModalDialog() method for window object,
 - explanation of, 453–456
- showModelessDialog() main page code sample, 460–461
- showTip() function, using in Decision Helper application, BC464
- sibling node, explanation of, 234
- sidebar property for window object, explanation of, 380
- signed pages, locking down, BC360
- signed script policy in Netscape, explanation of, BC349–BC351
- signed scripts. *See also* scripts
 - avoiding international characters in, BC360
 - exporting and importing, BC359–BC360
- signed-script examples
 - accessing private browser information, BC357–BC358
 - accessing protected window property, BC358
- signing scripts, BC352–BC354
- SignTool application, downloading, BC352
- size property
 - explanation of, 853
 - for font element object, BC8–9
 - for hr element object, BC13
 - for select element object, explanation of, 729
 - for text input object, 698
- sizeToContent() method for window object, explanation of, 465
- slicing a string code sample, 901–902
- software tools
 - browsers, 18
 - text editors, 17–18
- “<Something> * error message text, explanations of, BC328–BC330
- sorting data, 219
- sounds, scripting, BC317–BC320
- source code, viewing for frames, 375

- source files
 - referencing in `src` attributes, 148
 - reloading saved source files, 19
- source property, using with regular expression object, BC255
- sourceIndex property for HTML element objects, explanation of, 252
- space characters, omitting from variable names, 59
- span property, using with `col` and `colgroup` element objects, BC137–BC138
- specified attribute object property, description of, 203
- splitText () method, using with Text and TextNode objects, BC55–BC56
- square brackets ([])
 - using in Internet Explorer, 202
 - using with repeat loops, 70
- src attribute, referencing source file in, 148
- src property
 - for embed element object, BC238
 - for frame element object, 478
 - for iframe element object, 489–490
 - for image input object, 689
 - for img element object, 619–623
 - for script element object, BC100
 - for xml element, 870
- srcElement property for IE4+ event object, example of, 784–786
- srcFilter property for IE4+ event object, explanation of, 786
- srcUrn property for IE4+ event object, explanation of, 786
- standard output, outputting string of text to, 423
- standby property, using with object element object, BC235
- start () method, using with marquee element object, BC19
- start property
 - for img element object, 623
 - for ol element property, BC146–BC147
- startContainer property, using with Range object, BC25
- startOffset property, using with Range object, BC25–BC26
- startValue placeholder, using with repeat loops, 70
- statement delimiter, treatment of line terminator as, 161
- statements. *See also* labeled statements; script statements
 - execution of, 49–52
 - hiding script statements from older browsers, 146–147
 - uses of, 49
- static tables, generating, BC385–BC388
- static W3C DOM HTML objects, using, 187–188
- status attribute of open () method, browsers for and description of, 431
- status dialog box feature in IE, description of, 455
- status property
 - for XMLHttpRequest object, 876–877
 - window object, 412–414
- statusbar property for window object, explanation of, 387–388
- statusText property for XMLHttpRequest object, explanation of, 877
- sticky wait cursor, fixing, 564
- Stop button
 - clicking, 466
 - event handler for, 585–587
- stop () method
 - for marquee element object, BC19
 - for window object, 466
- stopPropagation () method for NN6+/Moz/Safari event object, explanation of, 809
- strictErrorChecking property for document object, explanation of, 561
- strictly does not equal (!=) comparison operator, operand and result for, 1003
- strictly equals (===) comparison operator, operand and result for, 1003
- string case, changing, 111
- string comparisons, calculating, 1003
- string data type, example and description of, 58
- string exceptions, throwing, 994
- string literals
 - explanation of, 110
 - joining with variables, 884–885
- string methods
 - charAt (), 112–113
 - nesting, 911–912
 - string.indexOf (), 111–112
 - string.substring (), 112
- string object
 - explanation of, 887
 - methods for, 886
 - properties for, 886
 - syntax for, 887
- string object methods, formatting methods, 910–912
- String objects, using, 110–113

