

Index

SYMBOLS AND NUMERICS

- &** (ampersand), XML (Extensible Markup Language) character data, 555–556
- &&** (AND), logical operator, 70–72
- '** (apostrophe), XML (Extensible Markup Language) character data, 555–556
- *** (asterisk), repetition character, 304–305
- &**, entity reference, 555–556
- '**, entity reference, 555–556
- >**, entity reference, 555–556
- <**, entity reference, 555–556
- **, space symbol, 23
- <A>** tag, event handlers and the, 170–175
- ** (backslash), escape character, 23
- \'** (backslash apostrophe), escape sequence, 23
- \"** (backslash quote), escape sequence, 23
- ** (double backslash), escape sequence, 23
- \b**
 - escape sequence, 23
 - position character, 305–308
- \\b**, regular expression, 324–329
- ^** (caret), position character, 305–308
- [^...]** (caret and periods in brackets), character class, 300–304
- ©** (copyright symbol), inserting in a string, 23
- { }** (curly braces), marking out a block of code with, 67
- (decrement), operators, 37–39
- \d**, character class, 300–304, 300–304
- \$** (dollar sign), position character, 305–308
- ==** (double equal to sign), operator symbol, 65
- !=** (exclamation point equal to), operator symbol, 65
- \f**, escape sequence, 23
- >** (greater than)
 - operator symbol, 65
 - XML (Extensible Markup Language) character data, 555–556
- >=** (greater than or equal to), operator symbol, 65
- ++** (increment), operators, 37–39
- ** tag, exercise questions, 186, 671
- <** (less than)
 - operator symbol, 65
 - XML (Extensible Markup Language) character data, 555–556
- <=** (less than or equal to), operator symbol, 65
- !** (NOT), logical operator, 70–72
- \n**, escape sequence, 23
- ||** (OR), logical operator, 70–72
- .** (period), character class, 300–304
- [...]** (periods in brackets), character class, 300–304
- +** (plus sign), repetition character, 304–305
- #** (pound sign), CSS (Cascading Style Sheets) and the, 471

? (question mark), repetition character

? (question mark), repetition character, 304–305

“ (quote marks)

escape sequence, 23

strings and, 22

\r, escape sequence, 23

\S, character class, 300–304

; (semicolon)

requirements for the, 660

using the, 8

\t, escape sequence, 23

| (vertical bar), alternation character, 309–310

\W, character class, 300–304

\xNN, escape sequence, 23

A

<A> tag, event handlers and the, 170–175

`abs()` method, overview, 122

absolute positioning, defined, 467

ActiveX controls

adding to a page in Internet Explorer, 602–605

browser differences with, 614–615

changes to Internet Explorer 6 Service Pack 1b and, 617

creating a page with, 713–715

defined, 601–602

exercise questions, 618, 713–715

exercise solutions, 713–715

general use of, 607–617

installing, 606–607

plug-ins versus, 601–602

problems with, 614–617, 652–653

summary, 617–618

using in Internet Explorer 5/6, 622–624

version differences, 616–617

ActiveX objects, overview, 577–579

`ActiveXObject()` object, using the, 577–579

`add()` method, `options[]` array, 220–221

`addBookToBasket()`, function, 683

adding

an ActiveX control to the page in Internet Explorer, 602–605

borders to a Web page, 710

buttons to a pop-up window, 681–686

data to an XML (Extensible Markup Language)

document, 561–566

elements to a document, 516–518

features to a window, 262–264

HTML to a new window, 261–262

links to a pop-up window, 681–686

new options, 216–220

new options with Internet Explorer, 220–221

a plug-in to a page in Firefox, 596–598

style to the DHTML (Dynamic HTML) toolbar, 529–532

style to HTML pages, 459–470

`addPage()`, function, 245–247

`addToBasket()`, function, 685–686

Ajax: A New Approach to Web Applications (Garrett), 621

Ajax (Asynchronous JavaScript + XML)

delays, 653–654

remote scripting and, 621

Same-Origin Policy, 652

summary, 655

`XMLHttpRequest` object, 622

`alert()`

function, 11–12

method in HTML Forms, 214

`align`, attribute, 521

alphanumeric characters, checking for, 301–304

alternation characters, | (vertical bar), 309–310

`altKey` property, `MouseEvent` object, 522–526

& (ampersand), XML (Extensible Markup Language), character data, 555–556

`&`, entity reference, 555–556

AND (&&), logical operator, 71

animation, requisites of, 483

`answerCorrect()` function

Answer Radio Buttons, 229–233

Trivia Quiz, 99–100, 287, 323–329

Apache HTTP Server, Web site, 636

' (apostrophe), XML (Extensible Markup Language), character data, 555–556

`'`, entity reference, 555–556

appearances, changing, 470–478

`appendChild(newNode)` method, `Node` object, 516–518

Apple Safari. See Safari

`apply_templates`, **elements**, 572–574

`appName` **property**, navigator **object** and the, 179–184

array

- alphabetizing an, 134–135
- converting into a single string, 133–134
- copying part of an, 132–133
- creating an, 50–54
- defined, 18, 49
- `document.images`, 672
- finding the number of elements in an, 130–131
- `for...in` loop and, 87–88
- `forms[]`, 188–190
- `frames[]`, 249, 288
- `images[]`, 167–169
- joining an, 131–132
- `links[]`, 169, 668
- `mimeType`s, 600–601
- `navigator.plugins`, 598
- `options[]`, 220–221
- putting in numerical order, 134–135
- putting in reverse order, 135–137
- sorting an, 665
- sorting example, 136–137
- using an, 151–156
- value attribute storage versus, 677

Array object

- `concat()` method, 131–132
- creating an, 107–109
- HTML Forms, 213
- `join()` method, 133–134
- length property, 130–131
- overview, 107, 130, 156
- `reverse()` method, 135–137
- `slice()` method, 132–133
- `sort()` method, 134–135

`AskQuestion.htm`, **creating for the trivia quiz**, 276–278, 283–284

assignment operator

- comparison operator versus, 66
- defined, 26, 36

***** (asterisk), repetition character, 304–305

`async` **property**, using the, 579–580, 717–726

asynchronous

- communication, 720
- requests, 627–629

Asynchronous JavaScript + XML (Ajax)

- delays, 653–654
- remote scripting and, 621
- Same-Origin Policy, 652
- summary, 655
- `XMLHttpRequest` object, 622

`Attr` **object**, high-level DOM **object**, 502

attribute

- `align`, 521
- `bgColor`, 7
- `CHECKED`, 208
- `class`, 475–478
- `classid`, 602–605
- `codebase`, 602–605
- `COLS`, 204–207
- creating an, 519–521
- event handler as an, 170–172
- `font-style`, 461
- `g`, 299
- getting an, 518–521
- `i`, 299
- `ID`, 240–241
- `m`, 299
- `name`, 233
- `onclick`, 170–172
- `pluginspage`, 598–601
- `rows`, 204–207, 240–241
- setting an, 518–521
- `src`, 151
- `style`, 461
- `target`, 188–189
- `type`, 7
- `value`, 677
- `wrap`, 204–207

B

**** (backslash), escape character, 23

\' (backslash apostrophe), escape sequence, 23

\" (backslash quote), escape sequence, 23

\\ (double backslash), escape sequence

\\ (double backslash), escape sequence, 23

\\b

- escape sequence, 23
- position character, 305–308

\\b, **regular expression, 324–329**

\\B, **position character, 305–308**

back button, browser, 653

back() **method, history object, 163**

background color

- changing, 463–464
- example for changing in the browser, 7–8

background-color, **property, 463–464**

backslash, escape sequence, 23

backspace, escape sequence, 23

backward compatible, defined, 13

base DOM objects, overview, 501

Berners-Lee, Tim, HTML creator, 490–492

bgColor, **attribute, 7**

binary numbers, defined, 2

bitwise operators, defined, 70

blocking, pop-ups, 257

blur **event**

- Firefox/Netscape and the, 203
- troubleshooting the, 203

blur() **method**

- <form> elements, 193–194
- summary, 233–234

BOM (Browser Object Model)

- <A> tag, 170–175
- browser properties checking, 176–178
- command window and the, 383
- document object, 165–169
- DOM (Document Object Model) versus, 497
- exercise questions, 667–672
- exercise solutions, 667–672
- history object, 163
- images[] Array, 167–169
- links[] Array, 169
- location object, 163–164
- navigator object, 164, 179–184
- <noscript> tag, 178
- overview, 159–160, 160–161, 489–490

screen object, 164

Boolean data, data type, 24

border, adding to a Web page, 710

, **tag, 665**

break **statement**

- looping and the, 91
- summary, 102
- switch statement, 80

breakpoints, Microsoft Script Debugger and, 380–381

Browser Object Model (BOM)

- <A> tag, 170–175
- browser properties checking, 176–178
- command window and, 383
- document object, 165–169
- Document Object Model (DOM) versus, 497
- exercise questions, 667–672
- exercise solutions, 667–672
- history object, 163
- images[] Array, 167–169
- links[] Array, 169
- location object, 163–164
- navigator object, 164, 179–184
- <noscript> tag, 178
- overview, 159–160, 160–161, 489–490
- screen object, 164
- similarities between the Document Object Model (DOM) and the, 506–507
- summary, 185
- window object, 161–163

browser properties, checking for supported, 176–178

browser support, remote scripting and, 621–622

browser type

- code for checking, 668–671
- exercise questions, 186

browser window, opening a new, 257–261

browsers

- back button, 653
- checking for different, 179–184
- compatibility of, 12–13
- differences between, 13

differences with plug-ins and ActiveX controls, 614–615

displaying errors in, 30–34

example for changing background color in the, 7–8

JavaScript tool, 5

script language defaults, 7

summary, 186

testing order, 625

version checking, 175–184

version compatibility, 12

W3C guidelines and, 160

bubbles **property**, event **object**, **522–526**

building

- a DHTML (Dynamic HTML) toolbar, 528–540
- an XML (Extensible Markup Language) application, 582–592

butAddWedexampleonclick(), **function**, **219–220**

butCalculate_onclick(), **function**, **402–403**

butCheckForm_onclick(), **function**, **201–203**

butCheckfunctiononclick(), **function**, **212–213**

butCheckQ_onclick(), **function**, **283–284**

butCheckValid_onclick(), **function**, **303–304**

butGetText_onclick(), **function**, **269**

butOpenWin_onclick(), **function**, **267–269**

butPlay_onclick(), **function**, **613–614**

butShowVisitedfunctiononclick(), **function**, **255–257**

butToCent_onclick(), **function**, **673–674**

button, adding to a pop-up window, **681–686**

button **property**, mouseEvent **object**, **522–526**

button1_onclick(), **function**, **320–321**, **384–388**

button_mouseHandler(), **function**, **535–537**

buttononCheckQfunctiononclick(), **function**, **229–233**

Buttons

- HTML Forms, 194–197
- summary, 234

C

cache, cleaning the, **408–410**

calcFactorial(), **function**, **400–401**

calculating

- the date, 663–664
- temperature conversion, 657–658
- times tables, 660–663

Call Stack Window, Microsoft Script Debugger, **383–386**

calling

- functions, 93
- methods, 156

cancelable **property**, event **object**, **522–526**

^ (caret), position character, **305–308**

[^...] (caret and periods in brackets), character **class**, **300–304**

carriage return, escape sequence, **23**

Cascading Style Sheets (CSS)

