

# Index

## SYMBOLS

-- (decrement operators), 28–30

# (pound sign), 537

\$ ( )

in jQuery, 531

in MooTools, 535, 575–576

in Prototype, 533

understanding jQuery, 550

\$ (dollar sign), 320

\$\$ ( )

in MooTools, 576

in Prototype, 559

\$.get ( ), 552–553

&& (AND), 58

\* (asterisk)

in numerical calculations, 26–27

precedence, 30–34

repetition characters, 319

[ . . . ] in regular expressions, 315

[ ^ . . . ] in regular expressions, 315

^ (carat), 320

{ } (curly braces)

in if statements, 54–55

incorrect number of, 90

{ n , }, 319

{ n , m }, 319

{ n }, 319

|| (OR), 58

+ (plus sign)

missing in concatenation, 92–93

in numerical calculations, 26–27

precedence, 30–34

repetition characters, 319

in strings, 34–35

++ (increment operators), 28–30

< (less than), 53–54

<= (less than or equal to), 53–54

= (equals)

vs. == (is equal to), 91

as assignment operator, 22

in numerical calculations, 27

!= (unequal), 53–54

== (is equal to)

vs. = (equals), 91

defined, 53–54

preventing infinite loops, 77

> (greater than), 53–54

>= (greater than or equal to), 53–54

\ (backslash)

reusing groups of characters, 325–326

in strings, 19

when using RegExp ( ), 336

“ (double quotation marks) in strings, 18–19

/ (forward slash)

in numerical calculations, 26–28

precedence, 30–34

in regular expressions, 307

setting cookie path, 377

- (hyphen) in CSS properties, 423

## - (minus sign)

---

### - (minus sign)

in numerical calculations, 26–27  
precedence, 30–34

! (NOT), **58**

. (period) in regular expressions, **315**

? (question mark), **319**

; (semicolons) in statements, **9**

' (single quotation marks) in strings, **18–19**

( ) (parentheses)

grouping regular expressions, 323–325  
incorrect number of, 90–91  
using functions as properties, 91–92

## A

abs(), **161**

absolute values, **161**

action attribute, **220**

**ActiveX controls**

checking for and embedding, 474–479  
exercise question, 489  
overview, 469–470  
security issues, 516  
summary, 488–489  
using, 480–488

**ActiveX objects**

defined, 454  
XMLHttpRequest, 494–496

ActiveXObject, **683**

**adding**

addEventListener(), 535  
new options to selection boxes, 248–252  
options to selection boxes, 248–252  
selection options, 252–253

**address validation, 343–344**

**advertisements**

adding timers to, 360–363  
animating, 432–436

**Ajax**

creating simple module, 500–505  
dealing with delays, 523–524  
defined, 490–494  
degrading gracefully, 524  
exercise questions, 525  
iframe smart form, 518–523

jQuery support, 550–558

MooTools support, 584–590

Prototype support, 567–575

summary, 525

things to watch out for, 515–518

validating forms fields with, 505–508

XMLHttpRequest, 494–500

XMLHttpRequest smart form, 508–515

ajaxError(), **551–552**

Ajax.Request(), **567–575**

alert()

defined, 12–13

in window, 191

**alphabetical comparison, 66**

altkey property

of Event, 439

of event, 444

AND

decision making, 58–59

in if statements, 61–64

**animated advertisements, 432–436**

**appearances**

changing in DOM, 422–428

jQuery, 539–541

MooTools, 576–577

Prototype, 560

appendChild(), **419**

**appending elements with jQuery, 541–543**

**applications**

tools for creating, 5–7

using JavaScript for, 4

appName, **209–210**

**arithmetic operators, 674–675**

Array

concat(), 150–151

creating, 136–137

JavaScript reference, 684–686

join(), 152–153

length property, 150

new array methods, 156–160

object, 150

reverse(), 154–156

slice(), 151–152

sort(), 153–154

**arrays**

- creating objects, 182–187
- looping, 74–75
- matching, 333–335
- multi-dimensional, 45–49
- searching for, 306
- working with, 40–45

**assignment operators**

- vs. comparison operators, 53
- defined, 22
- JavaScript reference, 672–673

**asterisk (\*)**

- in numerical calculations, 26–27
- precedence, 30–34
- repetition characters, 319

**asynch property**

- of `Ajax.Request()`, 568
- defined, 455–456
- jQuery's Ajax, 553
- in MooTools, 584

**asynchronous requests, 499–500****attr(), 531****Attr**

- defined, 403
- properties, 705

**attributes**

- `<form/>`, 220–222
- `<frameset/>`, 267
- ActiveX control, 475–478
- changing appearances with `class`, 426–428
- getting and setting with `Element`, 408, 410–412
- handling events with HTML, 200–202
- plug-in, 470–471
- radio button, 239–240
- `RegExp`, 308
- in text boxes, 230
- `textarea`, 236

**availability of plug-ins/ActiveX controls, 481–482****available value, 506****B****\B, 320****\b, 320****back(), 193****back button and Ajax, 516–517****backslash (\)**

- reusing groups of characters, 325–326
- in strings, 19
- when using `RegExp()`, 336

**base DOM objects, 402–403****Berners-Lee, Tim, 392****bgColor property, 195–197****bind(), 543–545****bitwise operators, 675****bitwise shift operators, 676****block statements, 8–9, 679–680****blur(), 225–226****blur events, 235****<body/> element, 400****BOM (Browser Object Model)**

- defined, 189–190
- `document`, 195–197
- vs. DOM, 399
- history, 193
- images collection, 198–199
- links collection, 199
- location, 194
- navigator, 194–195
- overview, 190–191
- screen, 195
- window, 191–193

**Boolean, 686****Boolean data, 20****boundary characters, 320–323****break statements**

- looping, 78–79
- in `switch`, 67–68, 70–71

**breakpoints**

- defined, 103
- setting in Dragonfly, 130
- setting in Firebug, 106–107
- setting in IE, 119–120
- setting with Web Inspector, 125–126

**Browser Object Model (BOM).** *see* **BOM (Browser Object Model)**

## **browser programming**

- browser checking with `navigator`, 209–215
- connecting code to events, 200–206
- determining user's browser, 206–209
- `document`, 195–197
- events, 199–200
- exercise questions, 217
- `history`, 193
- images collection, 198–199
- links collection, 199
- location, 194
- `navigator`, 194–195
- objects, 190–191
- overview, 189–190
- `screen`, 195
- summary, 215–217
- window, 191–193

## **browsers**

- Ajax support, 493–494
- compatibility and, 15
- cookie limitations, 387–388
- cookies in Firefox, 373–374
- cookies in IE, 368–373
- debugging. *see* *debugging*
- DOM standard compliance, 398
- exception objects, 95
- JavaScript and, 2–3
- methods for adding new options, 252–253
- needed to create applications, 5
- opening new windows in, 284–292
- potential problems with plug-ins, 485
- preventing errors in, 93
- reference, 669
- retrieving XML files cross-browser, 458–459
- using `XMLHttpRequest`, 494–496
- Web standards, 392–394
- writing cross-browser DHTML, 447–453

`bubbles` **property**, 439

## **button elements**

- combining with text boxes, 230–235
- defined, 224
- in HTML forms, 226–229
- properties and methods, 224–226