- string parsing methods. *See also* methods
 - `string.charAt(index)`, 890–892
 - `string.charCodeAt()`, 890–892
 - `string.concat()`, 893
 - `string.fromCharCode()`, 890
 - `string.indexOf()`, 894
 - `string.lastIndexOf()`, 894–895
 - `string.localCompare()`, 895
 - `string.match()`, 895–897
 - `string.replace()`, 897–900
 - `string.search()`, 900
 - `string.slice()`, 900–902
 - `string.split()`, 903
 - `string.substr()`, 903–904
 - `string.substring()`, 905–906
 - `string.toLocaleLowerCase()`, 907
 - `string.toLocaleUpperCase()`, 907
 - `string.toLowerCase()`, 907
 - `string.toStringig()`, 908
 - `string.toUpperCase()`, 907
 - `string.valueOf()`, 908
- string portion code sample, 904–907
- string utility functions, using, 908–910
- string values
 - assigning to variables, 110
 - versus string objects, BC302–BC303
- string variables, building, 884
- `string.IndexOf()` method, using, 111–112
- strings
 - comparing, 1003
 - components of, 883
 - converting numbers to, 62, 63–64, 918
 - converting to numbers, 63, 917–918
 - joining, 110–111
 - joining with plus (+) symbol, 62
 - replacing via regular expressions, BC252–BC253
 - returning, 325
 - searching, 111–112
 - treating dates as, 934–935
 - using escape codes with, 912
- `string.substring()` method, using, 112
- `Stripes()` filter, description and properties for, 865
- `stripZeros()` function, using in serverless database, BC404
- `stroke()` method for canvas element object, explanation of, 643
- `strokeRect()` method for canvas element object, explanation of, 643
- `strokeStyle` property for canvas element object, explanation of, 640
- style element object
 - explanation of, 812, 814–815
 - properties for, 814
 - references to, 561
 - syntax for, 814
- style object
 - explanation of, 812
 - returning, 215
 - returning with `runtimeStyle` property, 249
 - syntax for, 828
- style object properties
 - color values for, 830–831
 - combination values for, 832
 - explanation of, 829
 - length values for, 830
 - rectangle sides for, 831–832
 - URLs for, 832
- style object property, explanation of, 828
- style properties, reading, 813–814
- style property
 - availability in Internet Explorer 4+, 172
 - for `cssRule` object, 826–827
 - for HTML element objects, 253
- style property settings, changing, 249–250
- `styleFloat` property, explanation of, 842
- stylesheet headings, changing in DHTML, 138
- stylesheet object
 - creating, 568–570
 - explanation of, 816–817
 - methods of DOM objects, 816
 - properties for, 816
 - syntax for, 816
 - for XML data transformation application, BC500–BC501
- stylesheet properties, scriptability of, 173
- `styleSheet` property, using with `link` element object, BC95
- stylesheets
 - importing, 813
 - rules related to, 812
- `styleSheets[]` property for document object, description of, 561
- SubEthaEdit text editor, downloading, 18
- Submit input object
 - event handlers for, 669
 - explanation of, 670–671
 - form property for, 671–672

