INDEX

SYMBOLS

+ (addition operator), 10, 20, 77
&& (and operator), 27, 33–34
= (assignment operator), 13
\ (backslash), 23
` (backtick), 24
^ (caret), 292
-- (decrement operator), 17, 64
/= (division assignment operator), 18
/ (division operator), 11
$ (dollar sign), 292
== (double equals operator), 31
" (double quotation mark), 19
// (forward slashes), 165, 292
> (greater than), 30
>= (greater than or equal to), 30
# (hash mark), 122–123, 215
++ (increment operator), 17
< (less than), 30
<= (less than or equal to), 30
*= (multiplication assignment operator), 18
* (multiplication operator), 11
! (not operator), 28
|| (or operator), 27–28, 33–34
() (parentheses), 79
% (percent character), 301
. (period), 123
| (pipe character), 241–242
${} (placeholder syntax), 24–25
? (question mark), 301
; (semicolon), 10
' (single quotation mark), 19
[] (square brackets), 21, 48–49
!== (strict inequality operator), 32
-= (subtraction assignment operator), 18
- (subtraction operator), 11
=== (triple equals operator), 29

A

absolute positions, 262–263
abstract classes, 98
accidentals, 215
addEventListener method, 129
add function, 75–76
addition assignment operator (+=), 18, 132
addition (+) operator, 20, 77
adjustAngle function, 174
ADSR envelope, 211–212
AI (artificial intelligence), 181
alert function, 5
Amazon S3 (Simple Storage Service), 233
ancestors, 114
and operator (&&), 27, 33–34
animating
bar graphs, 296–297, 322–324
canvas elements, 152–153
anonymous functions, 80, 153
APIs (application programming interfaces), 115
authenticated APIs, 302–303
Canvas API, 144, 256
DOM API, 115–116, 129
GitHub Search API, 299–328
JSON API, 300
Web Audio API, 206–209
appended elements, 275
arc method, 146–147
arguments, 21
defined, 5
parameters vs., 75
passing functions as, 74, 78–79
arrays, 38–47
arrays of arrays, 39–41
creation and indexing, 38–39
methods, 41–47
adding element to array, 42
combining arrays, 44–45
arrays (continued)
methods (continued)
  finding index of element in array, 45
  removing element from array, 43–44
  turning array into string, 45–46
nesting
  exploring nested objects in console, 54–55
  with literals, 52–53
  printing nested objects with JSON.stringify, 55–56
  with variables, 53–54
taking callback functions, 85–87
arr.includes(elem) method, 46
arrow functions, 82–83
arr.reverse() method, 46
arr.slice(start, end) method, 46–47
arr.sort() method, 46
arr.splice(index, count) method, 47
artificial intelligence (AI), 181
assignment operator (=), 13
attack, ADSR, 211–212, 234
attributes, 116–118
attr method, 269
audio context, 207–208
authenticated APIs, 302–303
axes
drawing, 305–307
labeling, 288–291
labeling left, 318–319
B
backend code, 303
backslash (\), 23
backticks (‘), 24
ball, Pong, 161–162, 195–197
bandpass filter, 226
bandwidth method, 287
bar graphs, 279–297, 304–319
  animating changes, 296–297
  calculating character frequencies, 280–282
  cleaning data, 295
  color-coding bars, 314–318
  creating elements, 304–305
drawing, 282–291
  axes, 305–307
  bars, 308–309
  labeled axes, 288–291
  scaling bars, 284–288
  labeling left axis, 318–319
  setting up, 279–280
  showing repository info, 310–314
  styling with CSS and regular expressions, 292–295
baseline, 257
base URL, 301
bass lines, 240–242
beats per minute (BPM), 217
bezierCurveTo method, 147
bindings, 12–16
  constants, 14–15
  naming conventions, 15–16
  variables, 13–14
block body syntax, 82–83
body
  of control structure, 59
  function, 74
body element, 113
Booleans, 26–30
  comparison operators, 29–30
  as conditions, 60–61
  logical operators, 27–28
  using subexpressions in, 171
bouncing
  near paddle ends, 174–175
  overview, 165–166
BPM (beats per minute), 217
braces, 63
bubbling events, 130
C
callbacks
  array methods taking, 85–87
  custom functions taking, 88–89
  defined, 78
  event handlers and, 129
calling functions, 74–78
  parameter types, 77
  return values, 75–76
  side effects, 77–78
calling methods, 21
call method, 289
camelCase, 15
C and C++, 334
Canvas API, 144, 256
canvas elements, 141–154
  animating, 152–153
  creating, 142
  drawing Pong game on, 160–161
  interacting with, 147–151
  making static drawings, 142–147
  drawing other shapes using paths, 145–147
  drawing outlined rectangles, 144–145
caret (^), 292
Cascading Style Sheets. See CSS
case sensitivity, 15
CDN (content delivery network), 210
chained if...else statement, 61–63
