Index

Symbols

$ (dollar sign)
 beginning JavaScript variable names 13
 function in jQuery 624

0 (zero), as falsey value 292

&& (AND) operator 55, 62–63, 74

* (asterisk) operator, as multiplication arithmetic operator 15, 286

: (colon), separating property name and property value 179

, (comma), separating object properties 177, 179

{ } (curly braces)
 enclosing object properties 177, 179
 in body of function 97
 in code block 17
 matching in code 59
 using with object literals 177, 522

. (dot notation)
 accessing object properties 181, 209, 230
 using with reference variables 186
 using with this object 202

= (equal sign) operator, assigning values to variables using 11, 16, 275

== (equality) operator
 as comparison operator 16, 55, 275–285, 311, 459
 vs. === operator 289

=== (strict equality) operator
 as comparison operator 55, 280–285, 311
 vs. == operator 289

// (forward slashes), beginning comments in JavaScript 13

/ (forward slash) operator, as division arithmetic operator 15, 286

> (greater than) operator 16, 55, 459

>= (greater than or equal to) operator 16, 55

< (less than) operator 55, 459

<= (less than or equal to) operator 55

- (minus sign) operator
 as unary operator 287
 using as arithmetic operator with string and number 286–287, 312

!= (not equal to) operator 16, 55

! (NOT) operator 55

|| (OR) operator 54, 55, 62–63, 74

() (parentheses)
 in calling functions 68, 430, 439
 in parameters 97

+ (plus sign)
 as arithmetic operator 286–287, 312
 in concatenating strings 15, 133, 142, 354

-- (post-decrement operator) 146–147

++ (post-increment operator) 146–147

“ “ (quotation marks, double)
 surrounding character strings in JavaScript 13
 using around property name 179

; (semicolon), ending statements in JavaScript 11, 13

[ ] (square brackets)
 accessing properties using 209
 in arrays 127, 129, 550

!== (strict not equal to) operator 281

_ (underscore), beginning JavaScript variable names 13

A

action attributes 328

activities, about doing xxxiii

addEventListener method 630

alert function
 communicating with users using 25–26, 42, 46
 determining hits and misses in simplified Battleship game 59–60, 76

Copyright © 2014. O'Reilly. All rights reserved. May not be reproduced in any form without permission from the publisher, except fair uses permitted under U.S. or applicable copyright law.

EBSCOhost®

EBSCO Publishing : eBook Collection (EBSCOhost) - printed on 7/28/2014 3:08 PM via FULL SAIL UNIVERSITY
AN: 748459 ; Freeman, Eric, Robson, Elisabeth.; Head First JavaScript Programming
Account: s9105391

alt attribute 256

AND (&&) operator 55, 62–63, 74

anonymous functions
- about , 475–476, 482, xx–xxi
- accessing properties using 509, 518
- assigning to method in constructors 530–532, 557
- creating 477–478, 512–513
- making code tighter 479
- passing functions to functions 482, 486, 514
- readability of 480

API-specific events 413

applications. *See also* Battleship game, advanced; Battleship game, simplified
- coding JavaScript 29–35
- creating interactive 319
- JavaScript in 5
- web pages as 9

arguments, function
- about 85
- identifying 91, 105, 119, 120
- mixing up order of 97
- objects as 192
- passing 88–89, 92–94
- using pass-by-value 92–93
- vs. parameters 90

arguments object 628

array constructor object 549–551

array literal syntax 550

arrays
- about , xiii–xv
- abstracting code into functions 157–162
- accessing item in 129
- arranging code exercise 139, 168
- Auto-O-Matic app (example) 195–197
- creating
  - empty 151
  - most effective bubble solution code 164–165, 171
  - with values 128
- Cubicle Conversation on 148–149, 170
- declaring variables in 152
- empty 134, 153–154, 365, 367, 549–550
- for determining hits in advanced Battleship game 344–345
- for hits and ship locations in advanced Battleship game 338
- for loop in 140–142, 144–145, 147, 169
- indices in 129, 134, 152, 163
- initializing counter 140
- iterating over 138, 140
- length property in 130
- literals in 151–152
- number of levels deep to nest objects 348
- number of things in 134
- order of items in 134
- Phrase-o-Matic app (example) 131–133
- populating playlist items using 253, 262
- reusing code in 156
- sparse 152
- test drive 143, 147, 155
- undefined values and 268
- updating value in 129
- value types in 134
- while loop in 17–21

array sort method 457–463, 472–473

asterisk (*) operator, as multiplication arithmetic operator 15, 286

asynchronous coding
- about events 383
- alt attribute 256
- event handlers
  - about 383
  - adding using addEventListener 630
  - assigning to properties 407
  - callbacks and 250
  - creating 385–386
  - in advanced Battleship game 358–359, 361
  - kinds of 252
  - onload 249
  - timerHandler 407
  - using setInterval function 410, 425
  - using setTimeout function 407–413
- Event object in DOM 399–402, 423
- events 404
- exercise on notification of events 382, 421
- interview with browser about events 403
- reacting to events
  - about 387
  - adding images to image guessing game 393–397
  - assigning click handlers 396–397
  - assigning handler to onclick property 390–391, 393–398

EBSCOhost®

Copyright © 2014. O'Reilly. All rights reserved. May not be reproduced in any form without permission from the publisher, except fair uses permitted under U.S. or applicable copyright law.

EBSCO Publishing : eBook Collection (EBSCOhost) - printed on 7/28/2014 3:08 PM via FULL SAIL UNIVERSITY
AN: 748459 ; Freeman, Eric, Robson, Elisabeth.; Head First JavaScript Programming
Account: s9105391

Copyright © 2014. O'Reilly. All rights reserved. May not be reproduced in any form without permission from the publisher, except fair uses permitted under U.S. or applicable copyright law.

creating image guessing game 388–392, 411–414, 422, 426
exercise on 415, 427
asynchronous coding event handlers
event handlers
creating closure with 503–507
attributes
action 328
alt 256
class 255
getting with getAttribute 256
not in element 256
setting with code 254
setting with setAttribute 255
Auto-O-Matic app (example) 195–197

B

background CSS property 326
background image
cells in 322
creating HTML page 320–321
Battleship game, advanced
about , xvii–xix
cheating at 369
controller object
about 329
counting guesses and firing shot 355–356
determining end of game 356
developing parseGuess function 351–354
implementing 349–357
passing input to 360
processing player's guess 350–354
testing 357
creating HTML page
about 320
adding CSS 324–326
background 320–321
placing hits and misses 326
table 322
using hit and miss classes 327, 375
designing game 329–330, 376
event handler to fire button 359
generating random locations for ships 362–368, 380
getting players guess 358–361
keeping track of ships 337
model object
about 329, 336
determining hits 344–345
determining if ship is sunk 346
fire method in 342
implementing 341–348
interacting with view 336
looking for hits 343
notifying view of hits or misses 347
parseGuess function asking about size of board 353
representation of ships in 338–340, 378–379
representing state of ships in 337, 377
testing 348
QA in 370–371
starting location of ships 366
testing 370
toolkit for building 319
view object
about 329
implementing 329, 331–334
interacting with model 336
notifying of hits or misses 347
testing 335
Battleship game, simplified
about 44
adding hit detection code 56–57, 77
adding HTML to link code 45–46
assigning random ship locations 66–68
checking if ship sank 57
checking users guess 54
designing 45–46
displaying stats to user 58
doing quality assurance 61, 69–70
finding errors in code 68
procedural design in 329
pseudocode
determining hits and misses 59–60, 76
determining variables using 50
implementing loop using 51
translating into JavaScript code 52
working through 47–48, 73
behavior and state, in objects 210–212, 226
block of code 17, 23