- method for, 669
 - properties for, 669
 - syntax for, 670
 - submit () method for form object
 - explanation of, 661
 - quirky behavior of, 108
 - using, 106–107
 - submitted forms, inspecting with CGI programs, 30
 - substringData () method, using with Text and TextNode objects, BC52–BC55
 - substrings, extracting copies of, 112–113
 - subtract by value (–) assignment operator, example of, 1008
 - suffixes property, using with mimeType object, BC177
 - summary property, using with table element object, BC130
 - surroundContents () method, using with Range object, BC41–BC42
 - swapNode () method for HTML element objects,
 - explanation of, 324–325
 - switch statement, explanation of, 985–988
 - syntax
 - for anchor element objects, 600
 - for area element object, 627
 - for body element object, 587
 - for button element object, 670
 - for Button input object, 670
 - for canvas element object, 634
 - for checkbox input object, 675
 - for cssRule object, 825
 - for currentStyle object, 828
 - for Date object, 929
 - for document object, 88, 522
 - for a element objects, 600
 - for elementObject.accesskey, 199
 - for error object, 998
 - for fieldset element object, 664
 - for file input element object, 737
 - for filter object, 855
 - for form controls as objects, 98
 - for form object, 646
 - for frame element objects, 472
 - for frameset element object, 478
 - for function object, 1023–1024
 - for functions, 71
 - for generic HTML element objects, 199
 - for history object, 513
 - for IE4+ event object, 767
 - for if constructions, 68
 - for if...else constructions, 69
 - for iframe element object, 485
 - for image input object, 688
 - for img element objects, 608
 - for label element object, 665
 - for legend element object, 664
 - for link element objects, 600
 - for map element object, 630
 - for Math object, 919
 - for NN6+/Moz/Safari event object, 788
 - for Number object, 922
 - for object object, 1055
 - for optgroup element object, 734
 - for option element object, 732–733
 - for radio input object, 682
 - for referencing objects, 37
 - for Reset input object, 670
 - syntax versus runtime errors, BC323–BC324
 - for runtimeStyle object, 828
 - for select element object, 714
 - for string object, 887
 - for style element object, 814
 - for style object, 828
 - for styleSheet object, 816
 - for Submit input object, 670
 - for text input object, 692
 - for textarea element object, 708
 - for TreeWalker object, 595
 - for window object, 83, 378
 - for xml element object, 869
 - for XMLHttpRequest object, 871
 - systemLanguage property, using with
 - clientInformation and navigator objects, BC170
- ## T
- tabIndex property for HTML element objects, explanation of, 253–255
 - table cells
 - modifying content of, BC106–BC108
 - populating, BC106
 - table columns, modifying, BC114–BC117
 - table element object, explanation of, BC118–BC119
 - table frame property values, cycling through, BC124–BC126
 - table object classes, availability in W3C DOM, BC117
 - table object, family hierarchy of, BC104–BC105
 - table of contents code sample, 509
 - table properties, 851–852

- table rows
 - modifying, BC108–BC114
 - placing at tops of table columns, BC104
- tableRules property values, cycling through, BC128–BC130
- tableLayout property, explanation of, 852
- tables
 - constructing for XML data transformation application, BC508–BC510
 - generating dynamic HTML tables, BC393–BC397
 - generating dynamic tables, BC389–BC392
 - generating static tables, BC385–BC388
 - including captions in, BC104–BC105
- Tabular Data Control (TDC)
 - loading data with, 216
 - power of, 219
- tag attributes, binding events through, 754–755
- tagging language, HTML as, 4
- tagName property for HTML element objects, explanation of, 255–256
- tags in HTML
 - adding id attributes to, 36–37
 - use of, 31
- tags () method for HTML element objects, explanation of, 325
- tagUrn property for HTML element objects, explanation of, 250
- target property
 - for a element object, 605–606
 - for base element object, BC92
 - for form object, 659–660
 - for link element object, BC96
 - for NN6+/Moz/Safari event object, 803–805
- targets, specifying in Netscape, BC354–BC355
- tBodies property, using with table element object, BC130
- tbody element object, explanation of, BC133–BC135
- td and th element objects, explanations of, BC142–BC143
- TDC (Tabular Data Control)
 - loading data with, 216
 - power of, 219
- “Test for equality (==) mistyped as assignment (=)?” error
 - message text, explanation of, BC330
- test () method, using with regular expression object, BC257
- text
 - assigning to element properties, 25
 - inserting, 299–300
 - rendering to right or left, 221
 - replacing, 314
 - stringing together batches of, 24
- text and font properties, 833–838
- Text and TextNode objects, explanations of, BC50–BC51
- text boxes, using Enter/Return key with, 694
- text editors
 - choosing, 17–18
 - switching to browsers in Mac OS, 19
- text fields, designing event handlers for, 693
- text HTML-related node type, description of, 180
- text input object
 - event handlers for, 692
 - explanation of, 692–693
 - method for, 692
 - properties, 692
 - syntax for, 692
- text nodes
 - creating, 570
 - explanation of, 184
 - inserting as last child of current element, 266–267
- text nodes, creating, 92
- Text object data method laboratory, BC53–BC55
- text objects
 - behavior of, 99
 - names of, 697
 - resetting to default values, 695
 - setting values of, 699
- text property
 - for body element object, 588
 - for script element object, BC100
 - for select object, 103
 - for TextRange object, BC63
 - for title element object, BC101
- text ranges
 - creating, 245–246
 - working with, BC58–BC59
- text rollover IE behavior, example of, BC380–BC383
- text strings, outputting to standard output, 423
- textAlign property, explanation of, 835
- textAlignLast property, explanation of, 835
- textarea element object
 - carriage returns inside of, 709–710
 - event handlers for, 708
 - explanation of, 709
 - methods of DOM objects, 708
 - properties for, 708
 - syntax for, 708