`button` **property**

- of `Event`, 439
- of `event`, 444

## **buttons**

- radio buttons, 239–247
- timer, 360

# C

## **call stacks**

- defined, 103
- Firebug, 114–116

## **calling functions, 80–81**

`cancelable` **property**, 439

`cancelBubble` **property**, 444

`carat (^)`, 320

**Cascading Style Sheets (CSS).** *see* **CSS (Cascading Style Sheets)**

## **case**

- checking character, 146–149
- conversion, 144–145

## **case sensitivity**

- common mistakes, 89
- in ordering arrays, 153–154
- in `RegExp`, 308
- `RegExp()` flags, 336
- of special characters in regular expressions, 315

`case` **statements**, 67–71

`CDATASection`, 403

`ceil()`, 161–162

## **chaining methods**

- in jQuery, 531
- in MooTools, 535
- in Prototype, 533

`change` **event**, 254–260

## **character classes, 315**

## **characters**

- case sensitivity, 89
- checking for passphrases for alphanumeric, 316–318
- comparing strings, 66
- comparison operator, 53

- converting to strings, 149
- escaping, 378–379
- logical operator, 58
- repetition, 319–320
- reusing groups of, 325–326
- selecting from strings, 145–149
- special characters in regular expressions, 315–323
- String object, 140
- in strings, 18–19
- in variables, 21
- charAt() , **145–149**
- charCodeAt() , **145–149**
- check boxes in HTML forms, 239–247**
- checked **attribute, 239–240**
- checking code, 93**
- child nodes, 400**
- child window objects**
  - accessing parent variables and functions, 268–273
  - defined, 265–266
- Chrome**
  - Ajax support, 493
  - browser checking with navigator, 210–215
  - DOM standard compliance, 398
- class **attribute, 426–428**
- classes**
  - altering styles with MooTools, 576–577
  - manipulating styles with Prototype, 560
  - vs. reference types, 178
  - using CSS to change styles, 539–540
- classid **attribute, 475–478**
- clauses, 102–103**
- clearing timers**
  - continually firing timers, 363
  - one-shot timers, 360
- click **events**
  - counting button, 226–227
  - defined, 440
  - handling in DOM, 442
- clients, 3**
- client-side processing, 219**
- client.x **property, 579**
- clientX **property**
  - of Event, 440
  - of event, 444
- client.y **property, 579**
- clientY **property**
  - of Event, 440
  - of event, 444
- cloneNode() , **419**
- close()
  - windows, 285
  - windows security, 298
- closing tags, 9**
- closures, 501**
- code access, 273–276**
- code stepping**
  - defined, 103
  - with Dragonfly, 130–131
  - Firebug, 108–112
  - IE, 121–122
  - Web Inspector, 127
- codebase **attribute, 478–479**
- coding between frames, 268–273**
- collections**
  - defined, 195
  - elements, 222
  - forms, 220–222
  - images, 198–199
  - links, 199
  - options, 247–248
- colors**
  - changing with style property, 423
  - setting document, 195–197
- cols **attribute, 236**
- Comment, **403**
- comments, 12**
- comparison operators**
  - comparing strings, 66
  - defined, 52–54
  - JavaScript reference, 673–674
- compatibility, cross-browser. see cross-browser compatibility**
- compiled code, 2**
- compressed JavaScript frameworks, 529**
- computer language, 1**

# concat()

---

`concat()`, **150–151**

## **concatenation**

defined, 34–35  
missing + in, 92–93

## **conditional statements**

decision making, 51–52  
reference, 680

## **conditions**

evaluating with consoles, 113  
`if` statement, 54–55  
multiple in `if` statements, 60–64  
`for` statement, 72

## **consoles**

in debugging, 103  
Dragonfly, 129  
Firebug, 112–114  
IE, 122  
Web Inspector, 127

## **constructors**

ActiveX object, 454  
`Ajax.Request()`, 567–575  
creating, 178–179  
creating arrays, 183  
creating elements in Prototype, 561  
`HttpRequest`, 501–502  
object, 136  
reference type, 178  
`RegExp`, 335–339  
regular expression, 307

## **content**

animation, 433–436  
positioning and moving, 429–432  
separation of style and, 453–454

## **continually firing timers**

defined, 358  
setting, 363–364

`continue`, **78–79**

## **controls**

accessing between frames with `top`, 277–283  
ActiveX. *see ActiveX controls*  
code access between frames, 276  
HTML events, 726

**conversion, data type, 37–40**

**converting case, 144–145**

## **converting time**

setting and getting UTC date and time, 353–358  
setting timers to fire at regular intervals,  
363–364

## **cookie pairs, 387**

## **cookies**

creating, 377–381  
exercise questions, 389  
fresh-baked, 368–374  
getting value, 381–386  
limitations, 386–388  
overview, 367  
security and IE6+, 388–389  
strings, 374–377  
summary, 389

## **Coordinated Universal Time (UTC)**

defined, 348–351  
setting and getting date and time, 351–358

## **copying**

part of array, 151–152  
parts of strings, 143–144

`createElement()`, **541–542**

`createXmlHttpRequest()`, **500, 502–503**

## **cross-browser compatibility**

defined, 15  
determining user's browser, 206  
with DHTML, 447–453  
retrieving XML files, 458–459  
of XML, 395  
`XMLHttpRequest`, 494–497

## **ctrlKey property**

of `Event`, 440  
of `event`, 444

## **CSS (Cascading Style Sheets)**

altering styles with MooTools, 576–577  
changing appearances with, 422–428  
changing styles with jQuery, 539–541  
manipulating styles with Prototype, 560  
rewriting DHTML toolbar with jQuery, 545–550  
selecting elements with jQuery, 537–538  
selecting elements with Prototype, 559

**current date, 170–172**

**current time, 174–176**

**D****\D, 315****\d, 315****data**

- arrays, 40–45
- basic string operations, 34–35
- conversion, 37–40
- exercise questions, 50
- mixing numbers and strings, 35–37
- multi-dimensional arrays, 45–49
- numerical calculations, 26–30
- operator precedence, 30–34
- primitive, 134
- summary, 49
- transfer with XML, 454
- validation, 93

**data types**

- creating string objects, 140
- ECMAScript, 396
- objects and, 138–139
- overview, 17–20
- returning node, 413

**databases, 20****Date**

- calculations and, 173
- getting `Date` values, 169–172
- getting time values, 173–176
- JavaScript reference, 686–689
- objects, 168–169
- setting and getting UTC date and time, 351–358
- setting `Date` values, 172
- setting time values, 176–177
- summary, 364
- using `select` element to calculate differences, 254–260
- world time, 348–351

**daylight savings time**

- world time and, 348, 350
- World Time Converter, 353–358

**debuggers, 5****debugging**

- call stack window, 114–116
- exercise questions, 131–132

- Firebug code stepping, 108–112
- Firebug console, 112–114
- Firebug watches, 107–108
- in Firefox with Firebug, 104–106
- in IE, 116–119
- IE code stepping, 121–122
- IE console, 122
- IE watches, 121
- overview, 103–104
- in Safari, 123–127
- setting breakpoints in Firebug, 106–107
- setting breakpoints in IE, 119–120
- summary, 131
- using Dragonfly, 127–131

**decisions**

- `AND`, 58–59
- comparing strings, 66
- comparison operators, 52–54
- `else` and `else if`, 64–66
- exercise questions, 86
- `if` statement, 54–58
- logical operators, 58
- multiple conditions inside `if` statements, 60–64
- `NOT`, 60
- `OR`, 59
- overview, 51–52
- summary, 84–85
- `switch` statement, 67–71

**declarations, 680****decrement loop variables, 72****decrement operators, 28–30****default statement, 67–68, 70****default windows, 290–291****defaultStatus, 192–193****degrading gracefully, 524****delays in Ajax, 523–524****development tools, 5****DHTML (Dynamic HTML)**