EBSCO Publishing : eBook Collection (EBSCOhost) - printed on 7/28/2014 3:08 PM via FULL SAIL UNIVERSITY
AN: 748459 ; Freeman, Eric, Robson, Elisabeth.; Head First JavaScript Programming
Account: s9105391

<body> element
- about 4
- adding code on HTML page to 32
- replacing element content with innerHTML property 245

body, function
- declaring variables within 97
- parameters in 84–85

BOM (Browser Object Model) 627

boolean expressions
- true and false values 13, 15–16, 291–293, 313
- using to make decisions 22

boolean operators
- being verbose in specifying conditionals using 65, 74
- comparison vs. logical 71
- guide to 55
- writing complex statements using 62–64, 74–75

booleans
- about 11, 23
- as objects 296
- as primitive types 266
- using == operator with other types 277–278, 281

Browser Object Model (BOM) 627

browsers
- conditional statements handled by 434
- events 404
- executing code 433
- function declarations handled by 431–433, 436, 465, 483
- function expressions handled by 434–436, 465
- handling events 383, 403. *See also* handling events
- loading and executing JavaScript in 3
- loading code into 31
- objects provided by 214
- opening console 28
- parsing HTML and building DOM 236, 261
- recommended xxvi
- running JavaScript in xxxii
- setting up Event object in IE8 and older 399
- setting up event objects in IE8 and older 631
- tailoring size of window to web page 627

browser wars 6

built-in functions 71, 91, 119

built-in objects 548–551, 608–610

C

callbacks. *See also* event handlers 250

calling (invoking) functions
- about 85–86, 492–493
- browsers handling function declarations and expressions 431–437
- exercise for 117
- parentheses in 68, 430, 439
- recursive functions 634–635
- variables referring to functions in 436
- with arguments 90, 430

call method 603–604

camel case in multi-word variable names 13

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

case sensitivity in JavaScript 12

catch/try, handling exceptions using 629

cells, in table of advanced Battleship game 322

chaining 345, 348

chain of prototypes
- inheritance in 592
- setting up 591

characters, in JavaScript 297

character strings, quoting in JavaScript 13

charAt method 297

classes
- class attribute 255
- hit and miss classes for advanced Battleship game 326–327, 375
- labeling set of elements with 243

classic object-oriented programming 564

clear code, writing 12

cliche method 610

click events 407, 419, 624

click handlers, assigning 396–397

close() method 627

closures
- about 493, 495–497
- actual environment in 502

EBSCO Publishing : eBook Collection (EBSCOhost) - printed on 7/28/2014 3:08 PM via FULL SAIL UNIVERSITY
AN: 748459 ; Freeman, Eric, Robson, Elisabeth.; Head First JavaScript Programming
Account: s9105391

Copyright © 2014. O'Reilly. All rights reserved. May not be reproduced in any form without permission from the publisher, except fair uses permitted under U.S. or applicable copyright law.

EBSCOhost

creating with event handler 503–507
exercise for creating 500, 514–515
passing function expression as argument to create 501
using to implement magic counters 498–499

Coda, as HTML editor 31

code. *See also* Battleship game, advanced; Battleship game, simplified; writing JavaScript
analyzing 80, 116
browser executing 433
compiling 5
duplicating code in objects 208
finding errors in
exercise 14, 39, 203, 224
using console 68
using console.log function 25, 27
for JavaScript application 29–35
functions and reusing 83–88
guide to hygiene of 111
loading into browser 31
moving JavaScript from HTML page to file 33–34
recursive 635
refactoring 156, 159
reusing , 71, 79, xii

code block 17, 23

code file
linking external file from HTML page <body> element 32
linking from HTML page <head> element 32

collision method 364, 368

colon (:), separating property name and property value 179

comma (,), separating object properties 177, 179

comments in JavaScript 13

comparison operators 55, 71

compiling code 5

concatenating strings 15, 133, 142, 286–287, 312, 354

concat, method 300

conditionals
as variable or string 23
being verbose in specifying 65, 74
combining using boolean operators 62–63, 74
handled by browsers 434
test in arrays 140
using boolean expressions to make decisions with 22
using in while statement 17–21

confirm method 627

console
about 214
cheating at advanced Battleship game using 370
finding errors using 68
opening browser 28

console.log function
communicating with users using 25–27, 42
displaying object in console using 185, 608

constructor object, array 549–551

constructors, object
about , 521, 525, xxi
creating 525–526, 536–538, 555, 558, 566–567
creating objects
by convention 523
with object literals 522
finding errors in code exercise 529, 556
independent properties of 546–547
naming 532
parameters names in 532
putting methods into 530–532, 557
real world 548
understanding object instances 543–545, 560
updating constructor
chain of prototypes in 591–593, 606, 619
cleaning up code 600–605
creating prototype that inherits from another prototype 594–599, 620
design for 588–590
using 527, 555
using new keyword with 528, 532–535, 543, 548
vs. literals 532, 539–542, 559
workings of 528

controller object, advanced Battleship game
about 329
counting guesses and firing shot 355–356
determining end of game 356
developing parseGuess function 351–354
implementing 349–357
passing input to 360
processing player's guess 350–354
testing 357

counter, initializing, in array 140

Copyright © 2014. O'Reilly. All rights reserved. May not be reproduced in any form without permission from the publisher, except fair uses permitted under U.S. or applicable copyright law.

EBSCO Publishing : eBook Collection (EBSCOhost) - printed on 7/28/2014 3:08 PM via FULL SAIL UNIVERSITY
AN: 748459 ; Freeman, Eric, Robson, Elisabeth.; Head First JavaScript Programming
Account: s9105391

CSS (Cascading Style Sheets)
- about 2
- background property 326
- creating interactive web page using 319
- marking up text with 10
- positioning 324–326, 328
- writing 254, 263

curly braces ({ })
- enclosing object properties 177, 179
- in body of function 97
- in code block 17
- matching in code 59
- using with object literals 177, 522

D

Date object 214, 548, 551

declarations, function
- assigning to variables 439
- evaluating code 438, 466
- handled by browser 431–433, 436, 465, 483
- in anonymous functions 483–485
- parsing 431
- vs. function expressions 431, 437

defining functions, with parameters 90

delete keyword 184

designing
- simplified Battleship game game 45–46

designing, advanced Battleship game game 329–330, 376

diagrams, object 565, 570

displayHit method 331–334

displayMessage method 331–334

displayMiss method 331–334

<div> elements
- giving unique id 243
- positioning using CSS 324–325

document object
- about 214, 240
- getElementById method
  - accessing images using 388–392
  - as case sensitive 230
  - as document object 237
  - getting element with 231, 240, 247–248
  - getting reference to fire button in advanced Battleship game 359
  - passing id that does not exist 245
  - returning null 256, 270–271
  - using to locate element and change its content 238–239
- getElementsByClassName method 245, 626
- getElementsByName method 626
- getElementsByTagName method 245, 397
- in DOM 235, 626
- write function, communicating with users using 25–26, 42

dollar sign ($)
- beginning JavaScript variable names 13
- function in jQuery 624

DOM (Document Object Model)
- about , xv
- accessing images in 389–390
- changing 244
- communicating with users using 25–26, 42
- creating 234–235
- creating, adding, or removing elements getting 258
- document object in 235, 626
- DOM structure as tree 235
- elements grabbed from 241
- Event object in 399–402, 423
- events 413
- getting element by id from 245
- getting elements from, using jQuery 624
- interaction of JavaScript with 233
- NodeList in 397
- parsing HTML and building 236, 261
- setting attributes
  - with code 254
  - with setAttribute method 255, 333
- updating 247–248
- with secret message 246

"Don't Repeat Yourself" (DRY) 603

dot notation (.)
- accessing object properties 181, 209, 230
- using with reference variables 186
- using with this object 202

do while loop 364, 373

dragstart event 419

Dreamweaver, as HTML editor 31

EBSCOhost®

Copyright © 2014. O'Reilly. All rights reserved. May not be reproduced in any form without permission from the publisher, except fair uses permitted under U.S. or applicable copyright law.