- `textAutospace` property, explanation of, 835
- `textContent` property for HTML element objects, explanation of, 256
- `textContent` property, relationship to `innerText`, 227
- `textDecoration` property, explanation of, 835
- `textDecorationBlink` property, explanation of, 836
- `text-decoration-line-through` property, explanation of, 836
- `text-decoration-none` property, explanation of, 836
- `text-decoration-overline` property, explanation of, 836
- `text-decoration-underline` property, explanation of, 836
- `textIndent` property, explanation of, 836
- `textJustify` property, explanation of, 836
- `textJustifyTrim` property, explanation of, 836
- `textKashidaSpace` property, explanation of, 836
- `TextOverflow` property, explanation of, 836
- `TextQualifier` parameter for TDC, description of, 217
- `TextRange` object
 - browser compatibility considerations, BC59
 - determining availability of, 231
 - explanation of, BC57–BC58
 - versus `Range` object, BC59
- `TextRange.execCommand()` commands, descriptions of, BC68–BC69
- `TextRectangle` object
 - explanation of, BC86
 - returning array of, 293–294
- text-related input objects, form controls as, 98–100
- `textShadow` property, explanation of, 837
- `texttop` value of `align` property, description of, 486
- `textTransform` property, explanation of, 837
- `textUnderlinePosition` property, explanation of, 837
- `tFoot` element object, explanation of, BC133–BC135
- `tFoot` property, using with `table` element object, BC130–BC131
- `tHead` property, using with `table` element object, BC130–BC131
- `this` keyword
 - relationship to encapsulation, 1046
 - using with `form` object, 650
 - using with functions, 104
- `this` object operator, explanation of, 1017
- thread element object, explanation of, BC133–BC135
- throw statement, using, 993
- time, displaying, 452. *See also* `dateTime` property for HTML element objects
- time and date arithmetic, performing, 936–937
- time intervals, using with `Date` object, 116
- time stamp on page code sample, 552–553
- time tracking, managing in flag-update application, BC448–BC449
- time zones
 - accommodating, 934
 - relationship to GMT, 927–928
- timeouts, setting, 450–453
- times and dates, validating, BC269–BC273
- `timestamp` property for NN6+/Moz/Safari event object, explanation of, 805–806
- `title` element object, explanation of, BC100–BC101
- `title` property
 - for document object, 561–562
 - for HTML element objects, 256–257
 - for `styleSheet` object, 822
- `titlebar` attribute of `open()` method, browsers for and description of, 432
- `toArray()` method for `VbArray` object, description of, 1076
- `toElement` property for IE4+ event object, explanation of, 778–779
- `toggle()` function, using in outline-style TOC application, BC422, BC424, BC426, BC434
- “Too many JavaScript errors” error message text, explanation of, BC331
- `toolbar` attribute of `open()` method, browsers for and description of, 432
- `toolbar` property for window object, explanation of, 387–388
- toolbar-button actions, replicating, 427
- `top` attribute of `open()` method, browsers for and description of, 432
- `top` positioning property, explanation of, 844
- `top` property
 - for `TextRectangle` object, BC86–B88
 - for window object, 415
- `top` reference, using with frames, 373
- `top` value of `align` property, description of, 486
- top-level objects
 - document, 88–92
 - location, 87
 - navigator, 88
 - window, 82–87
- `topMargin` property for `body` element object, explanation of, 589–590