- animated advertisements, 432–436
- defined, 391
- exercise questions, 468
- moving and positioning content with, 429–432
- summary, 467
- writing cross-browser with, 447–453

## **DHTML (Dynamic HTML) toolbar**

- rewriting with jQuery, 545–550
- rewriting with MooTools, 579–584
- rewriting with Prototype, 563–567

`displayEvent()`, **238**

`Document`

- defined, 403
- properties and methods, 404–408, 705–706

**Document Object Model (DOM)**. *see* **DOM (Document Object Model)**

`document`

- cookie property, 367
- defined, 195–197
- forms in, 220
- navigating with DOM, 415–419
- as window property, 192–193

`DocumentFragment`

- defined, 403
- reference, 706–707

## **documents**

- defined, 9
- document representation as tree structure, 399–402

`DocumentType`, **403**

**Dojo**, **528**

**dollar sign (\$)**, **320**

## **DOM (Document Object Model)**

- animated advertisement example, 432–436
- vs. BOM, 399
- changing appearances, 422–428
- core objects, 402–404
- defined, 391–392
- `Document` object properties and methods, 404–408
- `Element` object properties and methods, 408–412
- event handling, 437–443
- event handling in IE, 443–447
- event model and objects, 723–725
- extensions in Prototype, 533
- high-level core objects, 704–709
- HTML document as tree structure, 399–402
- HTML elements, 711–723
- `HTMLCollection`, 709

`HTMLDocument`, 710–711

- low-level core objects, 701–704
- miscellaneous events, 726–727
- `Node` object methods, 419–422
- `Node` object properties, 412–419
- positioning and moving content, 429–432
- standard, 397–398
- summary, 467

`domain`, **377**

**domain name servers**, **3**

## **domain names**

- cookies and, 370
- validation, 342–344

`domready`, **535**

**double quotation marks (“) in strings**, **18–19**

`doubleAndAlert()`, **159–160**

`do...while`, **77–78**

**downloading frameworks**, **529–530**

**Dragonfly**, **127–131**

## **drop-down lists**

- adding and removing options from, 248–252
- creating, 247–248
- methods for adding new options, 252–253
- select element events, 254–260

**Drosera**, **123**

**DTD**, **403**

**Dynamic HTML (DHTML)**. *see* **DHTML (Dynamic HTML)**

**dynamic properties**, **13**

# **E**

`each()`, **559**, **576**

**ECMA (European Computer Manufacturers Association)**, **3**

## **ECMAScript**

- defined, 3
- Web standards, 396–397

## **editors**, **5**

`Element`

- creating, inserting and removing in Prototype, 561–562
- properties and methods, 408–412

**elements**

- accessing in DOM, 422
- accessing with `for...in`, 74–75
- animating, 433–436
- button, 226–229
- changing appearances, 422–428
- changing style with jQuery, 539–541
- counting with `length`, 150
- creating, appending and removing with jQuery, 541–543
- creating, inserting and removing in MooTools, 577–578
- creating, inserting and removing in Prototype, 561
- creating in DOM, 407
- creating with DOM methods, 420–422
- defined, 41
- finding with MooTools, 575–576
- form properties and methods, 224–226
- frame, 264–268
- in HTML forms, 223–224
- HTML objects, 711–723
- manipulating styles with Prototype, 560
- in multi-dimensional arrays, 45–49
- operations with `forEach()` and `map()`, 159–160
- ordering with `reverse()`, 154–156
- ordering with `sort()`, 153–154
- positioning and moving, 429–432
- properties and methods, 224–226, 707–708
- retrieving in Prototype, 558–559
- select events, 254–260
- selecting with jQuery, 537–538
- text, 229–236
- `textarea`, 236–238
- in tree structure, 400

**elements collection, 222**

**else, 64–66**

**else if, 64–66**

**e-mail address validation, 342**

**Email field**

- defined, 505–506
- validation, 508–515

**embedding**

- ActiveX controls, 474–479
- plug-ins, 470–474

**enabling**

- IE debugger, 116–119
- Web Inspector, 123–125

**Entity, 403**

**EntityReference, 403**

**equals (=)**

- vs. `==` (is equal to), 91
- as assignment operator, 22
- in numerical calculations, 27

**error handling**

- `finally` clauses, 102–103
- nested `try...catch`, 100–102
- prevention, 93–94
- throwing errors, 96–100
- `try...catch`, 94–96

**error messages**

- display example, 459–467
- jQuery's Ajax, 551–552

**errors**

- common, 87–93
- vs. exceptions, 94
- potential problems with plug-ins, 485–488

**escape(), 378**

**escape characters, 19**

**escape sequences, 19**

**European Computer Manufacturers Association (ECMA), 3**

**Event**

- accessing, 437–438
- properties, 439–443
- properties and methods, 724
- properties in MooTools, 578–579

**event**

- accessing, 443–444
- properties, 444–447

**event handling**

- button element mouse events, 227–228
- counting button clicks, 226–227
- defined, 200
- DOM, 437–443
- with HTML attributes, 200–202
- in IE, 443–447
- jQuery's Ajax, 551–552
- with object properties, 203–206
- radio button, 240, 244–245
- select element, 254
- in text boxes, 230

**Event.observe(), 533**

eventPhase **property**, 439

### events

- connecting code to, 200–206
- defined, 199–200
- DOM, 726–727
- DOM model and objects, 723–725
- jQuery, 543–545
- MooTools, 578–579
- Prototype, 562–563
- select element, 254–260

EventTarget, **723**

every(), **157**

**exception handling statements**, 681

**exception objects**, 95

**exceptions**, 94

**execution control statements**, 681

**exiting loops**, 78–79

expires, **375**

**expressions**, 26. *see also regular expressions*

**Ext JS**, 528

**external JavaScript files**, 7–8

**external web sites**, 263–264

## F

**features in windows**, 289–292

**files, XML**

- retrieving cross-browser, 458–459
- retrieving in Firefox and Opera, 457–458
- retrieving in IE, 454–457

filter(), **157–158**

finally **clauses**, 102–103

### Firebug

- call stack window, 114–116
- code stepping, 108–112
- console, 112–114
- setting breakpoints with, 106–107
- watches, 107–108

### Firefox

- Ajax support, 493
- browser checking with navigator, 210–215
- checking for and embedding plug-ins, 470–474
- compatibility, 15
- creating JavaScript applications, 5–7

debugging with Firebug. *see Firebug*

DOM standard compliance, 398

new array methods, 156–160

problems with blur events and, 235

retrieving XML files in, 457–458

viewing cookies in, 373–374

firstChild **property**, 412

fix(), **166–167**

**flags in RegExp**, 308

**floating-point numbers**

data type conversion, 38–39

defined, 18

floor(), **161–162**

focus(), **225–226**

for **statement**, 71–74

forEach(), **159–160**

for...in, **74–75**

form **property**, 224–225

### forms

HTML. *see HTML (HyperText Markup Language)*  
*forms*

jQuery's Ajax validation, 553–558

MooTools' Ajax validation, 585–590

Prototype's Ajax validation, 569–575

validating fields with Ajax, 505–508

XMLHttpRequest smart form, 508–515

forms **collection**, 220–222

forward(), **193**

**forward slash (/)**. *see / (forward slash)*

**fractional numbers**

numerical data, 18

in for statements, 74

<frame/>, **264–268**

### frames

code access between, 273–276

coding between, 268–273

exercise questions, 299–300

overview, 262–263

scripting iframes, 283–284

security, 298

summary, 298–299

support for hidden, 494

top property, 277–283

window and, 263–268

`<frameset/>`, **264–268**

### framesets

code access between frames, 273–276  
defining, 267  
using as modules, 269–273

**frameworks, JavaScript.** *see* **JavaScript frameworks**

`fromCharCode()`, **149**

`fromElement` **property**, **444**

`Function`, **689–690**

**functions.** *see also* **methods**

accessing `parent` in frames, 268–273  
closure, 501  
code access between frames, 273–276  
cookie creation, 377–381  
creating, 79–83  
defined, 12–13  
exercise questions, 86  
getting cookie value, 382–383  
`jQuery`, 531  
`MooTools`, 535  
`Prototype`, 533  
scripting between windows, 292–296  
summary, 85–86  
top-level, 682–683  
understanding `jQuery`, 550  
using as properties, 91–92  
variable scope and lifetime, 83–84  
`XMLHttpRequest`, 496–497