EBSCO Publishing : eBook Collection (EBSCOhost) - printed on 7/28/2014 3:08 PM via FULL SAIL UNIVERSITY
AN: 748459 ; Freeman, Eric, Robson, Elisabeth.; Head First JavaScript Programming
Account: s9105391

Copyright © 2014. O'Reilly. All rights reserved. May not be reproduced in any form without permission from the publisher, except fair uses permitted under U.S. or applicable copyright law.

drop event 419
DRY ("Don't Repeat Yourself") 603

E

ECMAScript 5 6
element objects
  about 231, 245
  returning list of 397
elements. *See also* getElementById method
  accessing with getElementById 238, 240
  attribute does not exist in 256
  classes used to label set of 243
  getting by id from DOM 245
  getting, creating, adding, or removing 258
  grabbed from DOM 241
  identifying elements 243
  labeling with classes set of 243
  target set to 398–399, 401
  using CSS to position 324–326
else if statements
  making decisions using 22–23
  writing 56
else statements, using as catch-all with if/else statements 23
empty arrays 134, 153–154, 365, 367, 549–550
encapsulation 200
equal sign (=) operator, assigning values to variables using 11, 16, 275
equality (==) operator
  as comparison operator 16, 55, 275–285, 311, 459
  vs. === operator 289
strict equality (===) operator
  as comparison operator 55, 280–285, 311
  vs. == operator 289
Error object 551, 629
errors in code, finding
  exercise 14, 39, 203, 224
  finding errors using console 68
  handling exceptions 629
  using console.log function 25, 27
event handlers
  about 384
  adding using addEventListener 630
  assigning to properties 407
  asynchronous coding
    assigning to properties 407
    callbacks and 250
    creating 385–386
    in advanced Battleship game 358–359, 361
    kinds of 252
    onload 249
    timerHandler 407
    using setInterval function 410, 425
    using setTimeout function 407–413
  callbacks and 250
  creating 385–386
  creating closure with 503–507
  event objects 399–402, 423
  in advanced Battleship game 358–359, 361
  kinds of 252
  onload 249
  timerHandler 407
  using setInterval function 410, 425
  using setTimeout function 407–413
Event object
  event handlers and 399–402, 423
  for properties 401
  setting up in IE8 and older browsers 399, 631
event objects
  setting up in IE8 and older browsers 631
events
  about 383, 419
  API-specific 413
  click 407, 419, 624
  creating game that reacts to 388–392
  DOM 413
  dragstart 419
  drop 419
  exercise on notification of 382, 421
  exercise on reacting to events 415, 427
  image game and
    adding images to 393–397
    assigning click handlers 396–397
    assigning handler to onclick property 390–391, 393–398
    creating 411–414, 422, 426
  interview with browser about events 403
  I/O 413
  keypress 419

EBSCO Publishing : eBook Collection (EBSCOhost) - printed on 7/28/2014 3:08 PM via FULL SAIL UNIVERSITY
AN: 748459 ; Freeman, Eric, Robson, Elisabeth.; Head First JavaScript Programming
Account: s9105391

list of 419
load 419. *See also* load events
mousemove 405–406, 419, 424
mouseout 419
mouseover 419
onclick property 390–391, 393–398
pause 419
queues and 404
reacting to 387
resize 419
timer 407–409
touchend 419
touchstart 419
types of 407
unload 419
using setInterval function 410, 425

example files, downloading xxxiii

exceptions, handling 629

exercises, about doing xxxiii

expressions 15–16, 40

expressions, function
assigning to variables 439
evaluating code 438, 466
handled by browser 434–436, 465
substituting 481
vs. declarations, function 431, 437

F

false (boolean value) 13

falsey and truthy values 291–293, 313

fire method, advanced Battleship game 342, 346

first class values, functions as
about 442–443
extreme JavaScript challenge 486, 508, 517
passing functions to functions 443–448, 468, 482, 486, 514
returning functions from 450–456, 470, 472
substituting function expressions 481
using array sort method 457–463, 472–473

Flash, creating dynamic web pages using 5

flowchart, for designing program 45

for/in iterator 209

for loop
in adding new ship locations in advanced Battleship game 365, 367
in guessing location of ships in advanced Battleship game 343
iterating arrays using 140–142, 169
redoing 147
vs. while loop 144

<form> element
action attribute in 328
adding inputs to 323
using CSS to position 325

forward slashes (//), beginning JavaScript comments 13

forward slash (/) operator, as division arithmetic operator 15, 286

free variables 495–496, 501, 516

functions
$ (dollar sign), in jQuery 624
about , 79, 88, xii
abstracting code from array for 157–162
alert
communicating with users using 25–26, 42, 46
determining hits and misses in simplified Battleship game 59, 76
anatomy of 97
anonymous
about , 475–476, 482, xx–xxi
assigning to method in constructors 530–532, 557
brain twister involving 509, 518
creating 477–478, 512–513
making code tighter 479
passing functions to functions 482, 486, 514
readability of 480
arguments in
about 85
identifying 91, 105
mixing up order of 97
passing 88–89, 92–94
using pass-by-value 92–93
vs. parameters 90
arguments object in 628
as first class values
about 442–443
extreme JavaScript challenge 487, 508, 517

Copyright © 2014. O'Reilly. All rights reserved. May not be reproduced in any form without permission from the publisher, except fair uses permitted under U.S. or applicable copyright law.