- # (pound sign) and, 471
- hexadecimal values and, 463
- overview, 458–459, 462
- property name guidelines, 471
- rgb() notation, 463–464
- <style> tag, 459–461
- style attribute, 461
- styling properties, 462–470
- url() notation, 464

case

- default, 80
- sensitivity errors, 363–364

case **statements**, switch **statement**, **80**

CDATA, XML (Extensible Markup Language), **556**

CDATASection **object**, high-level DOM **object**, **502**

ceil() **method**, overview, **123**

changeImg(), **function**, **174–175**

character class, defined, **300**

character data, XML (Extensible Markup Language), **555–556**

characters

- escape, 23
- reusing groups of, 310–311

charAt() **method**, String **object**, **291**

charCodeAt() method, String object

`charCodeAt()` **method**, `String` **object**, 291

check box

creating a, 207–214

summary, 234

`Checkbox` **object**, **HTML Forms**, 208–214

`CHECKED`, **attribute**, 208

`checkEmail()`, **function**, 643–644

`checkEmail_callBack()`, **function**, 643–644,
724–726

`checkUsername()`, **function**, 642–644,
650–651

`checkUsername_callBack()`, **function**,
643–644, 650–651, 724–726

child node, **defined**, 499

`chkDaylightSaving_onclick()`, **function**,
345–347

Choose File **dialalox box**, 613

`choosePageframesonchange()`, **function**,
254–257

class

character, 300

defined, 460

defining a, 147–149

`HttpRequest`, 655, 716

`JavaScript`, 146–156

object instances, 149–151

overview, 146–147

`class` **attribute**, **using the**, 475–478

`classId` **attribute**, **using the**, 602–605

clause, **finally**, 404

cleaning, **the cache**, 408–410

clearing, **the temporary Internet file folder in**
Internet Explorer, 408–410

`clearInterval()` **method**

summary, 360

using the, 352–353

`clearTimeout()` **method**, **using the**, 349–352

`click`, **mouse event**, 523–526

client

defined, 3

server versus, 651

`clientX` **property**, `MouseEvent` **object**,
522–526

`clientY` **property**, `MouseEvent` **object**,
523–526

`cloneNode(cloneChildren)` **method**, `Node`
object, 516–518

closing

braces errors, 364

parentheses errors, 365–366

tags versus opening tags, 558

tags in XML (Extensible Markup Language),
552–553

closure, **defined**, 630

`cmdStartQuiz_onclick()`, **function**, 354–359

code

access between frames, 247–257

branching defined, 438

for checking browser type, 670–671

defined, 1

examples opened in Internet Explorer 6 and 7, 6

executing, 1

repeating blocks of, 84–88

stepping through, 375–379

`codebase` **attribute**, **using the**, 602–605

coding. **See also** **coding explanation**

A Simple Form with Validation example, 198–200

A Two-Dimensional Array example, 55–56

Adding and Removing List Options example,
217–218

An Array example, 51–52

Animating Content example, 484–485

Assigning Variables the Values of Other Variables
example, 28–29

Calculations example, 36

Check Boxes and Radio Buttons example,
208–211

Checking a Character's Case example, 113–114

Checking for and Dealing with Different Browsers
example, 179–181

Checking a Passphrase for Alphanumeric Charac-
ters example, 301–302

Checking for Supported Browser Properties
example, 176–177

Concatenating Strings example, 42–43

- Converting a Series of Fahrenheit Values example, 85–86
- Converting Strings to Numbers example, 46–47
- Counting Button Clicks example, 194–195
- Counting Occurrences of Substrings example, 119
- Create HTML Elements and Text with DOM (Document Object Model) JavaScript example, 517
- Creating Attributes example, 519–520
- Creating an IFrame Smart Form example, 646–648
- Creating an XMLHttpRequest Smart Form example, 636–638
- Cross-Browser Image Rollover example, 451–453
- Declaring Variables example, 26–27
- Displaying a Random Image when the Page Loads example, 173–174
- Event Watching example, 205–206
- Fahrenheit to Centigrade example, 40–41
- Fahrenheit to Centigrade Function example, 95 between frames, 241–247
- Image Rollover with Property Event Handlers example, 455–456
- Image Selection example, 168
- Inter-Window Scripting example, 265–267
- JavaScript's Rounding Functions Results Calculator example, 124–125
- Making the Temperature Converter User-Friendly example, 44–45
- Moving an Element Around example, 478–481
- Multiple Conditions example, 73–74
- Multiple Frames example, 238–240
- Navigating Your HTML Document Using the DOM (Document Object Model) example, 512–513
- onmouseup and onmousedown example, 196
- Opening Up New Windows example, 258–260
- Painting the Document Red example, 7
- Replacing Single Quotes with Double Quotes example, 315–317
- Reversing the Order of Text example, 292–294
- Scripting Frames example, 251–254
- Setting Colors According to the User's Screen Color Depth example, 165–166
- Sorting an Array example, 136
- Splitting the Fruit String example, 312
- Splitting HTML example, 319–320
- Storing Previous Quiz Results example, 430–432 for tabulating a file, 585–588
- The Cross-Browser Toolbar example, 544–547
- The forms Array example, 188–189
- The if Statement example, 67–68
- The srcElement Property example, 443–445
- The target Property example, 448–449
- The Toolbar in Firefox and Opera example, 541–543
- The type Property for the Other Browsers example, 447
- The Way Things Flow example, 8–10
- The World Time Method of the Date Object example, 334–336
- try...catch and Throwing Errors example, 397–399
- Updating a Banner Advertisement example, 349–351
- Using the className Property example, 476–477
- Using Cookies to Check for Web Site Changes example, 424–427
- Using the Date Object to Retrieve the Current Date example, 138–139
- Using the DOM (Document Object Model) Event Model example, 525
- Using the Frameset Page as a Module example, 242–244
- Using pow() example, 127–128
- Using the select Element for Date Difference Calculations example, 221–224
- Using setCookie() example, 420–421
- Using the style Object example, 472–474
- Using the switch Statement example, 80–82
- Using the type Property example, 439–441
- World Time Converter (Part I) example, 341–344
- World Time Converter (Part II) example, 353
- Writing the Current Time into a Web Page example, 143–144

coding explanation. See also coding

- A Simple Form with Validation example, 200–203
- A Two-Dimensional Array example, 56–58
- Adding and Removing List Options example, 218–220
- An Array example, 52–54
- Animating Content example, 485–488
- Assigning Variables the Values of Other Variables example, 29
- Calculations example, 36–37
- Check Boxes and Radio Buttons example, 211–214
- Checking a Character's Case example, 114–116
- Checking for and Dealing with Different Browsers example, 181–184
- Checking a Passphrase for Alphanumeric Characters example, 302–304
- Checking for Supported Browser Properties example, 177–178
- Concatenating Strings example, 43
- Converting a Series of Fahrenheit Values example, 86–87
- Converting Strings to Numbers example, 47–48
- Counting Button Clicks example, 195
- Counting Occurrences of Substrings example, 119–120
- Create HTML Elements and Text with DOM (Document Object Model) JavaScript example, 517–518
- Creating Attributes example, 520–521
- creating an IFrame Smart Form example, 649–651
- creating an XMLHttpRequest Smart Form example, 641–644
- Cross-Browser Image Rollover example, 453–455
- Declaring Variables example, 27
- Displaying a Random Image when the Page Loads example, 174–175
- Event Watching example, 206–207
- Fahrenheit to Centigrade example, 41–42
- Fahrenheit to Centigrade Function example, 96
- Image Rollover with Property Event Handlers example, 456–458
- Image Selection example, 168–169
- Inter-Window Scripting example, 267–269
- JavaScript's Rounding Functions Results Calculator example, 125–126
- Making the Temperature Converter User-Friendly example, 45
- Moving an Element Around example, 481–483
- Multiple Conditions example, 74–76
- Multiple Frames example, 240–241
- Navigating Your HTML Document Using the DOM (Document Object Model) example, 514–515
- onmouseup and onmousedown example, 196–197
- Opening Up New Windows example, 260–261
- Painting the Document Red example, 7–8
- Replacing Single Quotes with Double Quotes example, 317–319
- Reversing the Order of Text example, 294–295
- Scripting Frames example, 254–257
- Setting Colors According to the User's Screen Color Depth example, 166–167
- Sorting an Array example, 136–137
- Splitting the Fruit String example, 313–314
- Splitting HTML example, 320–321
- Storing Previous Quiz Results example, 432–434
- for tabulating a file, 589–592
- The Cross-Browser Toolbar, 547–548
- The forms Array example, 189–190
- The if Statement example, 69–70
- The srcElement Property example, 445–446
- The target Property example, 449–450
- The Toolbar in Firefox and Opera example, 543–544
- The type Property for the Other Browsers example, 447–448
- The Way Things Flow example, 11–12
- The World Time Method of the Date Object example, 336–339
- try...catch and Throwing Errors, 400–401
- Updating a Banner Advertisement example, 352
- Using the className Property example, 477–478
- Using Cookies to Check for Web Site Changes example, 427–428

- Using the `Date` Object to Retrieve the Current Date example, 139–141
 - Using the DOM (Document Object Model) Event Model example, 526
 - Using the Frameset Page as a Module example, 245–247
 - Using `pow()` example, 128–129
 - Using the `select` Element for Date Difference Calculations example, 224–227
 - Using `setCookie()` example, 421–422
 - Using the `style` Object example, 474–475
 - Using the `switch` Statement example, 82–83
 - Using the `type` Property example, 441–442
 - World Time Converter (Part I) example, 344–347
 - World Time Converter (Part II) example, 353
 - Writing the Current Time into a Web Page example, 144–145
- color**
- changing background, 463–464
 - changing foreground, 462–463
- `color`, **property, 460–461**
- `colorDepth` **property, screen object, 164**
- `COLS` **attribute**
- overview, 204–207
 - summary, 234
- Command Window**
- BOM (Browser Object Model) and, 383
 - Microsoft Script Debugger, 381–383
- comment**
- defined, 11
 - text defined, 556
- `Comment` **object, high-level DOM object, 502**
- common mistakes. See also errors**
- Case Sensitivity, 363–364
 - Equals Rather than Is Equal To, 365
 - Incorrect Number of Closing Braces, 364
 - Incorrect Number of Closing Parentheses, 365–366
 - Missing Plus Signs During Concatenation, 364–365
 - Undefined Variables, 362–363
 - Using a Method as a Property and Vice Versa, 366
- comparing, strings, 78–79**
- comparison operators**
- assigning the results of comparisons, 66
 - assignment operator versus, 66
 - assignment versus comparison, 66
 - common, 65
 - decision making, 65
 - defined, 64
 - precedence, 65
 - summary, 101
- compatibility**
- browser, 12–13
 - version, 12
- compiled language, interpreted language versus, 2**
- computer language, defined, 1**
- computer requirements, for creating a form with XMLHttpRequest, 635–644**
- `concat()` **method, using the, 131–132**
- concatenation**
- defined, 42
 - errors, 364–365
- conditions**
- defined, 64
 - exercise questions/solutions, 659–663
- constructor**
- class, 147
 - defined, 108
- content, animating, 484–488**
- continue statement**
- looping and the, 91–92
 - summary, 102
- conversion, data type, 45–49**
- `convertToCentigrade()`, **function, 96**
- `cookie`, **property, 422–428**
- cookie pair, defined, 429**
- cookie security, Internet Explorer 6 and Internet Explorer 7, 429–434**
- cookie string**
- defined, 407
 - domain, 418
 - expires, 416
 - name and value, 415–416
 - path, 416–417
 - secure, 418

cookies

- checking for enabled, 429
 - components of, 415–418
 - creating, 408, 418–422
 - defined, 404
 - deleting, 422
 - exercise questions, 435, 697–701
 - exercise solutions, 697–701
 - getting the value of, 422–428
 - limitations of, 428–429
 - security and, 429–434
 - session, 428
 - summary, 434
 - using to load advertisements, 699–701
 - viewing in Firefox, 413–415
 - viewing in Internet Explorer, 408–413
- Cookies dialog box, 414**
- Coordinated Universal Time (UTC)**
- defined, 334
 - summary, 360
- `copyHistory`, **window feature, 262–264**
- copying, part of a string, 120–121**
- © (copyright symbol), inserting in a string, 23**
- `createAttribute(attributeName)` **method**,
Document **object, 516–518**
- `createDocument()`, **function, 578–582**
- `createElement(elementName)` **method**,
Document **object, 516–518**
- `createTextNode(text)` **method**, Document
object, 516–518
- `createToolBar()`, **function, 533, 537**
- `createXmlHttpRequest()`, **function, 623–625**
- `createXmlHttpRequest()` **method, using the,**
629, 631–632, 718–726
- creating**
- an array, 50–54
 - the `AskQuestion.htm` file for the trivia quiz,
283–284
 - attributes, 519–521
 - check box, 207–214
 - class object instances, 149–151
 - cookies, 408, 418–422
 - cross-browser DHTML (Dynamic HTML), 544–548
 - a Date object, 137–138
 - a DTD file, 559–561
 - forms, 188
 - forms with an `iframe`, 644–651
 - forms with `XMLHttpRequest`, 634–644
 - `fraGlobalFunctions` for the trivia quiz,
280–283
 - `fraMenuBar` for the trivia quiz, 279
 - frames-based Web pages, 678–680
 - the `fraQuizPage` file for the trivia quiz,
274–278
 - the `fraTopFrame` file for the trivia quiz,
278–279
 - functions, 92–96
 - the `GlobalFunctions.htm` file for the trivia
quiz, 284–287
 - hidden text boxes, 204
 - HTML files, 239–241
 - links on a Web page, 701–703
 - multi-dimensional arrays, 55–58
 - multiple frames, 238–241
 - objects, 107–109
 - pages, 667–671
 - pages with the RealPlayer plug-in/ActiveX control,
713–715
 - password text boxes, 203–204
 - the `QuizPage.htm` file for the trivia quiz, 283
 - radio buttons, 207–214
 - remote scripting methods, 629–634
 - Reset buttons, 197
 - Submit buttons, 197
 - text boxes, 197
 - a timer-based trivia quiz, 354–359
 - toolbars, 538–540
 - the top window frameset page for the trivia quiz,
273–274
 - tree structures, 708–710
 - trivia quiz answer radio buttons, 229–233
 - trivia quiz forms, 228–229
 - user interfaces, 672–678
 - visitor count pages, 697–699
 - XML (Extensible Markup Language) documents,
557–566, 711
 - `XMLHttpRequest` objects in Internet Explorer 7, Firefox,
Opera, and Safari, 622–624

cross-browser. See also DHTML (Dynamic HTML)

compatible defined, 12
 DHTML creation, 544–548
 event handlers as attributes, 438–439
 retrieval of an XML (Extensible Markup Language) file, 581–582
 event handlers as properties, 455–458
 events in Internet Explorer, 439–446
 events in other browsers, 446–450
 image rollover, 450–455
 overview, 437–438
 XMLHttpRequest object, 622–625