## G

**garbage collection**, **23**

**Garrett, Jesse James**, **491**

`get()` **functions**

`Date` object's UTC date and time,  
351–358  
for date values, 169–172  
finding elements by `idValue` or `tagName`,  
404–407  
getting attributes with `Element`, 410–412  
for time values, 173–176

**global matches**

in `RegExp`, 308  
`RegExp()` flags, 336

**global objects**, **191**

**global scope**, **83**

**GMT (Greenwich Mean Time)**, **348**

`go()`, **193**

**Google Maps**, **492**

**Google Suggest**, **493**

**greater than (>)**, **53–54**

**greater than or equal to (>=)**, **53–54**

**Greenwich Mean Time (GMT)**, **348**

**group elements**, **239**

**grouping regular expressions**, **323–325**

## H

`handleData()`, **505**

**hard wrapping text**, **236**

`hasChildNodes()`, **419**

`hasClass()`, **540–541**

`<head/>` **element**, **400**

`height`

screen, 195  
setting window, 285

**Hewitt, Joe**, **104**

**hidden frames**, **494**

**hidden text boxes**, **235–236**

**hierarchies**

BOM, 191  
frame, 275  
HTML document as tree structure, 399–400  
window object, 265–266  
in XHTML, 396

**high-level DOM objects**

defined, 403–404  
reference, 704–709

`history`, **193**

**history stacks**

Ajax usability, 516–517  
defined, 193

**hosts**, **3**

`href` **property**, **199**

# html()

---

html(), **531–532**

## **HTML (HyperText Markup Language)**

- adding to new window, 289
- control events, 726
- document representation as tree structure, 399–402
- dynamic. *see* *DHTML (Dynamic HTML)*
- elements, 711–723
- event handlers with attributes, 200–202
- navigating documents with DOM, 415–419
- properties vs. JavaScript properties, 13
- splitting with `match()`, 333–335
- String methods, 699
- Web standards, 392–394
- vs. XHTML, 396

## **HTML (HyperText Markup Language) forms**

- adding and removing options from selection boxes, 248–252
- button elements, 226–229
- check boxes and radio buttons, 239–247
- element properties and methods, 224–226
- elements in, 223–224
- exercises, 262
- methods for adding new options, 252–253
- object properties and methods, 221–222
- overview, 219–221
- select element events, 254–260
- selection boxes, 247–248
- summary, 260–262
- text elements, 229–236
- textarea element, 236–238

`<html/>` **element, 400**

HTMLAnchorElement, **712**

HTMLBodyElement, **712–713**

HTMLButtonElement, **713**

HTMLCollection, **709**

HTMLDivElement, **713**

HTMLDocument, **710–711**

HTMLElement, **712**

HTMLFormElement, **714**

HTMLFrameElement, **714–715**

HTMLFrameSetElement, **715**

HTMLImageElement, **716**

HTMLInputElement, **716–717**

HTMLOptionCollection, **718**

HTMLOptionElement, **717–718**

HTMLParagraphElement, **718**

HTMLSelectElement, **718–719**

HTMLTableCellElement, **719–720**

HTMLTableElement, **720–721**

HTMLTableRowElement, **721**

HTMLTableSectionElement, **722**

HTMLTextAreaElement, **722–723**

## **HTTP (HyperText Transfer Protocol)**

- defined, 3
- standards, 392–393

HttpRequest **module, 500–505**

## **Hungarian notation, 21**

## **HyperText Transfer Protocol (HTTP)**

- defined, 3
- standards, 392–393

## **hyphen (-) in CSS properties, 423**

## **I**

### **id attribute**

- of `<frameset/>`, 267
- accessing elements with, 422
- finding elements by, 404–407

### **identifiers to avoid, 671**

### **IE (Internet Explorer)**

- Ajax support, 493
- browser checking with `navigator`, 210–215
- checking for and embedding ActiveX controls in, 474–479
- code stepping, 121–122
- compatibility, 15
- console, 122
- cookies and security, 388–389
- creating JavaScript applications, 5–7
- debugging in, 116–119
- determining user's browser, 206–209
- event handling in, 443–447
- Node in, 414
- potential problems with ActiveX controls, 487–488
- retrieving XML files in, 454–457
- scripting iframes, 283–284

security issues and operator precedence, 32–33  
 setting breakpoints in, 119–120  
 using `XMLHttpRequest`, 494–496  
 viewing cookies in, 368–373  
 watches, 121

#### **if statement**

decision making, 54–58  
 else and, 64–66  
 multiple conditions inside, 60–64  
 throwing errors, 96–100

#### **iframes**

creating back/forward-capable forms with, 517  
 scripting, 283–284

#### **images**

displaying when page loads, 204–206  
 images collection, 198–199

`img`, **198–199**

#### **incomplete forms, 233**

#### **increment loop variables, 72**

#### **increment operators, 28–30**

index property, **248**

#### **index values, 41**

`indexOf()`

finding array elements with, 156  
 searching for substrings, 140–143

#### **infinite loops, 77**

information storage with cookies. *see* **cookies**

#### **initialization**

defined, 25  
 for statement, 72, 74

`<input/>` element, **223–224**

`insertBefore()`, **419**

#### **inserting elements**

in MooTools, 577–578  
 in Prototype, 561

#### **installation**

ActiveX control, 478–479  
 Firebug, 104–106  
 framework, 529–530  
 plug-in, 472–474  
 testing framework, 530–536

#### **instances, 180–182**

#### **integers**

data type conversion, 38–39  
 numerical data, 18

**Internet Explorer (IE)**. *see* **IE (Internet Explorer)**

**Internet protocol (IP) addresses, 3**

**interpreted code, 2**

**interpreters, 2**

**intervals, setting timer, 363–364**

**invoking functions, 80–81**

**IP (Internet protocol) addresses, 3**

**is equal to (==)**

vs. = (equals), 91  
 defined, 53–54  
 preventing infinite loops, 77

`isLessThan3()`, **157–158**

#### **iterations**

of `Array`, 156–160  
 for statement, 72

## **J**

### **JavaScript**

browsers and compatibility, 15

ECMAScript and, 396–397

introduction, 1–7

objects. *see* **objects**

objects in, 134–135

simple programs, 8–14

summary, 16

web pages, 7–8

XML and, 453–454

### **JavaScript frameworks**

Ajax support in MooTools, 584–590

Ajax support in Prototype, 567–575

creating, appending and removing elements with  
`jQuery`, 541–543

creating, inserting and removing elements in  
 MooTools, 577–578

creating, inserting and removing elements in  
 Prototype, 561

exercise questions, 590

finding elements with MooTools, 575–576

getting started, 528–536

`jQuery` event model, 543–545

`jQuery` for Ajax, 550–558

`jQuery` styles, 539–541

MooTools events, 578–579

MooTools styles, 576–577

Prototype events, 562–563

### JavaScript frameworks (continued)

- Prototype styles, 560
- retrieving elements in Prototype, 558–559
- rewriting DHTML toolbar with jQuery, 545–550
- rewriting DHTML toolbar with MooTools, 579–584
- rewriting DHTML toolbar with Prototype, 563–567
- selecting, 527–528
- selecting elements with jQuery, 537–538
- summary, 590