EBSCOhost®

EBSCO Publishing : eBook Collection (EBSCOhost) - printed on 7/28/2014 3:08 PM via FULL SAIL UNIVERSITY
AN: 748459 ; Freeman, Eric, Robson, Elisabeth.; Head First JavaScript Programming
Account: s9105391

Copyright © 2014. O'Reilly. All rights reserved. May not be reproduced in any form without permission from the publisher, except fair uses permitted under U.S. or applicable copyright law.

passing functions to functions 443–448, 468, 482, 486, 514
returning functions from 450–456, 470, 472
substituting function expressions 481
using array sort method 457–463, 472–473
as objects 612–613
assigning to variables 439, 449, 469
as values 439–440, 467
body of
declaring variables within 97
parameters in 84–85
built-in 71, 91, 119
calling 85–86, 117
calling recursively 634–635
closures and
about 493, 495–497
actual environment in closures 502
creating closure with event handlers 503–507
exercise creating 500, 514–515
passing function expression as argument to create 501
using to implement counters 498–499
console.log
communicating with users using 25, 42
displaying object in console using 185
creating 83–87
declarations
about 430
assigning to variables 439
evaluating code 438, 466
handled by browser 431–433, 436, 465, 483
in anonymous functions 483–485
parsing 431
vs. expressions 431, 437
defining 132
exercise identifying 91, 119
expressions
assigning to variables 439
evaluating code 438, 466
handled by browser 434–436, 465
substituting 481
vs. declarations 431, 437
extreme JavaScript challenge 487, 508, 517
guide to code hygiene 111
init 249, 251, 359, 361, 369, 389–391
isNaN 273, 353
keyword 430
life span of variables 102
location in JavaScript 109
Math.floor 68, 131, 365–367
Math.random 67–68, 88, 131, 365–367
naming 97
nesting 485–486, 488–490, 494
No Dumb Questions 97, 108
onload handler 249–250, 359, 389–390
order in defining 389
parameters in
about 84, 88
assigning argument values to 85
identifying 91, 105
naming 97
none used in function 94
passing arguments to 89
vs. arguments 90
parameters vs. arguments 90
parentheses (()) in calling 68, 430, 439
passing functions to 409, 443–448, 468, 482, 486, 514
passing objects to 192–194, 198
prompt 46, 53, 55
reference
about 430, 476
action in function 491, 494
assigning 477
in calling function 491
in passing function argument to another function 486
passing 479
substituting function expressions and 481
returning functions from 450–456, 470, 472
return statement in 95–97
scope
lexical scope of variables 488–490, 494
of local and global variables 101
setInterval 410, 413, 425, 627
setTimeout 407–413, 410, 480, 501, 627
standard declaration for 430
timerHandler 407
tracing through 96
variables in
declaring inside and outside of functions 97–99
declaring local 103, 108
default value of 50

EBSCO Publishing : eBook Collection (EBSCOhost) - printed on 7/28/2014 3:08 PM via FULL SAIL UNIVERSITY
AN: 748459 ; Freeman, Eric, Robson, Elisabeth.; Head First JavaScript Programming
Account: s9105391

lexical scope of 488–490, 494
local vs. global 99, 106–107
naming 12–13, 100, 103, 108
nesting of 486, 488–490, 494
passing into arguments 89
reloading page and 108
scope of local and global 101
shadowing 103
var keyword in declaring 97, 99
vs methods 206
weird 94
write 25–26, 42

G

generateShipLocations method 364, 367, 369
generateShip method 364–365, 367
getAttribute method 256, 391
getDay method 548
getElementById method
accessing images using 388–392
as case sensitive 230
as document object 237
getting element with 231, 240, 247–248
getting reference to fire button in advanced Battleship game 359
passing id that does not exist 245
returning null 256, 270–271
using to locate element and change its content 238–239
getElementsByClassName method 245, 626
getElementsByName method 626
getElementsByTagName method 245, 396, 397
getFullYear method 548
getSize method 531, 557
global variables
guide to code hygiene 111
identifying 105, 120
lexical scope of 488–490, 494
life cycle of 102
nesting affecting 486, 488–490, 494
overuse in JavaScript 108
overuse of 391
reasons for sparing use of 108
scope of 101
shadowing 104
vs. local variables 99, 106–107
greater than (>) operator 16, 55, 459
greater than or equal to (>=) operator 16, 55
guesses property, advanced Battleship game 349

H

handleFireButton function 359
handleKeyPress function 361
handling events
asynchronous coding
about 383
assigning to properties 407
creating 385–386
creating closure with 503–507
in advanced Battleship game 358–359, 361
kinds of 253
onload 249
timerHandler 407
using setInterval function 410, 425
using setTimeout function 407–413
event handlers
about 383
adding using addEventListener 630
assigning to properties 407
callbacks and 250
creating 385–386
creating closure with 503–507
in advanced Battleship game 358–359, 361
kinds of 252
onload 249
timerHandler 407
using setInterval function 410, 425
using setTimeout function 407–413
event objects 399–402, 423
exercise on notification of events 382, 383, 421
interview with browser about events 403
reacting to events
about 387
adding images to image guessing game 393–397
assigning click handlers 396–397
assigning handler to onclick property 390–391, 393–398

Copyright © 2014. O'Reilly. All rights reserved. May not be reproduced in any form without permission from the publisher, except fair uses permitted under U.S. or applicable copyright law.

EBSCOhost®

EBSCO Publishing : eBook Collection (EBSCOhost) - printed on 7/28/2014 3:08 PM via FULL SAIL UNIVERSITY
AN: 748459 ; Freeman, Eric, Robson, Elisabeth.; Head First JavaScript Programming
Account: s9105391

Copyright © 2014. O'Reilly. All rights reserved. May not be reproduced in any form without permission from the publisher, except fair uses permitted under U.S. or applicable copyright law.

creating image guessing game 388–392, 411–414, 422, 426
exercise on 415, 427
using setInterval function 410, 425
using setTimeout function 407–413
hasOwnProperty method 586, 607
<head> element
about 4
adding code on HTML page to 32
linking code on HTML page file from 32
Head First HTML5 Programming 413, 613, 626
Head First HTML and CSS 320, 328
history method 627
hits, in advanced Battleship game
classes for misses and 326–327, 375
determining 344–345
determining if ship is sunk 346
looking for 343
notifying view of 347
HTML
about 2
creating interactive web page using
about 319–320
adding CSS 324–326
background 320–321
placing hits and misses 326
player interaction 322
table 322
using hit and miss classes 327, 375
editors 31
identifying elements with ids 243
linking code in simplified Battleship game 49
marking up text with 10
parsing and building DOM from 236, 261
HTML5
standard doctype 4
using string of numbers for id attributes in 328
using with JavaScript 5
<html> element 4
HTML page
JavaScript interacting with 233
loading code into 32
with code in external file 230–232
HTML wrapper for examples xxxiv

I

id attributes
accessing elements by 238
identifying elements with 243
using string of numbers for 328
identity equality operator 284. *See also* strict equality (===) operator
IE8 (Internet Explorer 8)
setting up event objects in browsers older than 399, 631
if/else statements
making decisions using 23
writing 56
if statement, making decisions using 22
image guessing game
adding images to
about 393–397
assigning click handlers 396–397
assigning handler to onclick property 390–391, 393–398
creating 388–392, 411–414, 422, 426
indexOf method 298, 343, 345, 352, 368, 610
indices, in arrays 129, 134, 152, 163
infinite loop 55
Infinity (-infinity), JavaScript 274
inheritance
extending built-in object 610
in chain of prototypes 592–593, 619
in objects 569–572
of prototype from another prototype 594–599, 606, 620
overriding built-in behavior 608–609
init function 249, 251, 359, 361, 369, 389–391, 630
innerHeight property 627
innerHTML property
about 245
changing DOM using 244
replacing element content in <body> element with 245
setting text using 331
using to change element content 239, 241–242, 247–248

EBSCO Publishing : eBook Collection (EBSCOhost) - printed on 7/28/2014 3:08 PM via FULL SAIL UNIVERSITY
AN: 748459 ; Freeman, Eric, Robson, Elisabeth.; Head First JavaScript Programming
Account: s9105391

Copyright © 2014. O'Reilly. All rights reserved. May not be reproduced in any form without permission from the publisher, except fair uses permitted under U.S. or applicable copyright law.