Cross-Browser Image Rollover, example, 451–455**CSS (Cascading Style Sheets)**

(pound sign) and, 471
 hexadecimal values and, 463
 hiding an iframe through, 650
 overview, 458–459
 property name guidelines, 471
 rgb() notation, 463–464
 rule defined, 459
 <style> tag, 459–461
 style attribute, 461
 styling properties, 462–470
 url() notation, 464

ctrlKey **property**, mouseEvent **object**,
 523–526

{ } (curly braces), marking out a block of code
 with, 67

currentTarget **property**, event **object**,
 522–526

D

\D, character class, 300–304

data

adding to an XML (Extensible Markup Language) document, 561–566
 character, 555–556
 converting to numbers, 658–659
 numerical calculations, 35–39
 storing in memory, 24–25
 types of, 21–24

data storage

exercise questions, 657–659
 exercise solutions, 657–659

data type

conversion, 45–49
 exercise questions, 62

database, defined, 24**date**

calculating the, 663–664
 exercise questions, 360
 overview, 333

Date object

calculations and dates, 141–142
 creating a, 137–138
 example, 138–141
 exercise questions, 157, 663–667, 689–693
 exercise solutions, 663–667, 689–693
 get methods, 138–141
 getting date values, 138–141
 getting time values, 142–145
 overview, 107, 156
 setting date values, 141
 setting time values, 145–146

debugging

exercise questions, 405
 Firefox, 388–393

decision making, comparison operators, 65**decisions, exercise questions, 102–103****declaring, variables, 25****decrement (—), operators, 37–39**

default case, switch statement, 80

defining, a class, 147–149**Delete Browsing History dialog box, 409****deleting, cookies, 422****DHTML (Dynamic HTML). See also cross-browser**

accessing elements, 470
 browser compatibility and, 450–455
 changing appearances, 470–478
 creating cross-browser, 544–548
 defined, 437
 example, 483–488
 exercise questions, 488, 701–707
 exercise solutions, 701–707
 positioning and moving content, 478–483
 summary, 488

DHTML (Dynamic HTML) toolbar

- adding style to the, 529–532
- building a, 528–540
- Firefox and Opera, 540–544
- HTML structure, 528–529
- Internet Explorer 5+, 526–540
- storing button information, 532–533
- user interaction, 534–537

dialog box

- Choose File, 613
- Cookies, 414
- Delete Browser History, 409
- Delete Browsing History, 409
- Internet Explorer 6 error message, 33–34
- Internet Options, 409
- Options, 413–414
- Temporary Internet Files and History Settings, 410

directories, **window feature, 262–264**

disabling

- JavaScript in Firefox, 5–6
- JavaScript in Internet Explorer, 6

`displayDogs()`, **function, 589–592**

`DisplayEvent()`, **function, 207**

displaying

- browser errors, 30–34
- a daily message, 582–592
- XML (Extensible Markup Language), 566–576

`displayLinks()`, **function, 668**

<div/> elements

- animating content, 488
- `className` property, 477–478
- content positioning, 467–470
- exercise questions, 703–707
- exercise solutions, 703–707
- moving elements, 481–483

`divAdvert_onMouseOut()`, **function, 474–478**

`divAdvert_onMouseOver()`, **function, 474–478**

`doAnimation()`, **function, 486–488**

document

- defined, 8
- element defined, 554

- flow defined, 458
- navigating the, 509–515
- structure in XML, 554
- valid, 559
- validating an XML (Extensible Markup Language), 564

Document object

- `documentElement`, 512–513
- high-level DOM object, 502
- methods, 516–518
- properties of the, 507
- summary, 185
- using the, 165–169, 668

Document Object Model (DOM)

- accessing elements with the, 470
- Browser Object Model (BOM) versus, 497
- changing appearances with the, 471–478
- defined, 159
- event model, 521–526
- exercise questions, 708–710
- exercise solutions, 708–710
- HTML document tree structure, 497–501
- objects, 501–502
- overview, 489–490
- properties and methods, 502–521
- similarities between the Browser Object model and the, 506–507
- summary, 548
- Web standard, 496–497

Document Type Definition (DTD)

- creating a file, 559–561
- defined, 494
- overview, 559

`documentElement`, **property, 507, 512–513**

`DocumentFragment` **object, high-level DOM object, 502**

`document.getElementById()`, **method, 643–644, 706**

`document.images` **array, using the, 672**

`document.lastmodified`, **property, 427–428**

`DocumentType` **object, high-level DOM object, 502**

`document.write()`
 exercise questions, 102–103
 function, 378–383
 method, 660–661

\$ (dollar sign), position character, 305–308

DOM (Document Object Model)
 accessing elements with the, 470
 BOM (Browser Object Model) versus, 497
 changing appearances with the, 471–478
 defined, 159
 event model, 521–526
 exercise questions, 548–549, 708–710
 exercise solutions, 708–710
 objects, 501–502
 overview, 489–490
 properties and methods, 502–521
 similarities between the BOM (Browser Object Model) and the, 506–507
 summary, 548
 Web standard, 496–497

domain

cookie strings and, 418
 name defined, 412
 name servers defined, 3

`DoPlay()` **method, using the, 715**

== (double equal to sign), operator symbol, 65

`do...while` **loop**

overview, 90–91
 summary, 101

downloading

Microsoft Script Debugger, 367–368
 PHP, 636
 Venkman, 388

Dreamweaver, regular expressions, 297

DTD (Document Type Definition)

creating a file, 559–561
 defined, 494
 overview, 559

dynamic, defined, 12

Dynamic HTML (DHTML). See also cross-browser

accessing elements, 470
 adding a style to the, 529–532
 browser compatibility and, 450–455
 changing appearances, 470–478
 creating cross-browser, 544–548
 defined, 437

example, 483–488
 exercise questions, 488, 701–707
 exercise solutions, 701–707
 positioning and moving content, 478–483
 summary, 488

E

ECMA (European Computer Manufacturers Association), Web standards and the, 493

ECMAScript Web standards

overview, 492–493
 summary, 548

Element object

high-level DOM (Document Object Model) object,
 502

methods, 518–521
 properties of the, 508–509

elements, <form>, 193–194

adding to the document, 516–518
`apply_templates`, 572–574
 defined, 49
`<div/>`, 467–470, 477–478, 481–483,
 485–488, 703–707
 document, 554
`<embed/></embed/>`, 596–601, 607–614
 empty, 553
 finding involved, 528
`<input/>`, 481–483
 nesting in XML, 553
`<noembed/>`, 608–614
 object, 602–605
 opening versus closing tags, 558
`<p/>`, 468–470
`<param/>`, 605–614
 removing from the document, 516–518
 returning, 502–506
 returning a reference to the topmost, 507
 root, 554
`<script/>`, 703
 select, 215–227
 showing/hiding, 465–466
`<tbody/>`, 591–592
 text area, 204–207
`<thead>`, 590–592
`<tr/>`, 590–592
 XML (Extensible Markup Language), 554–555

elements[] array property, <form> elements

`elements[]` **array property, <form> elements, 193**

`elements[]` **property, Form object, 190**

`else if` **statement, example, 78**

`else` **statement**

example, 77–78

summary, 101

`<embed>/<embed/>`, **elements, 596–601, 608–614**

embedding, defined, 596

empty element, defined, 553

enabling, the script debugger in Internet Explorer, 5, 370

`Entity` **object, high-level DOM (Document Object Model) object, 502**

entity references, overview, 555–556

`Entityreference` **object, high-level DOM (Document Object Model) object, 502**

equals/is equal to, errors, 365

error handling

exercise questions, 693–696

exercise solutions, 693–696

summary, 404

errors. See also common mistakes

case sensitivity, 363–364

closing braces, 364

closing parentheses, 365–366

concatenation, 364–365

dealing with, 32–34

defined, 394

displaying in browsers, 30–34

displaying in Internet Explorer, 31–32

displaying in Firefox, 30–31

equals/is equal to, 365

example, 397–401

exercise questions, 405

finally clauses, 404

logic versus syntax, 693–696

methods versus properties, 366

nested `try...catch` statements, 401–403

preventing, 393–394

summary, 404

throwing, 397–401

`try...catch` statements, 394–397

undefined variables, 362–363

`escape()`, **function, 419–420**

escape character, defined, 23

escape sequences

common, 23

Unicode, 23

European Computer Manufacturers Association (ECMA), Web standards and the, 493

event handlers

as attributes, 170–172, 438–439

defined, 170

Image Rollover with Property, 455–458

`onblur`, 198, 233–234

`onchange`, 198

`onclick`, 673–674

`onfocus`, 193–194, 198, 233–234

`onkeydown`, 198

`onkeypress`, 198

`onkeyup`, 198

`onload`, 667–671

`onmouseover/onmouseout`, 477–478, 715

`onreadystatechange`, 627–629

`onselect`, 198

`onsubmit`, 723–726

as properties, 172–175, 455–458

summary, 186

event model, Internet Explorer 5+, 526–528

event watching, example, 204–207

`eventPhase` **property, event object, 522–526**

events. See also Web Page events

defined, 169

firing, 527

in non-Internet Explorer browsers, 446–450

`onload`, 186

`onmousedown/onmouseup`, 196–197

`onmouseover/onmouseout`, 672

select element, 221–227

`submit`, 482

example

A Simple Form with Validation, 198–203

Adding and Removing List Options, 217–220

Animated Advertising, 483–488

Animating Content, 484–488

- Assigning Variables the Values of Other Variables, 28–29
- Changing the Background Color of the Browser, 7–8
- Check Boxes and Radio Buttons, 208–214
- Checking a Character's Case, 113–116
- Checking for and Dealing with Different Browsers, 179–184
- Checking for the Existence of Cookies, 424–428
- Checking a Passphrase for Alphanumeric Characters, 301–304
- Checking for Supported Browser Properties, 176–178
- Concatenating Strings, 42–43
- Converting a Series of Fahrenheit Values, 85–87
- Converting Strings to Numbers, 46–49
- Counting Button Clicks, 194–195
- Counting Occurrences of Substrings, 119–120
- Create HTML Elements and Text with DOM JavaScript, 517–518
- Creating an Array, 51–54
- Creating Attributes, 519–521
- Creating a Multi-Dimensional Array, 55–58
- Creating Multiple Frames, 238–241
- Cross-Browser Image Rollover, 451–455
- Declaring Variables, 26–27
- Displaying a Random Image when the Page Loads, 173–175
- Event Watching, 205–207
- Fahrenheit to Centigrade converter, 40–42
- Fahrenheit to Centigrade Function, 95–96
- forms Array, 188–190
- functions, 95–96
- if Statement, 67–70
- iframe Smart Form, 646–651
- Image Rollover with Property Event Handlers, 455–458
- Image Selection, 168–169
- Inter-Window Scripting, 265–269
- JavaScript's Rounding Functions Results Calculator, 124–126
- Moving an Element Around, 478–483
- Multi-Condition if Statements, 73–76
- Navigating Your HTML Document Using the DOM, 512–515
- Numerical Calculations, 36–37
- onmousedown/onmouseup Events, 196–197
- Opening Up New Windows, 258–261
- Operator Precedence, 40–42
- Parsing a Web Page, 8–12
- pow Method, 127–129
- Replacing Single Quotes with Double Quotes, 315–318
- Reversing the Order of Text, 292–295
- Scripting Frames, 251–257
- Setting Colors According to the User's Screen Color Depth, 165–167
- Sorting an Array, 136–137
- Splitting the Fruit String, 312–314
- Splitting HTML, 319–321
- srcElement Property, 443–446
- for Statement, 85–87
- Storing Previous Quiz Results, 430–434
- switch Statement, 80–84
- Tabulating the Dogs, 585–592
- target Property for Non-Internet Explorer Browsers, 448–450
- temperature converter, 44–45
- The Cross-Browser Toolbar, 544–548
- The Toolbar in Firefox and Opera, 541–544
- The World Time Method of the Date Object, 334–339
- try...catch/throw Statements, 397–401
- Updating a Banner Advertisement, 349–352
- Using the className Property, 476–478
- using the Date Object to Retrieve the Current Date, 138–141
- Using the DOM Event Model, 525–526
- Using the Frameset Page as a Module, 242–247
- Using the Select Element for Date Difference Calculations, 221–227
- Using setCookie(), 420–422
- Using the style Object, 472–475
- Using the type Property, 439–442
- World Time Converter, 341–347, 353
- Writing the Current Time into a Web Page, 143–145
- XMLHttpRequest Smart Form, 636–644
- XSLT Template, 571–574