### JavaScript reference

- ActiveXObject, 683
- arithmetic operators, 674–675
- Array, 684–686
- assignment operators, 672–673
- bitwise operators, 675
- bitwise shift operators, 676
- Boolean, 686
- browsers, 669
- comparison operators, 673–674
- Date, 686–689
- Function, 689–690
- identifiers to avoid, 671
- logical operators, 676–677
- Math, 691–692
- miscellaneous operators, 677
- Number, 692–693
- Object, 693–694
- operator precedence, 678–679
- properties and functions, 682–683
- RegExp, 694–697
- reserved words, 670
- statements, 679–682
- String, 697–699
- join()
  - defined, 152–153
  - in simple regular expressions, 314
- joining arrays with `concat()`, 150–151
- jQuery
  - Ajax support, 550–558
  - creating, appending and removing elements with, 541–543
  - defined, 528
  - event model, 543–545

- rewriting DHTML toolbar with, 545–550
- selecting elements with, 537–538
- styles, 539–541
- testing installation, 531–533

## K

### keyboard events

- miscellaneous, 726
- in text boxes, 230
- watching, 237–238

`keyCode` **property**, 445

## L

`lastChild` **property**, 412

`lastIndexOf()`

- finding array elements with, 156
- searching for substrings, 140–143
- using `substr()` with, 144

**leaf nodes**, 400

**left operands**, 52–53

**left property**

- moving and positioning content with, 429–432
- performing animation with, 433–436

**left-hand side (LHS)**

- of comparison operators, 52–53
- of logical operators, 58–60

**length property**

- of Array, 150
- of `elements` collection, 222
- of history, 193
- of String, 140

**less than (<), 53–54**

**less than or equal to (<=), 53–54**

**level 0 of DOM**, 397

**level 1 of DOM**, 398

**level 2 of DOM**, 398

**level 3 of DOM**, 398

**LHS (left-hand side)**

- of comparison operators, 52–53
- of logical operators, 58–60

**lifespan, cookie**, 375

**lifetime, variable**, 83–84

**linking to external JavaScript files, 7–8**links **collection, 199****list boxes**

- adding and removing options from, 248–252
- creating, 247–248
- methods for adding new options, 252–253
- select element events, 254–260

**listeners, 200****literal values, 23****local time**

- vs. world time, 348–351
- World Time Converter, 353–358

## location

- defined, 194
- as window property, 192–193

**logical operators**

- AND, 58–59
- decision making, 58
- JavaScript reference, 676–677
- NOT, 60
- OR, 59

**looping**

- break and continue, 78–79
- to check for plug-ins, 473–474
- defined, 680–681
- do...while, 77–78
- exercise questions, 86
- for...in, 74–75
- iterative methods of Array, 156–160
- for statement, 71–74
- summary, 85
- while, 75–77

**M****machine code, 2****manipulating strings. see string manipulation**map(), **159–160**

## match()

- defined, 306
- regular expressions, 333–335
- in simple regular expressions, 314

**matches**

- repetition characters, 319–320
- in simple regular expressions, 308–314

## Math

- JavaScript reference, 691–692
- overview, 160–167

**mathematical capabilities, 26–30**max(), **161**maxLength **attribute, 229****members, Element, 408****memory, 20–25**message **property, 95****message-of-the day displays, 459–467**metaKey **property, 440**method **attribute**

- of <form/>, 220
- of Ajax.Request(), 568
- in MooTools, 584

**methods**

- for adding new options, 252–253
- Array, 684–686
- Boolean, 686
- browser checking with navigator, 209–215
- calling object, 138
- charAt() and charCodeAt(), 145–149
- concat(), 150–151
- creating, appending and removing elements with jQuery, 541–543
- creating arrays, 183–186
- creating objects, 179–180
- css(), 539
- Date, 687–689
- Document, 404–408
- DOM event model, 723–725
- Element, 408–412
- fromCharCode(), 149
- Function, 690
- hasClass(), 540–541
- high-level DOM object, 704–709
- HTML element object, 711–723
- HTML forms element, 224–226
- HTML forms object, 221–222
- HTMLCollection, 709
- HTMLDocument, 710–711
- HttpRequest module, 502–503
- indexOf() and lastIndexOf(), 140–143
- join(), 152–153
- jQuery, 531–532

### methods (continued)

`match()`, 333–335  
`Math`, 691–692  
`Math` object, 160–167  
`MooTools`, 535, 575–579  
moving and positioning content, 429–432  
moving and resizing windows, 297  
new array, 156–160  
`Node`, 419–422, 703–704  
`Number`, 693  
`Number` object, 167  
`Object`, 694  
`observe()`, 562–563  
`open()`, 284–292  
`Prototype`, 533  
reference type, 178  
`RegExp()`, 307  
`RegExp`, 695  
`replace()`, 329–332  
`reverse()`, 154–156  
`search()`, 332  
setting and getting UTC date and time, 351–358  
`slice()`, 151–152  
`sort()`, 153–154  
`split()`, 327–328  
`String`, 698–699  
string manipulation, 302–306  
`substr()` and `substring()`, 143–144  
`toFixed()`, 167–168  
`toggleClass()`, 540  
`toLowerCase()` and `toUpperCase()`, 144–145  
`XMLHttpRequest`, 500

**Microsoft Internet Explorer. see IE (Internet Explorer)**

**Microsoft MSXML library**  
recommended versions, 454–456  
`XMLHttpRequest`, 494

**Microsoft Web standards, 393**

**MIME (Multipurpose Internet Mail Extensions) types**  
checking for plug-ins, 474  
defined, 471

`min()`, **161**

**minified frameworks, 529**

### minus sign (-)

in numerical calculations, 26–27  
precedence, 30–34

### mistakes, common, 87–93

### mobile technology, 395

### MochiKit, 528

### modal, 13

### modular code

creating simple Ajax module, 500–505  
defined, 263  
using framesets as modules, 269–273

### MooTools

Ajax support, 584–590  
creating, inserting and removing elements in, 577–578  
defined, 528  
events, 578–579  
finding elements with, 575–576  
rewriting DHTML toolbar with, 579–584  
styles, 576–577  
testing installation, 535–536

### mouse events

changing appearances in DOM, 425–426  
handling in IE, 444–447  
miscellaneous, 726  
properties of, 439–443

### `mousedown` events, 440

### `MouseEvent`, 724–725

### `mousemove` events, 440

### `mouseout` events, 440

### `mouseover` events, 440

### `mouseup` events, 440

### `moveBy()`, 297

### `moveTo()`, 297

### moving

content with DHTML, 429–432  
windows, 296–297

### Mozilla Firefox. see Firefox

### multi-dimensional arrays, 45–49

### multi-line flags, 308

### Multipurpose Internet Mail Extensions (MIME)

#### types

checking for plug-ins, 474  
defined, 471

**N**name **property**

- accessing `Form`, 220–222
- exception object, 95
- in form elements, 224
- radio button, 239
- setting cookie, 374–375
- window object, 268