innerWidth property 627

<input> elements
- adding to <form> 323
- using CSS to style 325

instanceof operator 305, 315, 543, 547

interface transformations on elements, using jQuery 625

Internet Explorer 8 (IE8)
- setting up event objects in browsers older than 399, 631

interpreted languages 5

in variables 186–187

invoking (calling) functions
- about 85–86, 117, 492–493
- browsers handling function declarations and expressions 431–437
- exercise for 117
- parentheses (()) in 68, 430, 439
- recursive functions 634–635
- variables referring to functions in 436
- with arguments 90, 430

I/O events 413

isNaN function 273, 353

isSunk method, advanced Battleship game 346

J

Java
- introduction of 6
- vs. JavaScript 5

Java, classic object-oriented programming 564

JavaScript
- about , 1–2, x–xi
- API-specific events 413
- arrays 127, 132–134
- case insensitivity 12
- getElementById as wormhole to 237
- getting on web page 4–5
- grand unified theory of 612
- handling exceptions 629
- history of 6
- importance of 5
- in applications 5
- Infinity in 274
- interacting with DOM 233
- interaction with web page through DOM 237
- in web pages 2–3
- jQuery 624
- learning 9
- node.js library 637
- objects provided by 214
- reserved words 12
- server-side 637
- syntax 13–14, 39
- thinking about xxix
- values in 272–274, 281, 292
- vs. Java 5
- writing. *See* writing JavaScript

JavaScript: The Definitive Guide 296

jQuery
- about 624–625
- online documentation and tutorials 625

JScript 6

JSON object 214, 636

K

keypress event 419

keywords 12

L

lastIndexOf, method 300

learning
- JavaScript 9
- tips for xxxi

learning principles xxviii

length property 297, 301, 303

length property, in arrays 130

less than (<) operator 55, 459

less than or equal to (<=) operator 55

lexical scope of variables 488, 494

life cycle of variables 102

list
- adding songs to playlist with JavaScript (example) 253, 262
- NodeList 397
- returning element objects in 397

EBSCOhost®

EBSCO Publishing : eBook Collection (EBSCOhost) - printed on 7/28/2014 3:08 PM via FULL SAIL UNIVERSITY
AN: 748459 ; Freeman, Eric, Robson, Elisabeth.; Head First JavaScript Programming
Account: s9105391

listener 384. *See also* event handlers

list of event 419

lists
  adding songs to playlist with JavaScript (example) 253, 262
  finding all child <li> elements of element with id of playlist, using JQuery 625

literals
  creating objects with 522
  vs. constructors 532, 539–542, 559

literals, in arrays 151–152

LiveScript 6

load event
  about 419
  onload property
    assigning handler to 385–386, 389–390, 410
    in anonymous functions 476–477
    setting to function 249, 359, 392, 394–395, 406–407, 422

local variables
  declaring 103, 108
  guide to code hygiene 111
  identifying 105, 120
  in methods 206
  lexical scope of 488–489, 494
  life cycle of 102
  nesting affecting 486, 488–490, 494
  scope of 101
  shadowing global variable 104
  vs. global variables 99, 106–107

location method 627

logical operators 55, 71

loop
  do while 364, 373
  for
    in adding new ship locations in advanced Battleship game 365, 367
    in guessing location of ships in advanced Battleship game 343
    iterating arrays using 140–142, 169
    redoing 147
    vs. while loop 144–145
  implementing using pseudocode 51
  infinite 55
  translating pseudocode into JavaScript code 52
  using inner and outer 368
  while
    about 17
    executing code using 18–21
    vs. for loop 144–145

M

match method 300, 633

Math.floor function 68, 131, 365–367

Math object 214, 551

Math.random function 67–68, 88, 131, 365–367

metacognition xxix

methods
  about 198
  adding to prototypes 581–582, 584
  available in JavaScript 296
  designing advanced Battleship game 330–331, 376
  duplicating 568
  extending String prototype with 610–611, 621
  in chain of prototypes 592–593, 619
  putting in constructors 530–532, 557
  tour of string 297–300
  using this object in 202, 204–206, 219
  vs. functions 206
  writing 199–200, 207, 225

Microsoft, introduction of JScript 6

Mighty Gumball, Inc. 135–136, 148, 150

minus sign (-) operator
  as unary operator 287
  using as arithmetic operator with string and number 286–287, 312

misses, in advanced Battleship game
  classes for hits and 326–327, 375
  notifying view of 347

model object, advanced Battleship game
  about 329, 336
  determining hits 344–345
  determining if ship is sunk 346
  fire method in 342–343, 346
  implementing 341–348
  interacting with view 336
  keeping track of ships 337

Copyright © 2014. O'Reilly. All rights reserved. May not be reproduced in any form without permission from the publisher, except fair uses permitted under U.S. or applicable copyright law.

EBSCOhost®

EBSCO Publishing : eBook Collection (EBSCOhost) - printed on 7/28/2014 3:08 PM via FULL SAIL UNIVERSITY
AN: 748459 ; Freeman, Eric, Robson, Elisabeth.; Head First JavaScript Programming
Account: s9105391

notifying view of hits or misses 347
parseGuess function asking about size of board 353
representation of ships in 338–340, 378–379
representing state of ships in 337, 377
testing 348

mousemove event 405–406, 419, 424

mouseout event 419

mouseover event 419

N

naming
constructors 532
functions 97
local and global variables with same name 104
properties 179
variables 12–13, 100, 103, 108

NaN (Not a Number) values 272–274, 281, 292

nesting functions 485–486, 488–490, 494

Netscape 6

new keyword, using with constructors 528, 532–535, 543, 548

node.js library 637

NodeList 397

Notepad (Windows), as text editor 31

not equal to (!=) operator 16, 55

NOT (!) operator 55

null 55, 256, 270–271, 274, 278, 292

numbers
as objects 296
as primitive types 266
comparing to strings 275–277, 281
NaN (Not a Number) values and 273–274, 281, 292
prompt function returning strings for 55
using + sign with strings and 286–287

O

Object, as object
about 551, 607
as prototype 607
overriding built-in behavior 608–609

object constructors
about , 521, 525, xxi
creating 525–526, 536–538, 555, 558, 566–567
creating objects, by convention 523
finding errors in code exercise 529, 556
independent properties of 546–547
naming 532
parameters names in 532
putting methods into 530–532, 557
real world 548
understanding object instances 543–545, 560
updating constructor
chain of prototypes in 591–593, 606, 619
cleaning up code 600–605
creating prototype that inherits from another prototype 594–599, 620
design for 588–590
using 527, 555
using new keyword with 528, 532–535, 543, 548
vs. literals 532, 539–542, 559
workings of 528

object instances 543–545, 560

object models, building 3

object-oriented programming 180, 564

object prototype model
about , xxii–xxiii
better living through objects 612–613
chain of prototypes
inheritance in 592–593, 619
setting up 591
determining if properties are in instance or in prototype 586–587, 592, 618
duplicating methods and 568
extending String prototype with method 610–611, 621
implementing code 579, 583, 616–617
inheritance
extending built-in object 610
in chain of prototypes 592–593, 619
in objects 569–572
of prototype from another prototype 594–599, 606, 620
overriding built-in behavior 608–609
Object
about 607

Copyright © 2014. O'Reilly. All rights reserved. May not be reproduced in any form without permission from the publisher, except fair uses permitted under U.S. or applicable copyright law.