!= (exclamation point equal to), operator symbol

!= (exclamation point equal to), operator symbol, 65

exception, defined, 394

Exception object, error handling, 395–397

executing code, defined, 1

exercise questions

ActiveX controls, 618, 713–715

BOM (Browser Object Model), 667–672

browser type, 186

conditions, 659–663

cookies, 435, 697–701

data storage, 657–659

data type, 62

date, 360

Date object, 157, 663–667, 689–693

decisions, loops, and functions, 102–103, 659–663

DHTML (Dynamic HTML), 488, 701–707

Document Object Model (DOM), 548–549, 708–710

DOM (Document Object Model), 548–549, 708–710

error handling, 693–696

errors, 405

Extensible Markup Language (XML), 593, 710–713

`fix()` function, 157, 665

frames and windows, 289, 678

`href` property, 186, 667

HTML code, 548–549

HTML Forms, 235

HttpRequest class, 655, 716–726

`` tag, 186, 671

Math object, 663–667

Microsoft Script Debugger, 405, 693

plug-ins, 618, 713–715

pop-up window, 289, 681

`pow()` method, 157, 665

`prompt()` function, 62, 658–659

RegExp() object, 330, 686–689

regular expressions, 330–331, 687–689

remote scripting, 716–726

storing, 157

string manipulation, 686–689

time, 360

timers, 689–693

user interface, 235, 672–678

XML (Extensible Markup Language), 593, 710–713

exercise solutions

ActiveX controls, 713–715

Browser Object Model (BOM), 667–672

conditions, 659–663

cookies, 697–701

data storage, 657–659

Date object, 663–667, 689–693

decisions, loops, and functions, 659–663

Document Object Model (DOM), 708–710

error handling, 693–696

Extensible Markup Language (XML), 710–713

`fix()` function, 665

frames and windows, 678

`href` property, 667

HttpRequest class, 716–726

`` tag, 671

Math object, 663–667

plug-ins, 713–715

pop-up window, 681

`pow()` method, 665

`prompt()` function, 658–659

RegExp() object, 686–689

regular expressions, 687–689

remote scripting, 716–726

string manipulation, 686–689

timers, 689–693

user interface, 672–678

XML (Extensible Markup Language), 710–713

expires, cookie strings and, 416

Extensible Markup Language (XML)

CDATA, 556

character data, 555–556

closing tags, 552–553

comments, 556–557

creating a document, 557–566

displaying, 566–576

document structure, 554

- exercise questions, 593, 710–713
- exercise solutions, 710–713
- formatting, 566–576
- manipulating with JavaScript, 576–592
- nesting elements, 553
- opening tags, 552–553
- overview, 551–552
- style sheets and, 566–568
- summary, 592–593
- template rule, 570
- Web standard, 493–494
- as a well-formed language, 552
- XML (Extensible Markup Language) elements, 554–555
- XSLT transformations, 568–576

Extensible Markup Language (XML) documents

- adding data to, 561–566
- creating, 711
- creating a DTD file, 559–561
- Document Type Definition (DTD), 559
- linking to XSL style sheets, 574–576

Extensible Markup Language (XML) file

- cross-browser retrieval of an, 581–582
- exercise questions, 593
- loading an, 579–581, 711–713
- retrieving in Firefox and Opera, 580
- retrieving in Internet Explorer, 577–579

external, defined, 237

F

`\f`, **escape sequence, 23**

factorial, defined, 399

files. See also Trivia Quiz Files

- creating HTML, 239–241
- Zip, 552

finally clause, error handling, 404

Firefox

- adding a plug-in to the page in, 596–598
- blocking pop-ups in, 257
- the blur event and, 203

- checking for/installing plug-ins in, 598–601
- creating an XMLHttpRequest object in, 624
- Debugging, 388–393
- DHTML (Dynamic HTML) toolbar, 540–544
- disabling JavaScript in, 5–6
- displaying errors in, 30–31
- DOM (Document Object Model) Standards and, 497
- Exception object, 395–397
- retrieving an XML (Extensible Markup Language) file in, 580
- status bar text and, 359
- troubleshooting the blur event, 203
- validating XML (Extensible Markup Language) documents in, 564
- viewing cookies in, 413–415
- viewing the JavaScript Console from, 30
- web site, 5

Firefox 2.0, Show Cookies button, 415

Firefox Venkman, summary, 404

`firstCall`, **function, 385**

`firstChild` **property, Node object, 510–512**

`fix()` **function**

- defining, 128–129
- exercise questions, 157, 665
- exercise solutions, 665–667

floating-point numbers, defined, 22

`floor()` **method, overview, 123**

`focus()` **method**

- <form> elements, 193–194
- summary, 233

`font-family`, **property, 460–461, 464–465**

`font-size`, **property, 460–461, 464–465**

`font-style`

- attribute, 461
- property, 465, 474–475

`font-weight`, **property, 465**

fonts/text, styling, 464–465

<form>/</form> **tag**

- summary, 233
- using the, 188–189, 723–726

<form> elements

<form> **elements**

- `blur()` method, 193–194
- `elements[]` array property, 193
- `focus()` method, 193–194
- `form` property, 193
- HTML elements within, 191–192
- `name` property, 193
- `onfocus/onblur` event handlers, 193–194
- `value` property, 193

for **loop**

- summary, 101
- using the, 578, 661–662, 668
- work flow of the, 85

for **statement**

- example, 85–87, 88
- overview, 84–85, 87–88

foreground color, changing, 462–463

for...in **loop**

- arrays and the, 87–88
- summary, 101

form elements. See HTML forms

form feed, escape sequence, 23

Form **object**

- `elements[]` property, 190
- `reset()` method, 190
- `submit()` method, 190
- summary, 233
- using the, 188–189

form **property**

- <form> elements, 193
- summary, 233

formatting

- XML (Extensible Markup Language), 566–576
- XML (Extensible Markup Language) documents with XSLT transformations, 568–576

forms

- creating, 188, 691–693
- creating with an `iframe`, 644–651
- creating with `XMLHttpRequest`, 634–644
- exercise questions, 360
- HTML, 187–188
- received data, 635
- requesting information for, 634–635

`forms[]` **Array**

- overview, 188–190
- summary, 185

`form_submit()`, **function, 723–726**

`forward()` **method, history object, 163**

`fpCallback`, **function, 630–631**

<frame>/<frameset> **tags, window object, 238–257**

`fraGlobalFunctions`, **creating for the trivia quiz, 280–283**

`fraMenuBar`, **creating for the trivia quiz, 279**

frames

- code access between, 247–257
- coding between, 241–247
- exercise questions, 289, 678
- exercise solutions, 678–680
- ID attribute, 240–241
- multiple, 238–241
- overview, 237
- scripting, 251–257
- security, 270–271
- summary, 288
- trivia quiz, 272–273
- window object and, 238–257

`frames[]` **array**

- summary, 288
- window object, 249

frames-based, Web page creation, 678–680

frameset, diagrammed, 248

`fraQuizPage`, **creating for the trivia quiz, 274–278**

`fraTopFrame`, **creating for the trivia quiz, 278–279**

`fromCharCode()` **method, overview, 117**

`fromElement` **property, event object, 528**

function

- calling a, 93
- closure, 630
- creating a, 92–96
- creating for the trivia quiz, 97–100
- defined, 11, 92
- exercise questions, 102–103, 659–663
- exercise solutions, 659–663

- invoking a, 93
- overview, 92
- parameter passing, 93–94
- summary, 102
- function (ActiveX and Plug-Ins)**
 - play(), 715
 - window_onload(), 715
- function (ActiveX and Remote Scripting), but-**
 - Play_onclick(), 613–614
- function (Ajax and Remote Scripting)**
 - checkEmail(), 643–644
 - checkEmail_callBack(), 643–644, 724–726
 - checkUsername(), 642–644, 650–651, 724–726
 - checkUsername_callBack(), 643–644, 650–651
 - createXmlHttpRequest(), 623–625
 - form_submit(), 723–726
 - fpCallback, 630–631
 - handleData(), 633–634
 - oHttp_readyStateChange(), 628–629
 - request_readystatechange(), 630–631, 720–726
- function (Basic JavaScript), alert(), 11–12**
- function (Browser Programming)**
 - changeImg(), 174–175
 - displayLinks(), 668
 - getBrowserName(), 179–184, 670–671
 - getBrowserVersion(), 179–184
 - linkSomePage_onclick(), 172–173
- function (Common Mistakes, Debugging, and Error Handling)**
 - butCalculate_onclick(), 402–403
 - button1_onclick(), 384–388
 - calcFactorial(), 400–401
 - document.write(), 378–383
 - firstCall(), 385
 - secondCall(), 385
 - thirdCall(), 385
 - writeTimesTable(), 373–374, 390–393
- function (Cookies)**
 - escape(), 419–420
 - getCookieValue(), 431–434
 - getQuestion(), 431–434
 - setCookie(), 420–422, 431–434
 - unescape(), 419–420
- function (Data Types and Variables)**
 - isNaN(), 48–49
 - parseFloat(), 46–48, 658–659
 - parseInt(), 46–48
 - prompt(), 41, 62, 658–659
- function (Date, Time, and Timers)**
 - chkDaylightSaving_onclick(), 345–347
 - cmdStartQuiz_onclick(), 354–359
 - getQuestion(), 358–359
 - getTimeString(), 346–347
 - resetQuiz(), 355–359
 - switchImage(), 352
 - updateTime(), 345–347, 353
 - updateTimeLeft(), 356–359
 - updateTimeZone(), 344–347
 - window_onload(), 352, 353
- function (Decisions, Loops, and Functions)**
 - answerCorrect(), 99–100
 - convertToCentigrade(), 96
 - document.write(), 102–103
 - isNaN(), 663
 - Number(), 69
 - prompt(), 69
 - writeTimesTable(), 661–662
- function (Dynamic HTML)**
 - divAdvert_onMouseOut(), 474–478
 - divAdvert_onMouseOver(), 474–478
 - doAnimation(), 486–488
 - image_EventHandler(), 445–446, 449–458
 - moveBox(), 482–483
 - moveDiv(), 705–707
 - paragraph_eventHandler(), 441–442
 - showBoxOne(), 703
 - showBoxTwo(), 703
 - startTimer(), 705–707
- function (Dynamic HTML in Modern Browsers)**
 - button_mouseHandler(), 535–537
 - createToolBar(), 533, 537
 - handleClick(), 526

function (HTML Forms)

function (HTML Forms)

answerCorrect(), 229–233
butAddWedexampleonclick(), 219–220
butCheckForm_onclick(), 201–203
butCheckfunctiononclick(), 212–213
butToCent_onclick(), 673–674
buttononCheckQfunctiononclick(),
229–233
DisplayEvent(), 207
getMonth(), 226
getQuestion(), 229–233
myButton_onclick(), 195
myButton_onmousedown(), 196–197
myButton_onmouseup(), 196–197
radCPUSpeed_onclick(), 212
recalcDateDiff(), 226
txtAge_onblur(), 201–203
txtName_onchange(), 201–203
updateOrderDetails(), 677–678
windowfunctiononload(), 189–190
windowonload(), 225–226
writeMonthOptions(), 225
writeOptions(), 224