## NamedNodeMap

- defined, 402
- properties and methods, 704

**names**

- changing appearance with `className`, 426–428
- cookie, 380
- function, 80–81
- returning node, 413
- variable, 21

**native JavaScript objects, 133****navigation**

- Ajax usability, 516–517
- creating back/forward-capable forms with `iframe`, 517–523
- DOM with `Node`, 412–419
- with `location`, 194
- tree structure, 399–400

## navigator

- browser checking with, 209–215
- defined, 194–195
- `plugins` property, 472–474
- as window property, 192–193

**negation, 60****nested if statements, 60–64****nested try...catch, 100–102****Netscape, 393**new **keyword, 180**nextSibling **property, 412**

## Node

- defined, 402
- methods, 419–422
- properties, 412–419
- reference, 701–704

## NodeList

- defined, 402

- properties and methods, 703–704

<noscript/> **element, 209**

## NOT

- decision making, 60
- in if statements, 61–64

Notation, **403**

## Number

- defined, 167
- JavaScript reference, 692–693
- `toFixed()`, 167–168

**numerical data**

- calculations, 26–30
- converting in text boxes, 229
- data type conversion, 37–39
- defined, 18
- mixing strings and, 35–37
- in regular expressions, 315–318

**O**Object, **693–694****objects**

- array, 150
- array length, 150
- browser. see *BOM (Browser Object Model)*
- calculations and `Date`, 173
- `charAt()` and `charCodeAt()`, 145–149
- `concat()`, 150–151
- creating and using reference type instances, 180–182
- creating array, 182–187
- creating `Date`, 168–169
- creating new types of, 177–178
- defining reference types, 178–180
- Document properties and methods, 404–408
- DOM, 402–404
- Element properties and methods, 408–412
- exception, 95
- exercise questions, 188
- `Form`, 220–222
- `fromCharCode()`, 149
- getting `Date` values, 169–172
- getting time values, 173–176
- high-level DOM, 704–709
- in HTML forms, 221–222

### objects (continued)

- `indexOf()` and `lastIndexOf()`, 140–143
- `join()`, 152–153
- jQuery, 531–532
- low-level DOM, 701–704
- Math, 160–167
- new array methods, 156–160
- Node methods, 419–422
- Node properties, 412–419
- Number, 167
- object-based programming, 133–139
- RegExp, 306–307
- `reverse()`, 154–156
- setting Date values, 172
- setting time values, 176–177
- `slice()`, 151–152
- `sort()`, 153–154
- String, 139–140
- String's length, 140
- `substr()` and `substring()`, 143–144
- summary, 187–188
- `toFixed()`, 167–168
- `toLowerCase()` and `toUpperCase()`, 144–145
- XMLHttpRequest, 494–500
- `observe()`, **562–563**
- occurrences**
  - counting substring, 142–143
  - searching for, 305
- one-shot timers**
  - defined, 358
  - in web page, 359–363
- onFailure property**
  - of `Ajax.Request()`, 568
  - in MooTools, 584
- onSuccess property**
  - of `Ajax.Request()`, 568
  - in MooTools, 584
- `open()`
  - using XMLHttpRequest, 497
  - in window, 284–292
- opener property, 292–296**
- opening**
  - Dragonfly, 127–129
  - new browser windows, 284–292

### Opera

- Ajax support, 493
- browser checking with `navigator`, 210–215
- debugging with Dragonfly, 127–131
- determining user's browser, 207–208
- DOM standard compliance, 398
- retrieving XML files in, 457–458

### operating systems and plug-ins, 486

### operators

- arithmetic, 674–675
- assignment, 22, 672–673
- bitwise, 675
- bitwise shift, 676
- comparison, 52–54, 673–674
- logical, 58, 676–677
- miscellaneous, 677
- numerical calculations, 26–30
- precedence, 30–34, 678–679
- string, 34–35

### options

- adding and removing from selection boxes, 248–252
- methods for adding new, 252–253
- select element events, 254–260
- in selection boxes, 247–248

### OR

- decision making, 59
- in if statements, 61–64

### ordering arrays

- with `reverse()`, 154–156
- with `sort()`, 153–154

`ownerDocument` **property, 413**

## P

### page-level scope, 83

`page.x` **property, 578**

`page.y` **property, 578**

### parameters

- adding features to new windows, 289–292
- adding HTML to new window, 289
- of `Ajax.Request()`, 568
- creating functions, 81–82
- defined, 13
- function, 79

- jQuery's Ajax, 552–558
  - method, 138
  - open(), 284–285
  - of rounding methods, 162
  - using functions as, 92
- parent nodes, 400**
- parent **property**
  - code access between frames, 273–276
  - coding between frames, 268–273
  - vs. top property, 277
  - window object, 268
- parent window **objects, 265–266**
- parentheses ( )**
  - grouping regular expressions, 323–325
  - incorrect number of, 90–91
  - using functions as properties, 91–92
- parentNode **property, 413**
- parseFloat(), **38–39**
- parseInt()
  - data type conversion, 38–39
  - rounding numbers, 162
- parsing, 9–13**
- passphrases, 316–318**
- Password **field**
  - defined, 505–506
  - validation, 508–515
- password text boxes, 235**
- path, **375–377**
- patterns, character, 332**
- period (.) in regular expressions, 315**
- PHP form field validation with Ajax, 508–515**
- plug-ins**
  - checking for and embedding, 470–474
  - exercise question, 489
  - overview, 469–470
  - summary, 488–489
  - using, 480–488
- plus sign (+). see + (plus sign)**
- pointers, 92**
- position characters, 320–323**
- positioning content, 429–432**
- postal code validation, 341–342**
- pound sign (#), 537**
- pow(), **165–167**
- precedence, operator**
  - comparison, 53
  - defined, 30–34
  - reference, 678–679
- prevention, error, 93–94**
- previousSibling **property, 412**
- primitive data**
  - defined, 134
  - objects and, 138–139
- privacy and cookies, 388–389**
- processing, client-side, 219**
- processing code, 1**
  - ProcessingInstruction, **403**
- programming, browsers. see browser programming**
- programming, object-based, 133–139**
- properties**
  - accessing attributes as, 220–222
  - Ajax.Request(), 568
  - Array, 684
  - asynchronous request, 499–500
  - bgColor, 195–197
  - Boolean, 686
  - browser checking with navigator, 209–215
  - changing appearances with CSS, 422–428
  - checking for browser support, 207–209
  - cookie, 367
  - creating arrays, 183
  - creating objects, 179
  - Date, 686
  - defined, 134
  - document, 9
  - of document, 195
  - Document, 404–408
  - Element, 408–412
  - Event, 439–443
  - event, 444–447
  - exception object, 95
  - Function, 689–690
  - handling events with object, 203–206
  - high-level DOM object, 704–709
  - of history, 193
  - HTML element object, 711–723
  - HTML forms element, 224–226
  - HTML forms object, 221–222

### properties (continued)

- HTMLCollection, 709
- HTMLDocument, 710
- img, 198–199
- length. *see length property*
- of location, 194
- Math, 691
- Node, 412–419, 702–704
- Number, 692
- Object, 693
- opener, 292–296
- parent in window, 268
- plug-in, 472–474
- position, 429–432
- reference type, 178
- RegExp, 694–695
- Request, 584
- of screen, 195
- String, 698
- top, 277–283
- top-level, 682–683
- using as functions, 91–92
- using object, 137–138
- window, 192