EBSCOhost®

EBSCO Publishing : eBook Collection (EBSCOhost) - printed on 7/28/2014 3:08 PM via FULL SAIL UNIVERSITY
AN: 748459 ; Freeman, Eric, Robson, Elisabeth.; Head First JavaScript Programming
Account: s9105391

Copyright © 2014. O'Reilly. All rights reserved. May not be reproduced in any form without permission from the publisher, except fair uses permitted under U.S. or applicable copyright law.

as prototype 607
overriding built-in behavior 608–609
object diagram exercise 574, 615
prototypes in
about 569
adding methods to 581–582, 584
changing properties in 582, 585
dynamic 582
getting 575
overriding 573, 578
setting up 575
updating constructor
chain of prototypes in 591–593, 606, 619
cleaning up code 600–605
creating prototype that inherits from another prototype 594–599, 620
design for 588–590
object references 187
objects
about , 173–174, 215, xiv–xv
adding behavior to 198–201
arguments 628
array 549–551
array constructor 549–551
Auto-O-Matic app (example) 195–197
behavior 568
better living through 612–613
booleans as 296
built-in 548–551, 608
callback 250
comparing 459
console 214
controller (advanced Battleship game)
about 329
counting guesses and firing shot 355–356
determining end of game 356
developing parseGuess function 351–354
implementing 349–357
passing input to 360
processing player's guess 350–354
testing 357
creating 177–179, 183, 185, 188–191, 220–222
creating with object literals 522, 532, 539–542, 559
Date 214, 548
diagramming 565, 570
document
about 214, 240
getElementById method as 238. *See also* getElementById method
in DOM 235, 626
write function, communicating with users using 25–26, 42
duplicating code in 208
element 231
equality of 288–290, 314
Error 551
Event
event handlers and 399–402, 423
for properties 401
setting up in IE8 and older browsers 399, 631
finding errors in code (exercise) 203
functions as 612–613
inheriting 569–572
iterating through properties of 209
JavaScript provided 214
JSON 214
Math 214, 551
methods 198
methods vs. functions 206
model (advanced Battleship game). *See* model object, advanced Battleship game
models, building 3
No Dumb Questions 185
number of levels deep in arrays to nest 348
numbers as 296
passing to functions 192, 192–194, 198
properties in
about 175
accessing 181
adding new 182
adding or deleting at any time 184
changing 182
computing with 182
iterating through 209
undefined values and 268
values of 176, 219
putting methods into, by convention 523
RegExp 214, 551, 632–633
rules of road for creating 179
state and behavior in 210–212, 226
strings as primitives and 294–296

EBSCO Publishing : eBook Collection (EBSCOhost) - printed on 7/28/2014 3:08 PM via FULL SAIL UNIVERSITY
AN: 748459 ; Freeman, Eric, Robson, Elisabeth.; Head First JavaScript Programming
Account: s9105391

this
in constructors 528–529, 532
prototypes and 580
using 202
using with objects 204–206, 219
undefined values and 268
understanding object instances 543–545, 560
using null 270–271, 274
using this 202, 204–206, 219
variable declaration for 177
in variables 186
vs. primitives 187
window
about 214
creating onload event handler for 249, 402
history method 627
innerHeight property 627
innerWidth property 627
location method 627
methods and properties available for 627
onload property. *See* onload property
onresize property 415, 427
print method 627
setInterval method 627
setTimeout method 407–413, 627
support of BOM in 627

onclick property 390–391, 393–398, 504–507, 630

onload event handler 249, 359, 402

onload handler function 249–250, 359, 389–390

onload property
assigning handler to 385–386, 389–390, 410, 630
in anonymous functions 476–477
setting to function 249, 359, 392, 394–395, 406–407, 422
support of BOM using 627

onmousemove property 405–406, 414, 424, 426

onmouseout property 414, 426

onresize property 415, 427

OR (||) operator 54, 55, 62–63, 74

P

parameters, function
about 84, 88
assigning argument values to 85
identifying 91, 105, 119, 120
names in constructors 532
naming 97
none used in function 94
passing arguments to 89
vs. arguments 90

parentheses ()
in calling functions 68, 430, 439
in parameters 97

parseGuess function 351–354

parse method, in JSON 636

parsing function declaration 432

pass-by-values 92–93

passing arguments 88–89

passing by reference 192

passing by value 192

pause event 419

Phrase-o-Matic application (example) 131–133

playlists
in jQuery 625
populating with song titles using JavaScript array 253, 262
using document objects in 626

plus sign (+)
as arithmetic operator 286–287, 312
in concatenating strings 15, 133, 142, 354

position: absolute 324–325, 328

positioning, CSS 324–326, 328

position: relative 324, 328

post-decrement operator (--) 146–147

post-increment operator (++) 146–147

<p> (paragraph) elements
changing using JavaScript 247
innerHTML property changing 242

EBSCOhost®

Copyright © 2014. O'Reilly. All rights reserved. May not be reproduced in any form without permission from the publisher, except fair uses permitted under U.S. or applicable copyright law.

EBSCO Publishing : eBook Collection (EBSCOhost) - printed on 7/28/2014 3:08 PM via FULL SAIL UNIVERSITY
AN: 748459 ; Freeman, Eric, Robson, Elisabeth.; Head First JavaScript Programming
Account: s9105391

primitive types
  about 23, 266
  identifying undefined values 267–268, 308
  strings as objects and 294–296
  vs. objects 187
print method 627
procedural design 329
processGuess method 349–350, 357
programming languages, learning 9
prompt function 46, 53, 55
properties
  about 175
  accessing 181
  adding new 182
  adding or deleting at any time 184
  available in JavaScript 296
  changing 182
  changing in prototypes 582, 585
  computing with 182
  determining if in instance or in prototype 586–587, 592, 618
  finding errors in code (exercise) 203
  in chain of prototypes 592–593, 619
  independent constructor 546–547
  inheriting 571
  iterating through object 209
  methods vs. functions 206
  naming 179
  No Dumb Questions 185
  objects as collections of 174
  tour of string 297–300
  undefined values and 268
  using this object 202, 204–206, 219
  values and 176, 219
prototypal inheritance 569–572
prototype model, object
  about , xxii–xxiii
  chain of prototypes
    inheritance in 592–593, 619
    setting up 591
  determining if properties are in instance or in prototype 586–587, 592, 618
  duplicating methods and 568
  extending String prototype with method 610–611, 621
  implementing code 579, 583, 616–617
  inheritance
    extending built-in object 610
    in chain of prototypes 592–593, 619
    in objects 569–572
    of prototype from another prototype 594–599, 606, 620
    overriding built-in behavior 608–609
  Object
    about 607
    as prototype 607
    overriding built-in behavior 608–609
  object diagram exercise 574, 615
  prototypes in
    about 569
    adding methods to 581–582, 584
    changing properties in 582, 585
    dynamic 582
    getting 575
    overriding 573, 578
    setting up 575
  updating constructor
    chain of prototypes in 591–593, 606, 619
    cleaning up code 600–605
    creating prototype that inherits from another prototype 594–599, 620
    design for 588–590
pseudocode
  determining hits and misses in simplified Battleship game 59–60, 76
  determining variables using 50
  implementing loop using 51
  translating into JavaScript code 52, 164–165, 171
  working through 47–48, 73
push method 152

Q

quality assurance (QA)
  about 61
  doing 61, 69–70
  in advanced Battleship game 370–371
querySelectorAll method 626
querySelector method 626

Copyright © 2014. O'Reilly. All rights reserved. May not be reproduced in any form without permission from the publisher, except fair uses permitted under U.S. or applicable copyright law.