**function (Object-Based Language), fix,
128–129, 157**

function (String Manipulation)

answerCorrect(), 323–329
butCheckValid_onclick(), 303–304
button1_onclick(), 320–321
getAnswer(), 328–329
getQuestion(), 323–329
regExpIs_valid(), 302–304
replaceQuote(), 317–318
splitAndReverseText(), 294–295

function (Windows and Frames)

addBookToBasket(), 683
addPage(), 245–247
addToBasket(), 685–686
answerCorrect(), 285–287
butCheckQ_onclick(), 283–284
butGetText_onclick(), 269
butOpenWin_onclick(), 267–269

butShowVisitedfunctiononclick(),
255–257
choosePageframesonchange(), 254–257
getQuestion(), 285–287
removeItem(), 683
returnPagesVisited(), 245–247
setFrameAndPageControls(), 255–257
showBasket(), 684
updateOrderDetails(), 679–680
windowonload(), 241
window_onunload(), 269

function (XML)

createDocument(), 578–582
displayDogs(), 589–592
getDailyMessage(), 583–585
xmlDoc_readyStateChange(), 580

G

g, attribute, 299, 330

garbage collection, defined, 27

**Garrett, Jesse James, *Ajax: A New Approach to
Web Applications*, 621**

getAnswer(), function, 328–329

**getAttribute(attributeName) method,
Element object, 519–521**

**getBrowserName(), function, 179–184,
670–671**

getBrowserVersion(), function, 179–184

getCookieValue(), function, 431–434

getDailyMessage(), function, 583–585

getDate() method, overview, 138

getDay() method

overview, 138

using the, 692–694

getElementsById() method

DOM method, 486–488, 503–504

using the, 470

**getElementsByTagName() method, DOM
object, 505–506, 590–592**

getFullYear() method, overview, 138

getHours() method, overview, 142–145

getMilliseconds() **method, using the, 142–145**

getMinutes() **method, overview, 142–145**

getMonth(), **function, 226**

getMonth() **method**
 overview, 138
 using the, 664

getQuestion(), **function, 229–233, 285–287, 323–329, 358–359, 431–434**

getSeconds() **method, using the, 142–145**

getTimeString(), **function, 346–347**

getTimezoneOffset() **method**
 Date object, 334–339
 summary, 360

getting, attributes, 518–521

getUTCDate() **method, using the, 340–348**

getUTCDay() **method, using the, 340–348**

getUTCFullYear() **method, using the, 340–348**

getUTCHours() **method, using the, 340–348**

getUTCMilliseconds() **method, using the, 340–348**

getUTCMinutes() **method, using the, 340–348**

getUTCMonth() **method, using the, 340–348**

getUTCSeconds() **method, using the, 340–348**

global
 match attribute, 299
 object defined, 161
 scope defined, 96

global variable, switchDirection, 486–488

GlobalFunctions.htm, **creating for the trivia quiz, 284–287**

GMT (Greenwich Mean Time), defined, 334

go() **method, history object, 163**

Google Maps, remote scripting and, 620

Google Suggest, remote scripting and, 620–621

> (greater than)
 operator symbol, 65
 XML (Extensible Markup Language) character data, 555–556

>= (greater than or equal to), operator symbol, 65

Greenwich Mean Time (GMT), defined, 334

grouping, regular expressions, 308–311

>;, **entity reference, 555–556**

H

handleClick(), **function, 526**

handleData(), **function, 633–634**

hasChildNodes() **method, Node object, 516–518**

height, **window feature, 262–264**

height **property, screen object, 164**

hexadecimal values, assigning colors with, 463

hidden-frame technique, process of the, 644–651

hidden text box, creating a, 204

hiding
 elements, 465–466
 an iframe, 650

high-level DOM objects, overview, 501–502

history object
 overview, 163
 summary, 185

href property
 exercise questions, 186, 667
 exercise solutions, 667–668
 location object, 164

HTML
 adding to a new window, 261–262
 code exercise questions, 548–549
 comments versus XML (Extensible Markup Language) comments, 557
 document tree structure, 497–501
 editor as a JavaScript tool, 4–5
 elements summary, 233
 elements within <form> elements, 191–192
 Files, 239–241
 origination of, 490–492
 page elements in Netscape, 490
 page examples, 499–501
 page styling, 459–470
 splitting, 319–321
 structure of a DHTML (Dynamic HTML) toolbar, 528–529
 Web standard, 492
 Web standards summary, 548
 XHTML versus, 495

HTML Forms

- `alert()` method, 214
- Array object, 213
- Button object, 194–195
- check boxes and radio buttons, 207–214
- Checkbox object, 208–214
- CHECKED attribute, 208
- creating a Submit/Reset button, 197
- creating a text box, 197
- example, 198–203
- exercise questions, 235
- hidden text box, 204
- onmousedown/onmouseup event, 196–197
- Option object, 215–221
- options[] array, 216
- overview, 187–188
- password text box, 203–204
- Radio object, 208–214
- `<select>/</select>` tags, 215
- select elements, 215–227
- summary, 233
- Text Elements, 197–207
- textarea element, 204–207

HTTP (HyperText Transfer Protocol)

- defined, 3
- origination of, 490–492
- HttpRequest **class**
 - exercise questions, 655, 716–726
 - exercise solutions, 716–726
- HttpRequest **constructor, using the, 629–631**
- Hungarian notation, defined, 25**
- HyperText Transfer Protocol (HTTP)**
 - defined, 3
 - origination of, 490–492

I

i, attribute, 299, 330

icon

- Step Into, 274
- Step Out, 375
- Step Over, 376
- styling, 532

ID attribute, frames, 240–241

IE. See Internet Explorer

if statement

- example, 67–70
- multiple conditions inside an, 73–76
- overview, 66–67
- summary, 100, 101

iframes

- creating a form with, 644–651
- hiding, 650
- Safari and, 645
- server response, 645

image, inserting in a page, 671–672

Image Rollover, with Property Event Handlers, 455–458

`image_EventHandler()`, **function, 445–446, 449–458**

images[]

- Array, 167–169
- summary, 185

`` tag

- exercise questions, 186, 671
- exercise solutions, 671–672

img object, using the, 672

increment (++)

- operators, 37–39
- for statement, 85

index property, Option object, 215–216

index value, defined, 49

`indexOf()` **method, overview, 117–118, 291, 423**

infinite loop, defined, 90

initialization, for statement, 85

`<input>`, **tag, 197–203**

`<input/>`, **elements, 481–483**

`insertBefore(newNode, referenceNode)`
method, Node object, 516–518

inserting

- advertisements into a web page, 690–691
- an image in a page, 671–672
- JavaScript into a web page, 6–7

installing

- an ActiveX control, 606–607
- Microsoft Script Debugger, 368–370

plug-ins in Firefox, 598–601
 Venkman, 388–389
 a Web server, 635–636

integers, defined, 22

Internet Explorer

adding an ActiveX control to the page in,
 602–605
 adding new options with, 220–221
 blocking pop-ups in, 257
 clearing the temporary Internet file folder in,
 408–410
 cookie security and, 429–434
 disabling JavaScript in, 6
 displaying errors in, 31–32
 Exception object, 395–397
 JScript and, 3
 navigator.plugins array, 598
 retrieving an XML (Extensible Markup Language)
 file in, 577–579
 validating XML (Extensible Markup Language)
 documents in, 564
 viewing cookies in, 408–413
 Web standards and, 491–492
 World Wide Web Consortium (W3C) and, 491

Internet Explorer 5

creating an XMLHttpRequest object in, 622–624
 enabling the script debugger in, 370
 event model, 526–528
 using ActiveX in, 622–624

Internet Explorer 6

cookie security in, 429–434
 creating an XMLHttpRequest object in, 622–624
 DOM (Document Object Model) properties and,
 524
 DOM (Document Object Model) Standard and, 497
 error message dialog box, 33–34
 opening code examples in, 6
 Service Pack 1b and ActiveX controls changes,
 617
 using ActiveX in, 622–624

Internet Explorer 7

cookie security in, 429–434

creating an XMLHttpRequest object in, 624
 DOM (Document Object Model) properties and,
 524
 opening code examples in, 6
 status bar text and, 359

Internet Explorer events

srcElement property, 442–446
 type property, 439–442

**Internet file folder, clearing the temporary,
 408–410**

Internet Options dialog box, cookies and the, 409

Internet protocol (IP) addresses, defined, 3

**interpreted language, compiled language
 versus, 2**

invoking, functions, 93

IP (Internet protocol) addresses, defined, 3

is equal to, error, 365

isNaN(), function, 48–49, 663

J

Java language, JavaScript versus, 2

JavaScript. See also LiveScript

advantages of using, 3–4
 classes, 146–156
 disabling in Firefox, 5–6
 disabling in Internet Explorer, 6
 inserting into a web page, 6–7
 Java language versus, 2
 matching VBScript and, 309–310
 objects, 106–107
 overview, 1–2
 uses of, 4
 the Web and, 2–3

JavaScript Console

viewing from Firefox, 30
 viewing the, 362

JavaScript objects, using, 107–110

**JavaScript Web Applications, tools for creating,
 4–6**

join() method, using the, 133–134

JScript, Internet Explorer and, 3

L

language, computer, 1

`lastChild` property, `Node` object, 510–512

`lastIndexOf()` method, overview, 117–118, 291

leaf node, defined, 499

left, window feature, 262–264

left operand, defined, 65

length property

Array objects and the, 107, 233

`elements[]` property, 190

overview, 112

String object, 291

summary, 233

using the, 130–131

< (less than)

operator symbol, 65

XML (Extensible Markup Language) character data, 555–556

<= (less than or equal to), operator symbol, 65

Level 0, DOM (Document Object Model) Standard, 496

Level 1, DOM (Document Object Model) Standard, 496

Level 2, DOM (Document Object Model) Standard, 496

Level 3, DOM (Document Object Model) Standard, 497

`<`, entity reference, 555–556

lifespan, defined, 416

lifetime

defined, 97

summary, 102

links

adding to a pop-up window, 681–686

creating on a Web page, 701–703

`links[]` Array

summary, 185

using the, 169, 668

`linkSomePage_onclick()`, function, 172–173

literal value, defined, 27

LiveScript. See also JavaScript

Netscape Navigator 2 and, 2–3

loading

advertisements using cookies, 699–701
an XML (Extensible Markup Language) file, 579–581, 711–713

location object

overview, 163–164

summary, 185

window feature, 262–264

logic, errors versus syntax errors, 693–696

logical operators

AND (&&), 71

NOT (!), 72

OR (||), 71–72

overview, 70

summary, 101

loop

`do...while`, 90–91

exercise questions, 102–103, 659–663

exercise solutions, 659–663

`for`, 84–87, 578, 661–662, 668

`for...in`, 87–88

infinite, 90

`while`, 88–90, 662–663

looping

`break` statement and, 91

`continue` statement and, 91–92

defined, 84

`for/while` statements, 84–88

summary, 101–102

M

m, attribute, 299

Mac browsers, `url()` notation, 464

machine code, defined, 2

`match()` method

String object, 296, 318–319

summary, 329

matching, JavaScript and VBScript, 309–310

Math object

`abs()` method, 122

`ceil()` method, 123

example, 124–126

exercise questions, 663–667

exercise solutions, 663–667

- floor() method, 123
- overview, 122, 156
- pow() method, 127–129
- random() method, 126
- round() method, 123
- menubar, **window feature**, 262–264
- metaKey **property**, mouseEvent **object**, 523–526
- method**
 - troubleshooting a, 366
 - versus properties errors, 366
- method (ActiveX and Plug-Ins)**
 - DoPlay(), 715
 - setSource(), 613–614, 715
- method (Ajax and Remote Scripting)**
 - createXmlHttpRequest(), 629, 631–632, 718–726
 - document.getElementsById(), 643–644
 - open(), 626, 719–726
 - remote scripting, 631–632
 - send(), 626–627, 629, 631–632, 718–726
- method (Browser Programming)**
 - back(), 163
 - forward(), 163
 - go(), 163
 - replace(), 164
- method (Common Mistakes, Debugging, and Error Handling)**, getDay(), 694
- method (Cookies)**
 - indexOf(), 423
 - toGMTString(), 416
- method (Date, Time, and Timers)**
 - clearInterval(), 352–353, 360
 - clearTimeout(), 349–352
 - getDay(), 692–693
 - getTimeZoneOffset(), 334–339, 360
 - for getting UTC date and time, 339–348, 340–348
 - getUTCDate(), 340–348
 - getUTCDay(), 340–348
 - getUTCFullYear(), 340–348
 - getUTCHours(), 340–348
 - getUTCMilliseconds(), 340–348
 - getUTCMinutes(), 340–348
 - getUTCMonth(), 340–348
 - getUTCSeconds(), 340–348
 - setHours(), 334–339
 - setInterval(), 352–353, 360, 691
 - setTimeout(), 352
 - setUTCDate(), 339–348
 - setUTCFullYear(), 339–348
 - setUTCHours(), 334–339
 - setUTCMilliseconds(), 339–348
 - setUTCMinutes(), 339–348
 - setUTCMonth(), 339–348
 - setUTCSeconds(), 339–348
 - toLocaleDateString(), 334–339
 - toLocaleString(), 334–339
 - toLocaleTimeString(), 334–339
 - toTimeString(), 334–339
 - toUTCString(), 334–339
 - window_onload(), 353
- method (Decisions, Loops, and Functions)**, document.write(), 660–661
- method (Dynamic HTML)**
 - document.getElementsById(), 706
 - getElementsById(), 470, 486–488
- method (Dynamic HTML in Modern Browsers)**
 - appendChild(newNode), 516–518
 - cloneNode(cloneChildren), 516–518
 - createAttribute(attributeName), 516–518
 - createElement(elementName), 516–518
 - createTextNode(text), 516–518
 - Document object, 516–518
 - Element object, 518–521
 - getAttribute(attributeName), 519–521
 - getDay, 138
 - getElementsById(), 503–504
 - getElementsByTagName(), 505–506
 - hasChildNode(), 516–518
 - insertBefore(newNode, referenceNode), 516–518
 - Node object, 516–518
 - removeAttribute(attributeName), 519–521
 - removeChild(childNode), 516–518
 - replaceChild(newChild, oldChild), 516–518
 - setAttribute(attributeName, value), 519–521