### Prototype

- Ajax support, 567–575
- creating, inserting and removing elements in, 561
- defined, 528
- events, 562–563
- retrieving elements in, 558–559
- rewriting DHTML toolbar with, 563–567
- styles, 560
- testing installation, 533–535

### punctuation in regular expressions, 315–318

## Q

### question mark (?), 319

### QuickTime

- determining availability, 481–482
- finishing up, 482–484
- plug-in example, 479–480
- potential problems, 485–488
- testing the disabling of form, 484–485

## R

### radio buttons, 239–247

- random(), 164–165
- readonly attribute, 230
- readyState property

- asynchronous requests, 499–500
- defined, 456
- installing ActiveX controls, 479

### reference types

- creating and using instances, 180–182
- creating new object types, 177–178
- creating objects, 136–137
- defining, 178–180

### RegExp(), 307

- RegExp
  - defined, 306–307
  - domain name validation, 342–344
  - e-mail address validation, 342
  - JavaScript reference, 694–697
  - postal code validation, 341–342
  - telephone number validation, 339–341
  - using constructor, 335–339

### regular expressions

- covering all eventualities, 323
- defined, 301
- grouping, 323–325
- overview, 306–307
- reusing groups of characters, 325–326
- simple, 307–314
- special characters in, 315–323
- using `match()` with, 333–335
- using `replace()` with, 329–332
- using `search()` with, 332
- using `split()` with, 327–328

### relatedTarget property

- of Event, 440
- in MooTools, 579

### remote servers, 454

### removing

- attributes with `Element`, 408, 410–412
- child nodes, 419
- elements with jQuery, 541–543
- elements with MooTools, 577–578
- elements with Prototype, 561

methods for node, 419  
 options, 253  
 options from selection boxes, 248–252

**repetition characters, 319–320**

`replace()`  
 defined, 305  
 in location, 194  
 regular expressions, 329–332  
 in simple regular expressions, 307–309

`request` **property, 500**

**requests.** *see* `XMLHttpRequest`

**required fields, 233**

**reserved words**

defined, 21  
 overview, 670–671

`reset()`, **222**

**Reset buttons, 228–229**

**resizing windows, 296–297**

`responseXML` **property, 500**

**retrieving**

cookie value, 381–383  
 elements in Prototype, 558–559  
 XML files cross-browser, 458–459  
 XML files in Firefox and Opera, 457–458  
 XML files in IE, 454–457

`return` **statements, 80**

**reusability of external files, 8**

`reverse()`, **154–156**

**RHS (right-hand side)**

of comparison operators, 52–53  
 of logical operators, 58–60

**right operands, 52–53**

**right-hand side (RHS)**

of comparison operators, 52–53  
 of logical operators, 58–60

**root nodes, 400**

**rounding numbers, 161–164**

`rows` **attribute**

of `<frameset/>`, 267  
 of `textarea`, 236

**running code, 1**

**run-time errors, 93**

## S

`\s`, **315**

`\S`, **315**

**Safari**

Ajax support, 493  
 browser checking with `navigator`, 210–215  
 debugging in, 123–127  
 DOM standard compliance, 398

**same-origin policy**

Ajax security, 515–516  
 defined, 277  
 windows security, 298

**scope, 83–84**

`screen`

defined, 195  
 setting color depth, 195–197  
 as window property, 192–193

`screenX` **property**

of Event, 440  
 of event, 445

`screenY` **property**

of Event, 440  
 of event, 445

**script blocks, 8–9**

`</script>` **tag, 9**

`<script>` **tag**

defined, 8–9  
 linking to external files, 7–8

**scripting**

differences among plug-ins, 485–486  
 iframes, 283–284  
 with `top` property, 277–283  
 between windows, 292–296  
 windows security, 298

`search()`

defined, 305–306  
 regular expressions, 332

**searching**

for date values, 169–172  
 for substrings, 140–143

`secure`, **377**

## security

- Ajax, 515–516
  - cookies and IE6+, 388–389
  - IE issues and operator precedence, 32–33
  - windows and frames, 298
- `select()`, **229**
- ### select elements
- defined, 223
  - events, 254–260
  - properties and methods, 224–226
  - in selection boxes, 247–248
- ### selection boxes
- adding and removing options from, 248–252
  - in HTML forms, 247–248
  - methods for adding new options, 252–253
  - select element events, 254–260
- ### selectors, CSS
- selecting elements in jQuery, 537–538
  - selecting elements with Prototype, 559
- `self` **property, 283**
- ### semicolons (;) in statements, 9
- `send()`
- defined, 500, 503
  - using `XMLHttpRequest`, 498
- ### separation of content and style, 453–454
- ### session cookies
- lifespan, 375
  - security, 388–389
- `set()` **functions**
- continually firing timers, 363–364
  - cookie creation, 379–381
  - `Date` object's UTC date and time, 351–358
  - for date values, 172
  - frame and page controls, 282
  - setting attributes with `Element`, 410–412
  - setting one-shot timers, 359–363
  - styles, 560
  - for time values, 176–177
- ### SGML (Standard Generalized Markup Language)
- defined, 392
  - XML and, 394
- `shiftKey` **property**
- of `Event`, 440
  - of event, 445

- ### simple regular expressions, 307–314
- ### single quotation marks (') in strings, 18–19
- ### size attribute
- in selection boxes, 247
  - in text boxes, 229
- `slice()`, **151–152**
- ### smart forms
- `iframe`, 518–523
  - `XMLHttpRequest`, 508–515
- ### soft wrapping text, 236
- `some()`, **157**
- `sort()`, **153–154**
- ### special characters
- `RegExp`, 695–697
  - in regular expressions, 315–323
- ### speed in animation, 436
- `split()`
- defined, 302–305
  - regular expressions, 327–328
- `src` **property, 198–199**
- `srcElement` **property, 445**
- ### Standard Generalized Markup Language (SGML)
- defined, 392
  - XML and, 394
- ### standards
- DOM, 397–398
  - Web, 392–397
  - `XMLHttpRequest`, 494–500
- ### statements
- defined, 9
  - JavaScript reference, 679–682
- ### static methods, 149
- ### static properties, 13
- `status` **property, 498–499**
- ### stepping, code. see code stepping
- ### storing data with cookies. see cookies
- ### storing data with variables, 20–25
- `String`
- `charAt()` and `charCodeAt()`, 145–149
  - `fromCharCode()`, 149
  - `indexOf()` and `lastIndexOf()`, 140–143
  - JavaScript reference, 697–699
  - `length` property of, 140

objects, 139–140  
 substr() and substring(), 143–144  
 toLowerCase() and toUpperCase(), 144–145

### string manipulation

additional methods, 302–306  
 covering all eventualities, 323  
 domain name validation, 342–344  
 e-mail address validation, 342  
 exercise questions, 345–346  
 grouping regular expressions, 323–325  
 match(), 333–335  
 overview, 301  
 postal code validation, 341–342  
 RegExp constructor, 335–339  
 regular expressions, 306–307  
 replace(), 329–332  
 reusing groups of characters, 325–326  
 search(), 332  
 simple regular expressions, 307–314  
 special characters in regular expressions, 315–323  
 split(), 327–328  
 summary, 345  
 telephone number validation, 339–341

### strings

basic operations, 34–35  
 comparing, 66  
 converting arrays to, 152–153  
 cookie, 374–377  
 data type conversion, 37–39  
 defined, 18–19  
 mixing numbers and, 35–37  
 not convertible, 40