EBSCOhost®

EBSCO Publishing : eBook Collection (EBSCOhost) - printed on 7/28/2014 3:08 PM via FULL SAIL UNIVERSITY
AN: 748459 ; Freeman, Eric, Robson, Elisabeth.; Head First JavaScript Programming
Account: s9105391

queues, events and 404
quotation marks, double (" ")
  surrounding character strings in JavaScript 13
  using around property name 179

R

random locations for ships, generating in advanced Battleship game 362–368, 380
random numbers, generating 67–68
reader as learner xxviii
recursion 634–635
refactoring code 156, 159
reference, function
  about 430, 476
  action in function 491, 494
  assigning 477
  in calling function 491
  in passing function argument to another function 486
  passing 479
  substituting function expressions and 481
references, object 192
RegExp object 214, 551, 632
removeEventListener 630
replace, method 300
reportError method 587, 618
reserved words 12
resize event 419
return statement, function 95–97
rules of road, for creating objects 179

S

sample files xxxiv
scope, variables 101
<script> element
  about 4
  anatomy of 35
  src attribute of 34–35
scripting languages 5
<script> tags
  in <head> or <body> element of HTML page 32
  opening and closing 35
semicolon (;), ending statements in JavaScript 11, 13
server-side JavaScript 637
setAttribute method 255, 333, 391
setInterval function 410, 413, 425
setInterval method 627
setTimeout function 410, 480, 501
setTimeout method 407–413, 627
shadowing variables 104
showAnswer handler 413
slashes (//), beginning JavaScript comments 13
slash (/) operator, as division arithmetic operator 15, 286
slice, method 300
sort method, array 457–463, 472
sparse Array 152
split method 299
square brackets ([ ])
  accessing properties using 209
  in arrays 127, 129, 550
src attribute, of <script> element 34–35
src property 390, 391
state and behavior, in objects 210–212, 226
statements
  ending with semicolon 11, 13
  variables in 11–13
  writing 10
strict equality (===) operator
  as comparison operator 55, 280–285, 311
  vs. == operator 289
String prototype, extending with method 610–611, 621
strings
  in arrays 132
  as primitives and objects 294–296
  as primitive types 266
  comparing to numbers 275–277, 281
  concatenating 15, 133, 142, 286–287, 312, 354
  conditional as 23
  falsey value are empty 292
  operators to sort 459
  prompt function returning for numbers 55
  properties available in JavaScript 296

Copyright © 2014. O'Reilly. All rights reserved. May not be reproduced in any form without permission from the publisher, except fair uses permitted under U.S. or applicable copyright law.

EBSCOhost®

EBSCO Publishing : eBook Collection (EBSCOhost) - printed on 7/28/2014 3:08 PM via FULL SAIL UNIVERSITY
AN: 748459 ; Freeman, Eric, Robson, Elisabeth.; Head First JavaScript Programming
Account: s9105391

secret life of 294–296
tour of methods and properties 297
substring method 299–300, 302, 304
Sun Microsystems 6
support material xxxiv
syntax, JavaScript
about 13
finding errors in 14, 39

T

<table> element
creating HTML page 322
using CSS to position 325
target 398–399, 401
<td> element 322, 325, 327–328, 333
TextEdit (Mac), as text editor 31
text editors 31
this object
in constructors 528–529, 532
prototypes and 580
using 202, 204–206, 219
timer events 407–409
timerHandler function 407–409
toLowerCase, method 300
toString method 531, 548, 557, 607–608
touchend event 419
touchstart event 419
toUpperCase, method 300
tracing flow of execution (tracing) 96
treasure map game 405–406, 424
tree, DOM structure as 235
trim, method 300
true and false (boolean values) 13, 15–16, 291–293, 313
truthy and falsey values 291–293, 313
try/catch, handling exceptions using 629
typeof operator 269, 305, 309, 315, 542
types
about , xvi–xvii
conversion 277–280
equality of object 288–290, 314
instanceof 305, 315, 543, 547
knowing 301–304
typeof exercise and 269
using == and === operators on different 275–285, 311
using arithmetic operators for conversion of 286–287, 312
using null 55, 270–271, 274, 310
values and 272–274, 281, 292

U

undefined, as default value of variables 50
undefined values
about 267–268, 308
as falsey 292
comparing to null 278
returned by functions without return statement 97
vs. null 270, 310
underscore (_), beginning variable names 13
Unicode values 281
unload event 419
users
communicating with
using alert function 25–26, 46
using console.log function 25–27
using document object model 25
using prompt function 46, 53

V

values
equality and 276
functions as 439–440, 467
functions as first class
about 442–443
extreme JavaScript challenge 487, 508, 517
passing functions to functions 443–448, 468, 482, 486, 514
returning functions from 450–456, 470, 472
substituting function expressions 481
using array sort method 457–463, 472–473
identifying undefined 267–268, 308
NaN (Not a Number) 272–274, 281, 292
null 270

Copyright © 2014. O'Reilly. All rights reserved. May not be reproduced in any form without permission from the publisher, except fair uses permitted under U.S. or applicable copyright law.

EBSCOhost®

EBSCO Publishing : eBook Collection (EBSCOhost) - printed on 7/28/2014 3:08 PM via FULL SAIL UNIVERSITY
AN: 748459 ; Freeman, Eric, Robson, Elisabeth.; Head First JavaScript Programming
Account: s9105391

object properties and 176, 219
objects 266
operators and 286
operators to sort 459
passing values to functions 92–93, 192
primitive 266
strict equality 280
strings 291–297
truthy and falsey 291–293
type conversion and 276–281
types in arrays 134
undefined 268
variables and 11

variable declaration, object 177

variables
about 11
assigning functions to 439, 449, 469
conditional as 23
declaring in arrays 152
declaring inside and outside of functions 98
declaring local 103, 108
default value of 50
finding errors in 39
free 495–496, 501, 516
global
guide to code hygiene 111
identifying 105, 120
lexical scope of 488–490, 494
nesting affecting 486, 488–490, 494
overuse of 391
scope of 101
vs. local variables 99, 106–107
guide to code hygiene 111
identifying 91, 119
in arrays 127, 129
lexical scope of 488–490, 494
local
declaring 103, 108
guide to code hygiene 111
identifying 105, 120
in methods 206
lexical scope of 488–490, 494
nesting affecting 486, 488–490, 494
scope of 101
vs. global variables 99, 106–107
naming 12–13, 100, 103, 108
objects assigned to 192
objects in 186
passing into arguments 89
referring to functions to invoke functions 436
reloading page and 108
scope of 101
setting to null 270
shadowing 104
short life of 102
undefined values and 268
using pseudocode in determining 50
var keyword in declaring 97, 99

var keyword
about 11
in declaring variables 97, 99

view object, advanced Battleship game
about 329
implementing 331–334
interacting with model 336
notifying of hits or misses 347
testing 335

W

W3C documentation website 627

web browsers
conditional statements handled by 434
events 404
executing code 433
function declarations handled by 431–433, 436, 465, 483
function expressions handled by 434–436, 465
handling events 383, 403. *See also* handling events
loading and executing JavaScript in 3
loading code into 31
objects provided by 214
opening console 28
parsing HTML and building DOM from it 261
recommended xxvi
running JavaScript in xxxii
setting up Event Object in IE8 and older 399
setting up event objects in IE8 and older 631
tailoring size of window to web page 627

web browser wars 6

web pages
adding code to HTML page 32
as applications 9

Copyright © 2014. O'Reilly. All rights reserved. May not be reproduced in any form without permission from the publisher, except fair uses permitted under U.S. or applicable copyright law.