- `toSource()` method for `object` object, explanation of, 1058
 - `toString()` global function, explanation of, 1067–1069
 - `toString()` method
 - for `error` object, 1000
 - for HTML element objects, 325
 - for `object` object, 1058
 - for `Range` object, BC43
 - for `selection` object, BC50
 - `toString()` property for `function` object, explanation of, 1031
 - `tr` element object, explanation of, BC138–BC139
 - `trace()` function
 - debugging with, BC338–BC339
 - invoking, BC340–BC341
 - `trace.js`, preparing documents for, BC339–BC340
 - `translate()` method for `canvas` element object, explanation of, 643
 - transparency of `canvas` content, establishing, 638
 - `TreeWalker` object
 - creating instance of, 571–572
 - explanation of, 595–596
 - methods of, 595
 - properties for, 595
 - syntax for, 595
 - triggers for `script1.html`, significance of, 24
 - troubleshooting techniques. *See also* debugging tools; problems
 - checking HTML tags, BC332
 - checking runtime expression evaluation, BC334
 - commenting out statements, BC334
 - finding out what works, BC333–BC334
 - reopening files, BC333
 - for timing problems, BC332–BC333
 - viewing source code, BC332
 - `trueSpeed` property, using with `marquee` element object, BC19
 - `try-catch-finally` constructions, using, 990–993
 - `type` attribute, using with `<script>` tags, 145
 - `type` property
 - for `button` element object, 672
 - for `Button` input object, 672
 - for `checkbox` input object, 678
 - for `cssRule` object, 827–828
 - for `a` element object, 606
 - for IE4+ event object, 786–787
 - for `image` input object, 689
 - for `li` element property, BC149–BC150
 - for `link` element object, BC96
 - for `mimeType` object, BC176–BC177
 - for NN6+/Moz/Safari event object, 806–807
 - for `object` element object, BC235
 - for `ol` element property, BC147–BC148
 - for `radio` input object, 686
 - for `Reset` input object, 672
 - for `script` element object, BC100
 - for `select` element object, 729
 - for `selection` object, BC45–BC47
 - for `style` element object, 815
 - for `styleSheet` object, 822
 - for `Submit` input object, 672
 - for `text` input object, 698
 - for `ul` element property, BC148–BC149
 - `typeDetail` property, using with `selection` object, BC47
 - `typeof` operator, explanation of, 1019
- ## U
- `unbound()` method for `VArray` object, description of, 1076
 - `ul` element object, explanation of, BC148
 - `UnBookmark` command, parameter for and description of, 575
 - “Uncaught exception” error message text, explanation of, BC331
 - `underscore()`, using with variable names, 59–60
 - `Undo`, using with search-and-replace approaches, BC72–BC76
 - `unescape` global function, explanation of, 1064
 - `unicodeBidi` property, explanation of, 837
 - Uniform Resource Names (URNs), providing address lists as, 204
 - `uninitialized` value for `readyState` property, description of, 247
 - unique identifiers, getting elements by, 576–577
 - `uniqueID` property for HTML element objects, explanation of, 258–259
 - `units` property, using with `embed` element object, BC238
 - `Unlink` command, parameter for and description of, 575
 - unloading, event handler for, 470–471
 - `Unselect` command, parameter for and description of, 575
 - `unselectable` property for HTML element objects, explanation of, 259
 - “Unspecified error” error message text, explanation of, BC331