method (HTML Forms)

method (HTML Forms)

add(), 220–221
alert(), 214
blur(), 193–194
focus(), 193–194, 233
onblur(), 193–194
remove(), 220–221
reset(), 190, 234
submit(), 190, 234

method (Object-Based Language)

abs(), 122
ceil(), 123
charAt(), 112–116
charCodeAt(), 112–116
concat(), 131–132
document.write(), 665
floor(), 123
fromCharCode(), 117
getDate(), 138
getFullYear(), 138
getHours(), 142–145
getMilliseconds(), 142–145
getMinutes(), 142–145
getMonth(), 138, 664
getSeconds(), 142–145
indexOf(), 117–118
join(), 133–134
lastIndexOf(), 117–118
pow(), 127–129, 157
random(), 126
reverse(), 135–137
round(), 123
setHours(), 145–146
setMilliseconds(), 145–146
setMinutes(), 145–146
setSeconds(), 145–146
slice(), 132–133
sort(), 107, 134–135, 665
substr(), 120–121
substring(), 120–121
toDateString(), 138
toFixed(), 129–130
toLowerCase(), 121–122

toTimeString(), 142–145

toUpperCase(), 121–122

method (String Manipulation)

blur(), 233–234
charCodeAt(), 291
lastIndexOf(), 291
match(), 296, 318–319, 329
replace(), 295–296, 314–318, 329
search(), 296, 318, 329
split(), 292–295, 312–314, 329
substr(), 291
substring(), 291
test(), 302–304
toLowerCase(), 291
toUpperCase(), 291

method (Windows and Frames)

moveBy(), 270, 288
moveTo(), 269–270, 288
open(), 257–261, 288
resizeBy(), 270, 288
resizeTo(), 269–270, 288
substr(), 245–247
window.close(), 288
window.open(), 288

method (XML), getElementByTagName(), 590–592

Microsoft

MSXML library, 577–579
web site, 5

Microsoft Script Debugger

breakpoints, 380–381
Call Stack Window, 383–386
Command Window, 381–383
downloading the, 367–368
exercise questions, 405, 693
exercise solutions, 693
installing the, 368–370
opening a page in, 370–375
overview, 367
Running Documents Window, 386–388
stepping through code, 375–379
summary, 404
using the, 370–388

versions, 370
 viewing pages, 387
Microsoft Visual Studio, script debuggers and, 367
Microsoft Word, regular expressions, 297
MIME (Multipurpose Internet Mail Extensions), type, 597–601
 mimeTypeypes[] array, navigator object, 600–601
mistakes, common, 361–366
modal, defined, 11
modular, defined, 237
 mousedown, mouse event, 523–526
 mousemove, mouse event, 523–526
 mouseout, mouse event, 523–526
 mouseover, mouse event, 523–526
 mouseup, mouse event, 523–526
 moveBox(), function, 482–483
 moveBy() method
 summary, 288
 window object, 270
 moveDiv(), function, 705–707
 moveTo() method
 summary, 288
 window object, 269–270
moving
 content, 478–483
 windows, 269–270
multi-condition if statements, example, 73–76
multi-dimensional array, defined, 54
multi-frame application, defined, 18
multi-line flag, attribute for, 299
multiple frames, example, 238–241
Multipurpose Internet Mail Extensions (MIME), type, 597–601
 myButton_onclick(), function, 195
 myButton_onmousedown(), function, 196–197
 myButton_onmouseup(), function, 196–197

N

\n, escape sequence, 23
 {n}, repetition character, 304–305

{n, }, repetition character, 304–305
name, of cookie strings, 415–416
 name attribute, summary, 233
 name property
 <form> elements, 193
 summary, 233
 NamedNodeMap object, base DOM object, 501
names, storing and sorting, 664–665
namespace, defined, 495
 Nan, value, 48
navigating
 the document, 509–515
 a tree structure, 708–710
 navigator object
 browser checking with the, 179–184
 summary, 185
 using the, 164
 navigator.plugins array, in Internet Explorer, 598
 navigator.userAgent, property, 181–184
 , space symbol, 23
negation, defined, 72
nested
 elements in XML, 553
 try...catch statements, 401–403
Netscape
 the blur event and, 203
 HTML page elements and, 490
 troubleshooting the blur event, 203
 Web standards and, 491–492
 World Wide Web Consortium (W3C) and, 491
new line, escape sequence, 23
 nextSibling property, Node object, 510–512, 515
 {n,m}, repetition character, 304–305
NN, escape sequence, 23
“No Plug-in or ActiveX Control” Redirection Script, testing your, 614
node
 child, 499
 defined, 498
 leaf, 499
 parent, 499

Node object

Node **object**

- appendChild(newNode) method, 516–518
 - base DOM object, 501
 - firstChild property, 510–512
 - hasChildNodes() method, 516–518
 - insertBefore(newNode, referenceNode) method, 516–518
 - lastChild property, 510–512
 - methods of the, 516–518
 - nextSibling property, 510–512
 - nodeName property, 510–512
 - nodeType, 510–512
 - nodeValue, 510–512
 - ownerDocument property, 510–512
 - parentNode property, 510–512
 - previousSibling property, 510–512
 - properties of the, 509–515
 - removeChild(childNode) method, 516–518
 - replaceChild(newChild, oldChild) method, 516–518
- NodeList **object, base DOM object, 501**
- nodeName **property, Node object, 510–512**
- nodeType **property, Node object, 510–512**
- nodeValue **property, Node object, 510–512**
- <noembed/>, **element, 608–614**
- non-Internet Explorer browsers**
- target property for, 448–450
 - type property for, 447–448
- non-Internet Explorer events, 446–450**
- <noscript> **tag, using the, 178**
- NOT (!), logical operator, 72**
- notation**
- rgb(), 463–464
 - url(), 464
- Notation **object, high-level DOM object, 502**
- Number(), **function, 69**
- Number Characters, regular expressions, 300–304**
- Number **object**
- overview, 129
 - toFixed() method, 129–130

numbers

- binary, 2
- floating-point, 22
- mixing strings and, 44–45

numerical

- calculations and data, 35–39
- calculations example, 36–37
- data type, 22

O

<object> **tag, using the, 608–614**

object

- ActiveXObject(), 577–579
- Array, 107–109, 130–137, 156, 213, 233
- Attr, 502
- base DOM, 501
- Button, 194–195
- CDATASection, 502
- Checkbox, 208–214
- Comment, 502
- creating an, 107–109, 156
- Date, 107, 137, 138, 156, 157, 334–339
- defined, 105–106
- document, 165–169, 507, 512–513, 516–518, 668
- DocumentType, 502
- Element, 502, 508–509, 518–521
- Entity, 502
- Entityreference, 502
- event, 522–526, 528
- Exception, 395–397
- Form, 188–190, 233
- global, 161
- high-level DOM, 501–502
- history, 163, 185
- img, 672
- instance defined, 147
- JavaScript, 106–107
- location, 163–164, 185
- Math, 122, 123, 124–126, 127–129, 156, 663–667

- mouseEvent, 522–526
- NamedNodeMap, 501
- navigator, 164, 179–184, 185
- Node, 509–515, 516–518
- NodeList, 501
- Notation, 502
- Number, 129–130
- Option, 215–221, 677–678
- Plugin, 598–601
- primitives and, 110–111
- ProcessingInstruction, 502
- prototype, 632
- Radio, 208–214
- RegExp, 330
- screen, 164, 185
- String, 111–122, 156, 291, 295–296, 314–318
- style, 506–507
- summary, 156
- Text, 198, 502
- using JavaScript, 107–110
- window, 161–163, 249, 269–270
- window.event, 457–458
- XMLHttpRequest, 622–629
- object-based programming**
 - objects, 105–106
 - objects in JavaScript, 106–107
 - overview, 105
 - primitives and objects, 110–111
 - summary, 156
 - using JavaScript objects, 107–110
- object methods, calling and, 110**
- object properties, using, 109–110**
- offsetLeft, **property, 484**
- offsetTop, **property, 484**
- oHttpRequest_readyStateChange(), **function, 628–629**
- onblur **event handler**
 - summary, 233–234
 - Text object, 198
- onblur **method, <form> elements, 193–194**
- onchange **event handler, Text object, 198**
- onclick
 - attribute, 170–172
 - event handler, 673–674
- one-shot timers, defined, 347**
- onfocus **event handler**
 - <form> elements, 193–194
 - summary, 233–234
 - Text object, 198
- onkeydown **event handler, Text object, 198**
- onkeypress **event handler, Text object, 198**
- onkeyup **event handler, Text object, 198**
- onload **event, exercise questions, 186**
- onload **event handler, page creation and the, 667–671**
- onmousedown **event, HTML Forms, 196–197**
- onmouseout, **event handler, 715**
- onmouseout **event, using the, 672**
- onmouseover, **event handler, 715**
- onmouseover **event, using the, 672**
- onmouseup **event, HTML Forms, 196–197**
- onreadystatechange, **event handler, 627–629**
- onselect **event handler, Text object, 198**
- onsubmit, **event handler, 723–726**
- open() **method**
 - summary, 288
 - using the, 626, 719–726
 - window object, 257–261
- opening**
 - code examples in Internet Explorer 6 and 7, 6
 - a new browser window, 257–261
 - new windows, 257–264
 - a page in Microsoft Script Debugger, 370–375
 - tags versus closing tags, 558
 - tags in XML (Extensible Markup Language), 552–553
- Opera**
 - creating an XMLHttpRequest object in, 624
 - DHTML (Dynamic HTML) toolbar, 540–544
 - DOM (Document Object Model) Standards and, 497
 - retrieving an XML (Extensible Markup Language) file in, 580

operating systems, plug-in and ActiveX control

differences between, 615

operations, basic string, 42–43

operators

AND (&&), 71

bitwise, 70

comparison, 64

comparison versus assignment, 66

decrement (—), 37–39

defined, 35

increment (++), 37–39

logical, 70–72, 101

NOT (!), 72

OR (||), 71–72

precedence, 39–42

precedence example, 40–42

(`ophone`)?, **regular expression, 324–329**

Option object

HTML Forms, 215–221

properties of the, 215–216

using the, 677–678

options

adding with Internet Explorer, 220–221

adding new, 216–220

`options[]`, **array, 220–221**

`options[]` **array, HTML Forms, 216**

Options dialog box, cookies and the, 413–414

OR (||), operator, 71–72

`ownerDocument` **property, Node object, 510–512**

P

`<p/>`, **elements, 468–470**

P3PEdit, policy creation software, 430

page

creating a, 667–671

creating a visitor count, 697–699

inserting an image in a, 671–672

viewing a, 387

page-level scope, defined, 96

`<param/>`, **elements, 605–614**

`paragraph_eventHandler()`, **function, 441–442**

parameter

defined, 11, 92

`indexOf()` **method, 117–118**

`lastIndexOf()` **method, 117–118**

passing, 93–94

parent node, defined, 499

`parent` **property, summary, 288**

`parentNode` **property, Node object, 510–512**

`parseFloat()`, **function, 46–48, 658–659**

`parseInt()`, **function, 46–48**

parsing

defined, 8

a web page example, 8–12

password text box, creating a, 203–204

path, cookie strings and, 416–417

pattern is case-sensitive, attribute for, 299

. (period), character class, 300–304

[...] (periods in brackets), character class, 300–304

Perl

regular expressions, 297

usability of, 4

PHP, downloading, 636

pixels, positioning and, 467–470

`play()`, **function, 715**

+ (plus sign), repetition character, 304–305

plug-ins

ActiveX control versus, 601–602

adding to the page in Firefox, 596–598

browser differences with, 614–615

checking for/installing in Firefox, 598–601

defined, 595

exercise questions, 618, 713–715

exercise solutions, 713–715

overview, 595–596

problems with, 614–617

RealPlayer, 595–596, 713–715

scripting differences, 615

summary, 617–618

using, 607–617

version differences, 616

Plugin object, using the, 598–601

`pluginspage`, **attribute, 598–601**

policy creation software, P3PEdit, 430

pop-up window

- adding a button/link to a, 681–686
- blocking, 257
- exercise questions, 289, 681
- exercise solutions, 681–686

positioning

- characters in regular expressions, 305–308
- content, 478–483
- of content, 467–470
- overview, 467–470