EBSCOhost®
EBSCO Publishing : eBook Collection (EBSCOhost) - printed on 7/28/2014 3:08 PM via FULL SAIL UNIVERSITY
AN: 748459 ; Freeman, Eric, Robson, Elisabeth.; Head First JavaScript Programming
Account: s9105391

Copyright © 2014. O'Reilly. All rights reserved. May not be reproduced in any form without permission from the publisher, except fair uses permitted under U.S. or applicable copyright law.

creating dynamic 5–6
creating interactive 319
JavaScript in 2–3
JavaScript interacting with 233
tailoring size of browser window to 627
using <script> element in 4

while loop
about 17
executing code using 18–21
vs. for loop 144–145

white space in JavaScript code 13

wickedlysmart.com 640

window object
about 214
addEventListener method 630
confirm method 627
creating onload event handler for 249, 359, 402
history method 627
innerHeight property 627
innerWidth property 627
location method 627
methods and properties available for 627
onload property
assigning handler to 385–386, 389–390, 410, 630
in anonymous functions 476–477
setting to function 249, 359, 392, 394–395, 406–407, 422
support of BOM using 627
onresize property 415, 427
print method 627
setInterval method 627
setTimeout function 407–413
setTimeout method 627
support of BOM in 627

write function 25–26, 42

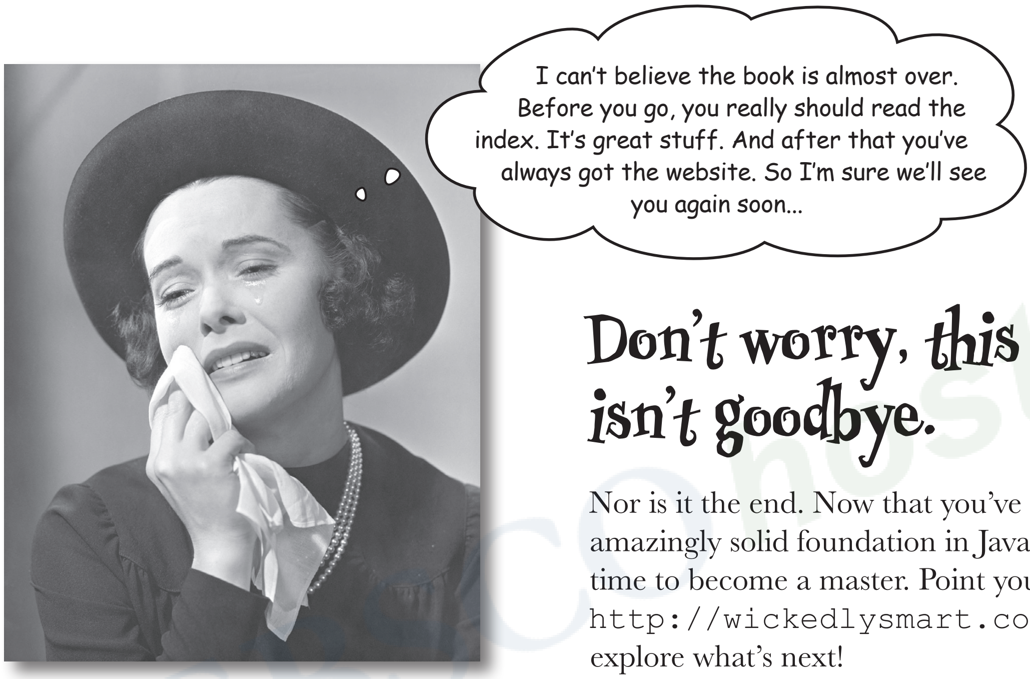
writing JavaScript. *See also* Battleship game, advanced; Battleship game, simplified; code
about 3, 7–8
analyzing code 80–81, 116
being verbose in specifying conditionals 65, 74
chaining 345, 348
comparing to numbers to strings 275–277, 281
doing quality assurance 61, 69–70
duplicating code in objects 208
else if statements, making decisions using 22–23
expressions in 15–16, 40
finding errors in code
exercise 14, 39, 203, 224
using console 68
flowchart for designing program 45
functions and reusing code 83–88
generating random numbers 67–68
guide to code hygiene 111
null in 256, 270–271, 274, 278
prompt function in 46, 53
pseudocode
determining hits and misses in simplified Battleship game 59–60, 76
determining variables using 50
implementing loop using 51
translating into JavaScript code 52, 164–165, 171
working through 47–48, 73
refactoring code 156, 159
reusing code , 71, 79, xii
statements 10
undefined as default value of variables 50
using = operator, assigning values to variables using 11, 16, 275
using == operator
as comparison operator 16, 55, 275–285, 311, 459
vs. === operator 289
using === operator
as comparison operator 55, 280–285, 311
vs. == operator 289
using variables 11–13
values in 272–274, 281
while statement, executing code using 18–21

Z

zero (0), as falsey value 292

EBSCO Publishing : eBook Collection (EBSCOhost) - printed on 7/28/2014 3:08 PM via FULL SAIL UNIVERSITY
AN: 748459 ; Freeman, Eric, Robson, Elisabeth.; Head First JavaScript Programming
Account: s9105391

Copyright © 2014. O'Reilly. All rights reserved. May not be reproduced in any form without permission from the publisher, except fair uses permitted under U.S. or applicable copyright law.


I can't believe the book is almost over. Before you go, you really should read the index. It's great stuff. And after that you've always got the website. So I'm sure we'll see you again soon...

Don't worry, this isn't goodbye.

Nor is it the end. Now that you've got an amazingly solid foundation in JavaScript, it's time to become a master. Point your browser to `http://wickedlysmart.com/hfjs` to explore what's next!

What's next? So much more! Join us at http://wickedlysmart.com/hfjs to continue your journey.

EBSCO Publishing : eBook Collection (EBSCOhost) - printed on 7/28/2014 3:08 PM via FULL SAIL UNIVERSITY
AN: 748459 ; Freeman, Eric, Robson, Elisabeth.; Head First JavaScript Programming
Account: s9105391

Copyright © 2014. O'Reilly. All rights reserved. May not be reproduced in any form without permission from the publisher, except fair uses permitted under U.S. or applicable copyright law.

EBSCOhost®

EBSCO Publishing : eBook Collection (EBSCOhost) - printed on 7/28/2014 3:08 PM via FULL SAIL UNIVERSITY
AN: 748459 ; Freeman, Eric, Robson, Elisabeth.; Head First JavaScript Programming
Account: s9105391

Copyright © 2014. O'Reilly. All rights reserved. May not be reproduced in any form without permission from the publisher, except fair uses permitted under U.S. or applicable copyright law.

EBSCOhost®

EBSCO Publishing : eBook Collection (EBSCOhost) - printed on 7/28/2014 3:08 PM via FULL SAIL UNIVERSITY
AN: 748459 ; Freeman, Eric, Robson, Elisabeth.; Head First JavaScript Programming
Account: s9105391