- “Unterminated string constant” error message text,
 - explanation of, BC328
- “Unterminated string literal” error message text, explanation of, BC328
- `unwatch()` global function, explanation of, 1069
- `unwatch()` method for `object` object, explanation of, 1056–1058
- update expression* item, using with repeat loops, 70
- `updateInterval` property, using with `screen` object, BC190
- uppercase letters, converting strings to, 111
- URLs, determining base URI, 529
- URL example, frameset for, 554–555
- URL property for `document` object, explanation of, 554–556
- `url` property, using with `meta` element object, BC98
- URLs. *See also* pages
 - assigning, 510
 - describing hostnames and ports of, 500–503
 - hostname of, 503–504
 - passing data among pages by means of, 507
 - pathname component of, 505
 - port numbers for, 506
 - protocols of, 506
 - providing to video source files, 614
 - replacing, 512
 - searching, 506–507
 - string encoding and decoding of, 912
 - using hash (#) marks with, 498–500
 - using `location` objects with, 497
- URLs loaded in windows, representing with `location` object, 87
- `URLUnencoded` property for `document` object, explanation of, 562
- `urn` property for `a` element object, explanation of, 606
- URNs (Uniform Resource Names), providing address lists as, 204
- `urns()` method for HTML element objects, explanation of, 326
- U.S. state name sample validation, BC276–BC279
- `UseHeader` parameter for TDC, description of, 217
- `useMap` property
 - for `img` element object, 623–624
 - for `object` element object, BC235
- user actions, triggering scripts from, 52
- user data
 - accessing, 295
 - adding to nodes, 324

- `userAgent` property
 - for `navigator` object, 25
 - for `clientInformation` and `navigator` objects, BC155, BC158–BC162
- `userLanguage` property, using with `clientInformation` and `navigator` objects, BC170
- `userProfile` object, explanation of, BC191–BC193
- `userProfile` property, using with `clientInformation` and `navigator` objects, BC170
- user-selected text, capturing, 425–426
- UTC
 - versus local date methods, 933
 - relationship to GMT, 928
- `UTC()` method, using with `Date` object, 936

V

- `val` (negation) connubial operator, operand and result for, 1005
- validation
 - controlling in Decision Helper application, BC462
 - of date entries in forms, 941–944
 - of dates and times, BC269–BC273
 - designing filters for, BC263–BC264
 - event handlers for, BC286
 - example of presubmission validation, 30
 - of forms, 106–108
 - real-time versus batch validation, BC261–BC263
 - of serverless database, BC404–BC405
- validation functions
 - combining, BC267–BC269
 - customizing, BC266–BC267
- validation samples
 - cross-confirmation fields, BC285–BC286
 - date validation, BC279–BC285
 - U.S. state name, BC276–BC279
- validation solution
 - dispatch mechanism for, BC273–BC275
 - structure of, BC273
- `validation.js` code sample, BC287–BC288
- `vAlign` property, using with `tbody`, `tfoot`, and `thead` element objects, BC135
- `value` attribute object property, description of, 203
- `value` attribute, using with radio input object, 101–102
- `value` property
 - for `button` element object, 672
 - for `Button` input object, 672

- value property (*continued*)
 - for checkbox input object, 678–679
 - for li element property, BC150
 - for radio input object, 686
 - for Reset input object, 672
 - for select element object, 729
 - for Submit input object, 672
 - for text input object, 698–700
 - using with text-related elements, 98–100
 - value types, descriptions of, 58
 - valueOf() method for object object, explanation of, 1058
 - valueOf() property for function object, explanation of, 1031
 - values
 - assigning to variables, 59
 - defining with typeof operator, 1019
 - passing by reference and by value, 160
 - var global statement, explanation of, 1072
 - var keyword, advisory about use of, 59
 - variable scope, relationship to IE behaviors, BC375
 - variable scope workbench code sample, 1033
 - variables
 - assigning string values to, 110
 - assigning values to, 59
 - declaring, 59
 - global and local types of, 72, 159
 - local and global scope of, 1033–1035
 - naming, 59–60
 - parameter variables, 1035–1036
 - scope of, 72–74
 - using with arrays, 77
 - using with expressions, 62
 - using with functions, 72
 - VBAArray object
 - description of, 1075–1076
 - methods for, 1076
 - VBScript, using with objects, BC186
 - vendor property, using with clientInformation and navigator objects, BC169
 - vendorSub property, using with clientInformation and navigator objects, BC169
 - version numbers, plus (+) sign after, 157
 - version property, using with html element object, BC90
 - verticalAlign property, explanation of, 843
 - video clips
 - running, 617
 - using start property for img element object with, 623
 - video source files, providing URLs to, 614
 - view property for NN6+/Moz/Safari event object, explanation of, 807
 - viewer, returning reference for, 542
 - visibility property, explanation of, 843
 - vlinkColor property for document object, explanation of, 524–527
 - void operator, explanation of, 1020
 - vspace property
 - for applet object, BC229
 - for iframe element object, 488
 - for img element object, 616
 - for marquee element object, BC18
 - for object element object, BC234
 - vtext property for body element object, explanation of, 588
- ## W
- \w matching metacharacter, using in regular expressions, BC242
 - \W matching metacharacter, using in regular expressions, BC242
 - W3C DOM. *See also* DOM (document object model); object model families
 - bidirectional event model in, 188–189
 - event listener types, 263
 - event object properties and methods for, 759
 - event propagation in, 748–753
 - event types in, 765–766
 - features missing from, 176
 - generating new node content in, 184
 - hierarchy of nodes in, 178–180
 - HTML practices in, 177
 - innerHTML property, 186–187
 - Levels 0-3 of, 175–176
 - modifying select options in, 721–722
 - node methods in, 183
 - node properties in, 180–183
 - nodes in, 38–39
 - object-oriented nature of, 182
 - replacing node content in, 185–186
 - significance of, 175
 - static W3C DOM HTML objects, 187–188
 - table modification methods for, BC109
 - table object classes for, BC117
 - trend toward using where possible, 190–191
 - underlying architecture of, 178