(pound sign), CSS (Cascading Style Sheets) and the, 471**pow() method**

- example, 127–129
- exercise questions, 157, 665
- exercise solutions, 665–666
- overview, 127

precedence

- defined, 39
- operator, 65

preventing, errors, 393–394

`previousSibling` **property**, `Node` **object**, 510–512, 515

primitives, objects and, 110–111**processed, defined, 1**

`ProcessingInstruction` **object**, high-level

DOM object, 502**prompt() function**

- exercise questions, 62, 658–659
- exercise solutions, 658–659
- using the, 41, 69

properties

- browser, 176–178
- class, 147
- of the `Document` object, 507
- of the `Element` object, 508–509
- event handlers as, 172–175, 455–458
- of the `Node` object, 509–515
- setting, 12
- styling, 462–470
- troubleshooting, 366
- versus methods errors, 366

property

- `altKey`, 522–526
- `appName`, 179–184
- `async`, 579–580, 717–726
- `background-color`, 463–464
- `bubbles`, 522–526
- `buttons`, 522–526
- `cancelable`, 522–526
- `clientX`, 522–526
- `clientY`, 523–526
- `color`, 460–461
- `colorDepth`, 164
- `cookie`, 422–428
- `ctrlKey`, 523–526
- `currentTarget`, 522–526
- `documentElement`, 507, 512–513
- `document.lastmodified`, 427–428
- `elements[]`, 190
- `elements []` array, 191–194
- `eventPhase`, 522–526
- `firstzchild`, 510–512
- `font-family`, 460–461, 464–465
- `font-size`, 460–461, 464–465
- `font-style`, 465, 474–475
- `font-weight`, 465
- `form`, 193, 233
- `fromElements`, 528
- `height`, 164
- `href`, 164, 186, 667–668
- `index`, 215–216
- `lastChild`, 510–512
- `length`, 107, 112, 130–131, 190, 233, 291
- `name`, 193, 233
- `navigator.userAgent`, 181–184
- `nextSibling`, 510–512, 515
- `nodeName`, 510–512
- `nodeType`, 510–512
- `nodeValue`, 510–512
- `offsetLeft`, 484
- `offsetTop`, 484
- `ownerDocument`, 510–512
- `parent`, 288

property (continued)

parentNode, 510–512
previousSibling, 510–512, 515
readyState, 579–580
relatedTarget, 523–526
request, 629, 630
responseXML, 629
screenX, 523–526
screenY, 523–526
selectedIndex, 677–678
shiftKey, 523–526
srcElement, 442–446, 528
status, 626–627
style, 471–472
tagName, 508–509
target, 448–450, 522–526
text, 215–216
text-decoration, 465, 475
toElement, 528
top, 250–257, 288
type, 233, 447–448
userAgent, 179–184
value, 193, 215–216, 234
visibility, 466
window.event.type, 442
prototype, **object, 632**
Punctuation Characters, regular expressions, 300–304

Q

? (question mark), repetition character, 304–305
questions, storing using arrays, 58–61

QuizPage.htm
creating for the trivia quiz, 283
trivia quiz, 274–275

“ (quote marks)

escape sequence, 23
strings and, 22

R

`\r`, **escape sequence, 23**
radCPUSpeed_onclick(), **function, 212**

radio buttons

creating, 207–214, 229–233
summary, 234
Radio **object, HTML Forms, 208–214**
random() **method, overview, 126**
readyState **property, using the, 579–580**
RealPlayer plug-in, using to create a page, 595–596, 713–715
recalcDateDiff(), **function, 226**
reference, defined, 108
RegExp() **constructor, RegExp object, 321–322**
RegExp() **object**
exercise questions, 330, 686–689
exercise solutions, 686–689
overview, 291, 297
RegExp() constructor, 321–322
summary, 330
test() method, 302–304
using the, 298–311
regExpIs_valid(), **function, 302–304**
regular expressions
defined, 291
exercise questions, 330–331, 687–689
exercise solutions, 687–689
grouping, 308–311
Number Characters, 300–304
(ophone)?, 324–329
overview, 297
position characters, 305–308
repetition characters, 304–305
Sax, 324–329
simple, 297–300
summary, 329–330
Text, Numbers, and Punctuation Characters, 300–304
trivia quiz and, 322–329
relatedTarget **property, mouseEvent object, 523–526**
relative positioning, defined, 467
remote scripting
class creation, 629–634
defined, 619
exercise questions, 716–726

- exercise solutions, 716–726
 - HttpRequest constructor, 629–631
 - methods, 631–632
 - uses of, 619–622
 - remove() **method**, options[] **array**, 220–221
 - removeAttribute(attributeName) **method**, Element **object**, 519–521
 - removeChild(childNode) **method**, Node **object**, 516–518
 - removeItem(), **function**, 683
 - removing, elements from a document**, 516–518
 - repetition characters, regular expressions**, 304–305
 - replace() **method**
 - location object, 164
 - String object, 295–296, 314–318
 - summary, 329
 - using the, 670–671
 - replaceChild(newChild, oldChild) **method**, Node **object**, 516–518
 - replaceQuote(), **function**, 317–318
 - request **property**, XMLHttpRequest **object**, 629
 - request_readystatechange(), **function**, 630–631, 720–726
 - reserved words, defined**, 25
 - Reset button, creating a**, 197
 - reset() **method**
 - Form object, 190
 - summary, 234
 - resetQuiz(), **function**, 355–359
 - resizable, **window feature**, 262–264
 - resizeBy() **method**
 - summary, 288
 - window object, 270
 - resizeTo() **method**
 - summary, 288
 - window object, 269–270
 - resizing, windows**, 269–270
 - responseXML, **property**, 629
 - returnPagesVisited(), **function**, 245–247
 - reverse() **method, using the**, 135–137
 - rgb() **notation, Cascading Style Sheets (CSS)**, 463–464
 - right operand, defined**, 65
 - root element, defined**, 554
 - root node, defined**, 498
 - round() **method**, 123
 - rounding**
 - functions example, 124–126
 - methods summary, 123–124
 - ROWS
 - attribute, 204–207
 - summary, 234
 - rows **attribute, frames**, 240–241
 - rules, style**, 567
 - running code, defined**, 1
 - Running Documents Window, Microsoft Script Debugger**, 386–388
- ## S
- \S, **character class**, 300–304
 - Safari**
 - creating an XMLHttpRequest object in, 624
 - DOM (Document Object Model) Standard and, 497
 - iframes and, 645
 - Safari 2, XML (Extensible Markup Language) support and**, 577
 - Same-Origin Policy, Ajax**, 652
 - Sax, **regular expression**, 324–329
 - scope**
 - defined, 96
 - summary, 102
 - <script/>, **elements**, 703
 - screen color depth, defined**, 165
 - screen **object**
 - summary, 185
 - using the, 164
 - screenX **property, MouseEvent object**, 523–526
 - screenY **property, MouseEvent object**, 523–526
 - </script>, **close tag**, 8

script block, defined, 6

Script Debugger. See **Microsoft Script Debugger**

script language, browser defaults, 7

`<script>` **tag**

attributes of, 7

overview, 6–7

using the, 6–12

scripting

frames, 251–257

between windows, 264–269

scrollbars, **window feature, 263–264**

`search()` **method**

String object, 296, 318

summary, 329

`secondCall()`, **function, 385**

secure, cookie strings and, 418

Secure Sockets Layer (SSL), cookies and, 418

security, frames and windows, 270–271

`<select>/</select>` **tags**

HTML Forms, 215

summary, 235

select elements

adding new options, 216–220

events, 221–227

HTML Forms, 215–227

`<select>/</select>` tags, 215

`selectedIndex` **property, using the, 677–678**

selector, defined, 459

; **(semicolon)**

requirements for the, 660

using the, 8

`send()` **method**

exercise questions, 718–726

exercise solutions, 719–726

remote scripting, 629

XMLHttpRequest object, 626–627, 631–632

Sequence, Unicode/Common escape, 23

server, client versus, 651

server response, iframe, 645

server-side processing, defined, 187

session

cookies defined, 428

defined, 416

`setAttribute(attributeName, value)`

method, Element object, 519–521

`setCookie()`, **function, 420–422, 431–434**

`setFrameAndPageControls()`, **function, 255–257**

`setHours()` **method**

Date object, 334–339

using the, 145–146

`setInterval()` **method**

summary, 360

using the, 352–353, 691

`setMilliseconds()` **method, using the, 145–146**

`setMinutes()` **method, using the, 145–146**

`setSeconds()` **method, using the, 145–146**

`SetSource()`, **method, 613–614**

`setSource()` **method, using the, 715**

`setTimeout()`, **method, 352**

`setTimeout()` **method**

summary, 360

using the, 348–352

setting

attributes, 518–521

changing a time, 340

the UTC date/time, 339–348

`setUTCDate()` **method, using the, 339–348**

`setUTCFullYear()`, **using the, 339–348**

`setUTCHours()`

Date object, 334–339

method, 339–348

`setUTCMilliseconds()` **method, using the, 339–348**

`setUTCMinutes()` **method, using the, 339–348**

`setUTCMonth()` **method, using the, 339–348**

`setUTCSeconds()` **method, using the, 339–348**

SGML (Standard Generalized Markup Language), origination of, 490–492

`shiftKey` **property, mouseEvent object, 523–526**

Show Cookies button, Firefox 2.0, 415

`showBasket()`, **function, 684**

`showBoxOne()`, **function, 703**

`showBoxTwo()`, **function, 703**

showing, elements, 465–466**slice() method**

- parameters of the, 132
- using the, 132–133

software, P3PEdit policy creation, 430**sort() method**

- Array objects and the, 107
- sorting an array with the, 664
- using the, 134–135

sorting

- an array, 665
- names, 664–665

split() method

- String object, 292–295, 312–314
- summary, 329

splitAndReverseText(), function, 294–295**splitting**

- HTML, 319–321
- the string, 292–295

src, attribute, 151**srcElement property**

- event object, 528
- events and the, 442–446

SSL (Secure Sockets Layer), cookies and, 418**Standard Generalized Markup Language (SGML),
origination of, 490–492****startTimer(), function, 705–707****statement**

- break, 80, 91, 102
- case, 80
- continue, 91–92, 102
- defined, 8
- else, 77–78
- else if, 78
- for, 84–88
- if, 66–70, 73–76, 100, 101
- nested try...catch, 401–403
- switch, 79–84, 100, 101
- try...catch, 394–397, 401–403
- while, 84–88

static, defined, 12**status**

- property, 626–627
- window feature, 263–264

Step Into, icon, 374, 375**Step Out, icon, 375****Step Over, icon, 376****storing**

- data in memory, 24–25
- DHTML (Dynamic HTML) toolbar button information, 532–533
- exercise questions, 157
- names, 664–665
- the results of a comparison, 66

string manipulation

- exercise questions, 686–689
- exercise solutions, 686–689

string methods, additional, 292–296**String objects**

- charAt()/charCodeAt() methods, 112–116, 291
- fromCharCode() method, 117
- indexOf()/lastIndexOf() methods, 117–120, 291
- length property, 112, 291
- match() method, 296, 318–319
- overview, 111–112
- replace() method, 295–296, 314–318
- search() method, 296, 318
- split() method, 292–295, 312–314
- substr()/substring() methods, 120–121, 291
- toLowerCase()/toUpperCase() methods, 121–122, 291

strings

- basic operations, 42–43
- changing the case of, 121–122
- comparing, 78–79
- concatenating example, 42–43
- converting arrays into single, 133–134
- converting character codes to, 117
- copying, 120–121
- copying part of, 120–121
- defined, 22
- finding strings inside other, 117–118
- mixing numbers and, 44–45
- non-convertible, 48–49
- “ (quote marks) and, 22

strings (continued)