### styles

changing appearances with, 423–426  
 jQuery, 539–541  
 MooTools, 576–577  
 Prototype, 560  
 separation of content and, 453–454  
 style object, 406

submit()

defined, 222  
 vs. focus() and blur(), 225

### Submit buttons, 228–229

substr(), 143–144  
 substring(), 143–144

### substrings

creating array with split(), 302–305  
 searching for, 140–143  
 searching for occurrences, 305

switch statement, 67–71

### synchronous requests, 498–499

## T

### tab strip creation, 447–453

### tables in tree structure, 401–402

tagName

in Element, 408–409  
 finding elements by, 404–407

target property

of <form/>, 220  
 of Event, 439  
 in MooTools, 579

### telephone number validation, 339–341

### templates, 177–178

test(), 314

test conditions. *see conditions*

### test expressions, 67

### testing

disabled forms, 484–485  
 frameworks, 530–536  
 methods of Array, 157  
 regular expressions, 309–314

### text

boxes, 229–235  
 changing appearances in DOM, 424–426  
 defined, 18–19  
 mixing with numbers, 35–37  
 nodes, 407  
 in regular expressions, 315–318  
 reversing order with split(), 302–305  
 searching for, 305–306

Text

defined, 403  
 methods, 709

### text elements

- creating with DOM methods, 420–422
  - defined, 223
  - in HTML forms, 229–236
  - properties and methods, 224–226
  - textarea, 236–238
- text **property**, 248
- textarea, **236–238**
- throwing errors, 96–100**
- tick box elements**
- defined, 223
  - properties and methods, 224–226
- time**
- getting values, 173–176
  - setting and getting UTC date and, 351–358
  - setting values, 176–177
  - summary, 364
  - world time, 348–351
- time zones, 348**
- timers**
- defined, 347
  - exercise questions, 365
  - in web page, 358–365
- timestamp **property**, 439
- toElement **property**, 445
- toFixed(), **167–168**
- toggleClass(), **540**
- toLowerCase(), **144–145**
- toolbar, DHTML. see DHTML (Dynamic HTML)**
- toolbar**
- tools, debugging. see debugging**
- tools needed to create applications, 5–7**
- top **property**
- in frames, 277–283
  - moving and positioning content with, 429–432
  - performing animation with, 433–436
- top-level properties and functions, 682–683**
- toUpperCase(), **144–145**
- transferring data with XML, 454**
- tree structure of HTML document**
- getting document's root element, 407–408
  - navigating with DOM, 415–419
  - overview, 399–402
- truth tables, 59–60**

- try...catch
  - error handling, 94–96
  - finally clauses, 102–103
  - nested, 100–102
  - throwing errors, 96–100
- two-dimensional arrays, 46–49**
- type **attribute**
  - of <script>, 9
  - defined, 7
  - for plug-ins, 470–471
- type **property**
  - of Event, 439
  - of event, 445
  - in form elements, 225
  - in MooTools, 579
- types, data. see data types**

## U

- unavailable **value**, 506
- uncompressed JavaScript frameworks, 529**
- undefined variables, 88–89**
- unequal (!=), 53–54**
- unescape(), **378–379**
- updates**
  - DOM, 422
  - finding changes with cookies, 384–386
  - in Prototype, 561
  - World Time Converter, 356–358
- url **property**, 584
- userAgent, **209–215**
- Username **field**
  - defined, 505–506
  - validation, 508–515
- users**
  - base, 15
  - browser checking, 206–209
  - browser checking with navigator, 209–215
  - disabling cookies, 386–387
  - input errors, 96–100
  - interacting with HTML forms. *see HTML (HyperText Markup Language) forms*
- UTC (Coordinated Universal Time)**
  - defined, 348–351
  - setting and getting date and time, 351–358

**V****validation**

- domain names, 342–344
- e-mail addresses, 342
- form field with Ajax, 505–508
- jQuery's Ajax form, 553–558
- MooTools' Ajax form, 585–590
- postal codes, 341–342
- Prototype's Ajax form, 569–575
- RegExp(), 337–339
- telephone numbers, 339–341
- XMLHttpRequest smart form, 508–515

**value property**

- in form elements, 224
- of Option, 248
- setting cookie, 374–375

**values**

- assigning to other variables, 24–25
- assigning to variables, 21–23
- getting cookie, 381–386
- readyState property, 456
- returning node, 413

**var keyword, 21****variables**

- accessing parent in frames, 268–273
- arrays and, 40–45
- assigning comparison results in, 54
- code access between frames, 273–276
- creating objects, 136–137
- IE watches, 121
- multi-dimensional arrays, 45–49
- overview, 20–25
- scope and lifetime, 83–84
- scripting between windows, 292–296
- setting in Watch tab, 107–108
- undefined, 88–89
- Web Inspector watches, 126

**Venkman, 104****Verify Password field**

- defined, 505–506
- validation, 508–515

**versions**

- browser checking with navigator, 210–215
- browsers and compatibility, 15

DOM, 397–398

JavaScript framework, 529

MSXML, 454–456

potential problems with plug-ins and ActiveX controls, 486–487

of Web standards, 394–397

XMLHttpRequest, 494–495

**viewing**

- cookies in Firefox, 373–374
- cookies in IE, 368–373

**W**

\W, 315

\w, 315

**W3C (World Wide Web Consortium)**

- cookie security, 388
- DOM. *see DOM (Document Object Model)*
- Web standards, 392–394

**watches**

- defined, 103
- Firebug, 107–108
- IE, 121
- key events, 237–238

**web and JavaScript, 2–3****Web Inspector, 123–127****web pages**

- adding ActiveX controls, 475–478
- adding frameworks to, 530
- adding plug-ins, 470–471
- displaying results in, 14
- placing script in, 7–8
- timers in, 358–365

**web servers**

- Ajax usability and response, 517–523
- defined, 3
- form field validation with Ajax, 506–507

**Web standards, 392–397****Webkit, 123**

while, 75–77

**whole numbers, 18****width property**

- screen, 195
- setting window, 285

## window

---

### window

- accessing `event` property, 443–444
- code access between frames, 273–276
- coding between frames, 268–273
- defined, 191–193
- frames and, 263–268
- scripting iframes, 283–284
- setting timers in, 359–363
- `top` property, 277–283

### windows

- events, 727
- exercise questions, 299–300
- moving and resizing, 296–297
- opening new browser, 284–292
- scripting between, 292–296
- security, 298
- summary, 298–299

### word-boundary characters, 320–323

### world time, 348–351

#### World Time Converter

- setting and getting UTC date and time, 353–358
- setting to fire at regular intervals, 363–364

### World Wide Web Consortium (W3C)

- cookie security, 388
- DOM. *see* *DOM (Document Object Model)*
- Web standards, 392–394

### `wrap` attribute, 236

### `write()`

- adding HTML to new window, 289
- setting screen color depth, 195–197

### `writeOptions()`, 257

## X

### XHTML, 395–396

#### XML (eXtensible Markup Language)

- error message display example, 459–467
- JavaScript and, 453–454
- retrieving file cross-browser, 458–459
- retrieving file in Firefox and Opera, 457–458
- retrieving file in IE, 454–457
- Web standards, 394–395

#### XMLHttpRequest

- browser checking with, 208–209
- browser's back button and, 516–517
- creating simple Ajax module, 500–505
- dealing with delays, 523–524
- defined, 494–500
- smart form, 508–515

## Y

### Yahoo! User Interface Framework (YUI), 528

## Z

### Zip files, 454