- W3C DOM Level 2, node object properties for, 180
- W3C DOM `Node` object inheritance tree, 182
- W3C DOM nodes, creating dynamic content with, 138–139
- W3C DOM-compatible browsers, attribute object properties for, 289
- W3C event model, event listeners in, 188–189
- W3C listeners, binding events through, 756–757
- `watch()` global function, explanation of, 1069
- `watch()` method for `Object` object, explanation of, 1056–1057, 1059
- `wave()` filter, description and properties for, 857
- `wave()` filter, description and properties for, 865
- WC3 DOM, browser support for, 168
- Web Forms 2.0, features of, 666–667
- web pages, testing, BC342–BC343. *See also* URLs
- web sites
 - BBedit text editor, 18
 - ECMA-262 specification, 144
 - HTAs (HTML applications) resource, 174
 - Jasob obfuscator, 147
 - JavaScript Obfuscator, 147
 - JavaScript-related resources on, 1107
 - Mozilla-based browsers, 380
 - newsgroups and FAQs, 1106
 - obfuscators, 147
 - online documentation, 1106
 - SubEthaEdit text editor, 18
 - support and updates for book, 1105
- Web technologies
 - CSS (Cascading Style Sheets), 5
 - helpers and plug-ins, 5–6
 - HTML (Hypertext Markup Language), 4–5
 - server programming, 5
- Welcome message code sample, 937
- `whatToShow` method for `TreeWalker` object, explanation of, 597
- `wheelData` property for IE4+ event object, explanation of, 787
- wheeled mice, unwanted scrolling activated by, 443
- `while` loop, explanation of, 979–980
- whitespace, insignificance of, 161
- `whiteSpace` property, explanation of, 837
- `widows` property, explanation of, 852
- width and height of elements, reporting, 237–238
- width and height settings for windows code sample, 395–396
- `width` attribute of `open()` method, browsers for and description of, 432
- width of windows, adjusting, 394–396
- `width` positioning property, explanation of, 844
- `width` property
 - for `applet` object, BC229
 - for `col` and `colgroup` element objects, BC138
 - for `document` object, 550
 - for `embed` element object, BC237
 - explanation of, 843
 - for `frame` element object, 474–475
 - for `hr` element object, BC13
 - for HTML element objects, 225–226
 - for `iframe` element object, 488
 - for `img` element object, 615
 - for `marquee` element object, BC18
 - for `object` element object, BC234
 - for `screen` object, BC188–BC189
 - for `table` element object, BC126–BC127
 - for `td` and `th` element objects, BC145
- window chrome code sample, 387–388
- window features, setting, 430–432
- window names, specifying with `open()` method, 435
- window object checks, explanation of, BC348
- window object methods, invoking with `fullScreen()` function, 167
- window objects
 - confusion about, 369
 - event handlers for, 376–377
 - explanation of, 82, 378–379
 - methods of, 376–377
 - naming, 396
 - properties for, 376–378
 - references for, 82–83
 - relationship to W3C DOM Level 2, 379
 - role in browser-window hierarchy, 33
 - role in frame object model, 371
 - synonym for, 83
 - syntax for, 378
 - using `blur()` method with, 271–272
 - using `focus()` method with, 272
 - using `load` event with, 87
 - using `opener` property with, 126
- window properties and methods, accessing, 82–83
- window properties, displaying, 392
- window property for window object, explanation of, 415
- `window.alert()` method, using, 85
- `window.confirm()` method, using, 85–86
- `window.open()` method, generating windows with, 83