- regular expressions, 297–311
- searching and replacing in, 295–296
- selecting a single character from, 112–113
- splitting, 292–295
- sUrl, 630–631
- <style> **tag, adding style with the, 459–461**
- style, **attribute, 461**
- style**
 - declarations defined, 459
 - rules, 567
 - sheets and XML (Extensible Markup Language), 566–568
- style **object, using the, 506–507**
- style **property, using the, 471–472**
- styling**
 - backgrounds and color, 463–464
 - buttons, 529–532
 - colors and background, 462
 - the DHTML toolbar, 529–532
 - fonts and text, 464–465
 - foreground color, 462–463
 - icons, 532
 - position, 466–470
 - showing and hiding elements, 465–466
- submit, **event, 482**
- Submit button, creating a, 197**
- submit() **method**
 - Form object, 190
 - summary, 234
- substr() **method, using the, 120–121, 245–247, 291**
- substring, defined, 117**
- substring() **method, overview, 120–121, 291**
- sUrl, **string, 630–631**
- switch statement**
 - elements of the, 79–80
 - example, 80–84
 - summary, 100, 101
- switchDirection, **global variable, 486–488**
- switchImage(), **function, 352**
- syntax, errors versus logic errors, 693–696**

T

- \t, **escape sequence, 23**
- tab, escape sequence, 23**
- tag**
 - </script>, 8
 - <A>, 170–175
 -
, 665
 - closing in XML (Extensible Markup Language), 552–553
 - <form>/</form>, 188–189, 233, 723–726
 - <frame>/<frameset>, 238–257
 - , 186, 671–672
 - <input>, 197–203
 - <noscript>, 178
 - <object>/<object/>, 608–614
 - <script>, 6–12
 - <select>/</select>, 215, 235
 - <style>, 459–461
 - <textarea>, 204–207
 - <textarea>/</textarea>, 204–207, 234
- tagName, **property, 508–509**
- target, **attribute, 188–189**
- target **property, event object, 522–526**
- <tbody/>, **elements, 591–592**
- temperature conversion, calculating, 657–658**
- temperature converter, example, 44–45**
- template**
 - rule defined, 570
 - rule in XML (Extensible Markup Language), 570
- Temporary Internet Files and History Settings dialog box, 410**
- test(), **method, 302–304**
- test condition, for statement, 85**
- test expression, switch statement, 80**
- testing**
 - browsers order, 625
 - your “No Plug-in or ActiveX Control” Redirection Script, 614
- <textarea>/</textarea> **tag**
 - summary, 234
 - textarea element, 204–207

text

- comment, 556
- data type, 22–23
- editor JavaScript tool, 4

text box

- creating a, 197
- creating a hidden, 204
- creating a password, 203–204

Text Characters, regular expressions, 300–304

text-decoration, **property, 465, 475**

Text Elements

- creating a text box, 197–198
- select() method, 197–198
- summary, 234
- Text object, 197–198

text/fonts, styling, 464–465**Text object**

- event handlers, 198
- high-level DOM object, 502

text **property, Option object, 215–216**

textarea **element, HTML Forms, 204–207**

<thead/>, **elements, 590–592**

thirdCall(), **function, 385**

throw **statement, error handling, 397–401**

time

- exercise questions, 360
- overview, 333
- settings, 340
- world, 334

timers

- clearing, 349–352
- exercise questions, 689–693
- exercise solutions, 689–693
- one-shot, 347–352
- overview, 333
- regular interval, 352–353
- summary, 360
- Web Page, 347–352

times tables, calculating, 660–663

timestamp **property, event object, 522–526**

toDateString() **method**

- Date object, 334–339
- overview, 138

toElement **property, event object, 528**

toFixed() **method, overview, 129–130**

toGMTString(), **method, 416**

toLocaleDateString() **method, Date object, 334–339**

toLocaleString() **method, Date object, 334–339**

toLocaleTimeString() **method, Date object, 334–339**

toLowerCase() **method, overview, 121–122, 291**

toolbar

- creating a, 538–540
- DHTML, 528–540
- toolbar, **window feature, 263–264**

top

- property summary, 288
- property window object, 250–257
- window feature, 263–264

toTimeString() **method**

- Date object, 334–339
- using the, 142–145

toUpperCase() **method, overview, 121–122, 291**

toUTCString() **method, Date object, 334–339**

<tr/>, **elements, 590–592**

tree structure

- creating/navigating a, 708–710
- overview, 498–499

Trivia Quiz

- building the functionality of the, 18
- creating the answer radio buttons, 229–233
- creating AskQuestion.htm for the, 283–284
- creating the form, 228–229
- creating fraGlobalFunctions for the, 280–283
- creating fraMenubar, 279
- creating the fraQuizPage, 274–278
- creating fraTopFrame, 278–279
- creating a function for the, 97–100
- creating GlobalFunctions.htm for the, 284–287
- creating QuizPage.htm for the, 283
- creating a timer-based, 354–359

Trivia Quiz (continued)

Trivia Quiz (continued)

creating the top window frameset page, 273–274
design and programming overview of the, 16–18
frames, 272–273
overview, 13–16
regular expressions and the, 322–329
storing questions using arrays, 58–61

Trivia Quiz Files

AskQuestion.htm, 283–284
GlobalFunctions.htm, 284–287
QuizPage.htm, 283

troubleshooting

ActiveX, 652–653
ActiveX controls, 614–617
case sensitivity, 363–364
closing braces, 364
closing parentheses, 365–366
concatenation, 364–365
Equals/Is Equal To, 365
exercise questions, 102
Firefox, Netscape, and the blur event, 203
methods, 366
properties, 366
variables, 362–363

truth table

AND operator and a, 70
NOT operator and a, 72
OR operator and a, 72

try...catch statements

error handling and, 394–397
nested, 401–403

txtAge_onblur(), function, 201–203

txtName_onchange(), function, 201–203

type, attribute, 7

type property

event object, 522–526
events and the, 439–442
for non-Internet Explorer browsers, 447–448
summary, 233

U

undefined, variables, 362–363

unescape(), function, 419–420

Unicode, escape sequence, 23

updateOrderDetails(), function, 677–678,
679–680

updateTime(), function, 345–347, 353

updateTimeLeft(), function, 356–359

updateTimeZone(), function, 344–347

url() notation, CSS (Cascading Style Sheets),
464

usability, concerns, 653–654

user base, defined, 13

user interaction, DHTML toolbar, 534–537

user interface

creating a, 672–678
exercise questions, 235, 672–678
exercise solutions, 672–678
userAgent property, navigator object and
the, 179–184

UTC (Coordinated Universal Time)

defined, 334
setting the date, 339–348
setting the time, 339–348
summary, 360
time summary, 360

V

valid

defined, 559
documents defined, 559

validating, XML (Extensible Markup Language) documents, 564

value, attribute storage versus arrays, 677

value

of cookie strings, 415–416
of cookies, 422–428
hexadecimal, 463
Nan, 48

value property

<form> elements, 193
Option object, 215–216
summary, 234

variables

advantages of, 24–25
assigning values to, 26

declaring, 25
 defined, 24
 example on assigning variables the values of
 other, 28–29
 example on declaring, 26–27
 exercise questions, 62
 scope summary, 102
 storing data in memory, 24–25
 troubleshooting, 362–363
 undefined, 362–363
 valid/invalid names of, 25

VBScript

matching JavaScript and, 309–310
 regular expressions, 297
 usability of, 4

Venkman

downloading, 388
 Firefox Debugging with, 388–393
 installing, 388–389
 summary, 404

| (vertical bar), alternation character, 309–310**viewing**

cookies in Firefox, 413–415
 cookies in Internet Explorer, 408–413
 the JavaScript console, 362
 pages, 387

visibility, **property, 466**

W

\W, **character class, 300–304**

W3C (World Wide Web Consortium)

goals of the, 491
 guidelines for browsers, 160
 origination of, 491–492
 Web site, 492

Web, JavaScript and the, 2–3**web browsers**

back button, 653
 checking for different, 179–184
 compatibility of, 12–13
 differences between, 13
 differences with plug-ins and ActiveX controls,
 614–615

displaying errors in, 30–34
 example for changing background color in the,
 7–8

JavaScript tool, 5
 script language defaults, 7
 summary, 186
 testing order, 625
 version checking, 175–184
 version compatibility, 12
 W3C guidelines and, 160

web page

adding a border to a, 710
 creating a frames-based, 678–680
 creating links on a, 701–703
 creating with the RealPlayer plug-in/ActiveX con-
 trol, 713–715
 example of parsing a, 8–12
 inserting advertisements into a, 690–691
 inserting JavaScript into a, 6–7
 timers, 347–352

web page events. See also events

event handlers as attributes, 170–172
 event handlers as properties, 172–175
 overview, 169–170

web server

defined, 3
 installing a, 635–636

web space, free, 417**Web standards**

DOM (Document Object Model), 496–497
 ECMAScript, 492–493
 HTML, 492
 overview, 490–492
 summary, 548
 XHTML, 494–495
 XML, 493–494

well-formed documents, defined, 552**while loop**

overview, 88–90
 summary, 101
 using the, 662–663

while statement, radio buttons, 232**width, window feature, 263–264****width property, screen object, 164**

window

- adding features to a, 262–264
 - adding HTML to a new, 261–262
 - Call Stack, 383–386
 - Command, 381–383
 - exercise questions, 289
 - moving and resizing a, 269–270
 - opening a new, 257–264
 - overview, 237, 257
 - Running Documents, 386–388
 - scripting between, 264–269
 - security, 270–271
- window **object**
- frames and the, 238–257
 - overview, 161–162
 - parent/child, 239
 - summary, 185
 - top property, 250–257
 - using the, 162–163
- windowarrayonload(), **function, 189–190**
- window.close() **method, summary, 288**
- window.event, **object, 457–458**
- window.event.type, **property, 442**
- windowonload(), **function, 225–226, 241**
- window_onload(), **function, 352, 353, 715**
- window_onunload(), **function, 269**
- window.open() **method, summary, 288**
- Windows 2000, script debuggers and, 367**
- World Time**
- Converter example, 341–347, 353
 - overview, 334
- World Wide Web Consortium (W3C)**
- goals of the, 491
 - origination of, 491–492
 - Web site, 492
- wrap, **attribute, 204–207**
- writeMonthOptions(), **function, 225**
- writeOptions(), **function, 224**
- writeTimesTable(), **function, 373–374, 390–393, 661–662**

X

XHTML

- HTML versus, 495
- Web standard, 494–495, 548

XML (Extensible Markup Language)

- building an application, 582–592
 - CDATA, 556
 - character data, 555–556
 - closing tags, 552–553
 - comments, 556–557
 - comments versus HTML comments, 557
 - creating a document, 557–566
 - displaying, 566–576
 - document structure, 554
 - exercise questions, 593, 710–713
 - exercise solutions, 710–713
 - formatting, 566–576
 - manipulating with JavaScript, 576–592
 - nesting elements, 553
 - opening tags, 552–553
 - overview, 551–552
 - schema defined, 494
 - style sheets and, 566–568
 - summary, 592–593
 - template rule, 570
 - Web standard, 493–494
 - Web standards summary, 548
 - as a well-formed language, 552
 - XML (Extensible Markup Language) elements, 554–555
 - XSLT transformations, 568–576
- XML (Extensible Markup Language) documents**
- adding data to, 561–566
 - creating, 711
 - creating a DTD file, 559–561
 - Document Type Definition (DTD), 559
 - exercise questions, 593
 - linking to XSL style sheets, 574–576
- XML (Extensible Markup Language) file**
- cross-browser retrieval of an, 581–582
 - exercise questions, 593
 - loading an, 579–581, 711–713
 - retrieving in Firefox and Opera, 580
 - retrieving in Internet Explorer, 577–579
- xmlDoc_readyStateChange(), **function, 580**
- XMLHttpRequest **object**
- creating in Internet Explorer 5/6, 622–624
 - creating in Internet Explorer 7, Firefox, Opera, and Safari, 624

XMLHttpRequest object

- asynchronous requests, 627–629
- creating a form with, 634–644
- cross-browser issues, 622–625
- Internet Explorer 5 and 6 and the, 622–625
- other browsers, 624–625
- overview, 622
- summary, 654–655
- using the, 626–627

\xNN, escape sequence, 23**XSL style sheets, linking XML (Extensible Markup Language) documents to, 574–576****XSLT**

- defined, 568–570
- overview, 568–570
- template example, 571–574
- template rules, 570–574
- transformations and formatting XML (Extensible Markup Language) documents, 568–576

Z**Zip, files, 552**