`window.prompt()` method, using, 86–87

windows

- adjusting height and width of, 394–396
- arranging in authoring environment, 19
- controlling appearance of, 430
- creating, 83–85, 436–437
- generating with `newWindow()` function, 127
- loading content into, 435–436
- loading documents into, 430
- references for, 126–128
- repositioning onscreen, 427–430
- resizing, 441–442, 469
- sizing to content, 465

Windows 32-bit operating systems, using Internet Explorer with, 173

Windows authoring environment, setting up, 19

Windows Media Player (WMP), relationship to `geckoActiveXObject()` methods, 425

Winhelp window, opening, 453

WinIE, plug-in detection in, BC183–BC186

WinIE Objects global statement, explanation of, 1073

WinIE Script Debugger, features of, BC335

WinIE4+, keyboard event tasks in, 351–355

WinIE5.5+, filter syntax changes in, 861–865

with statement

- explanation of, 982–983
- using with `Math` object, 921–922

WMP (Windows Media Player), relationship to `geckoActiveXObject()` methods, 425

`wordBreak` property, explanation of, 837

`wordSpacing` property, explanation of, 838

`wordWrap` property, explanation of, 838

workflow, establishing for authoring environment, 18

`wrap` property for `textarea` element object, explanation of, 710

`write()` method

- of document object, 581–585
- using `document.write()` method, 89

`writeln()` method for document object, explanation of, 581–585

`writingMode` property, explanation of, 838

WRONG_DOCUMENT_ERR exception, throwing, 991

X

`x` property

- for IE4+ event object, 771–776
- for `img` element object, 624

XHTML style, case requirements for, 23

XHTML validators, hiding scripts from, 148

XML data reading script code sample, 874

XML data transformation application

- constructing table for, BC508–BC510
- converting data for, BC502–BC504
- implementation plan for, BC500
- initialization sequence for, BC501–BC502
- `select` elements in, BC499
- selecting controls for, BC510
- sorting JavaScript database for, BC504–BC505
- structure of, BC498
- `styleSheet` object for, BC500–BC501

XML data, transforming into viewable HTML, BC524–BC525

XML document, manipulating in Google Maps application, BC523–BC524

`xml` element object

- explanation of, 869–870
- properties for, 869
- syntax for, 869

XML element objects, returning references to, 543

XML element reading, properties and methods for, 868–869

XML specification, relationship to AJAX outline, BC426

XML tags, associating namespace URIs with, 233

`XMLDocument` property for `xml` element, explanation of, 870–871

`xmlEncoding` property for document object, explanation of, 562

`XMLHttpRequest` object

- event handlers for, 871
- explanation of, 871–875
- methods for, 871
- properties for, 871
- syntax for, 871

`xmlStandalone` property for document object, explanation of, 562

`xmlVersion` property for document object, explanation of, 562

XPath expressions, evaluating, 574

`xRay()` filter, description and properties for, 857, 865

Y

`y` property

- for IE4+ event object, 771–776
- for `img` element object, explanation of, 624

Z

zero (0), avoiding with `Date` objects, 115

zero fill by value (`>>=`) assignment operator, example of, 1008

zero fill right shift (`>>>`) operator, operands for, 1014

`zIndex` positioning property, explanation of, 845

`zIndex` values, relationships among, BC216–BC218

`z-lock` attribute of `open()` method, explanation of, 432

`zoom` property, explanation of, 843