chaining methods, 130, 269
change event handler, 151
characters
  frequencies of, 280–282
  from strings, 21–22
checkCollision function, 165–166
child elements, 114
chords, 242–244
Chrome
  accessing JavaScript console, 4
  calling functions in, 75–76
  exploring nested objects in,
    54–55
  indentation in, 40
  Web Audio API, 206
circle element, 260–261
  advanced joins, 276–279
  data binding, 270–271
  data joins, 271–273
class attribute, 122
classes, 93–108
  creating instances and, 94–97
  inheritance, 97–101
  prototype-based inheritance,
    101–108
  comparing constructors and
classes, 104–105
  exploring Object.prototype,
    105–106
  overriding method, 107–108
using constructors and
  prototypes, 102–104
  walking the prototype chain,
    106–107
class selectors, 123
cleaning data, 295
clearInterval function, 83
clearRect method, 151
click handler, 148
clock, JavaScript, 220–221
closures, 90
CodePen, 331
correction, 30–35, 77
  equality with, 31–32
  truthiness, 32–35
collision detection
  bouncing, 165–166
  defined, 163–164
  in object-oriented Pong, 195–197
  for paddles, 170–173
color
  color-coding bars, 314–318
  RGBA color, 150
  RGB color, 261
  sound of oscillator as, 213
combining arrays, 44–45
combining objects, 51–52
comments, 165
comparison operators, 29–30
complete code
  GitHub Search API, 325–328
  Pong, 185–188
  song writing, 247–252
composition, 199
compound data types, 37–56
  arrays, 38–47
    arrays of arrays, 39–41
    creation and indexing,
      38–39
    methods, 41–47
  nesting objects and arrays,
    52–56
  exploring nested objects in
    the console, 54–55
    with literals, 52–53
  printing nested objects with
    JSON.stringify, 55–56
    with variables, 53–54
compound data types (continued)
  objects, 47–52
    accessing values in, 48
    creating, 47–48
    setting values to, 49
    working with, 49–52
compound expression, 10
Computer class, 197–198
computer control, 180–182
concat method, 44
concise body syntax, 82–83
concrete classes, 98
condition, of control structure, 58
conditionals, 57–63
  chained if...else statement, 61–63
  complex conditions, 60–61
  if...else statement, 59–60
  if statement, 58–59
console.log method, 23
constants
  naming conventions, 15–16
  overview, 14–15
  static, 194
const keyword, 13, 16
constructors, 49, 95, 102–104
  comparing classes and, 104–105
  defined, 101
content, element, 112
content delivery network (CDN), 210
control structures
  conditionals, 57–63
    chained if...else statement, 61–63
    complex conditions, 60–61
    if...else statement, 59–60
    if statement, 58–59
loops, 63–71
  for...in loop, 70–71
  for loop, 65–66
  for...of loop, 67–69
  while loop, 63–64
CSS (Cascading Style Sheets), 120–124
  flexbox layouts, 311–312
link element, 120
rulesets, 121
selectors
  overview, 121–124
  using in JavaScript, 124–125
styling bar graphs with, 292–295
styling SVG elements with, 264–266
D
D3 (D3.js) library, 255–297
  advanced joins, 276–279
  bar graphs, 279–297
    animating changes, 296–297
    calculating character frequencies, 280–282
    cleaning data, 295
    drawing, 282–291
    setting up, 279–280
    styling with CSS and regular expressions, 292–295
  data binding, 270–271
  data joins, 271–273
  real-time updates, 273–275
  selections, 269–270
  setup, 268
SVG graphics format, 256–268
  adding interactivity with JavaScript, 266–268
  defining paths, 261–264
  drawing circles, 260–261
  grouping elements, 258–259
  styling elements with CSS, 264–266
  transitions and key functions, 275–276
data attributes, 267
data binding, 268, 270–271
Data-Driven Documents. See D3 library
data visualizations
  D3 library, 255–297
    advanced joins, 276–279
    bar graphs, 279–297
    data binding, 270–271
    data joins, 271–273
    real-time updates, 273–275
    selections, 269–270
    setup, 268
  SVG graphics format, 256–268
    transitions and key functions, 275–276
GitHub Search API, 299–328
  adding interactivity, 319–324
  basic visualization, 304–309
complete code, 325–328
fetching data, 300–304
improving visualization, 310–319
setting up, 300
decay, ADSR, 211–212
decibels (dB), 212
declarations
  defined, 13
  in rulesets, 121
  separating from initialization, 178–179
declaring functions, 74–78
  parameter types, 77
  return values, 75–76
  side effects, 77–78
decrementing, 16–19
  addition and subtraction
    assignment, 18
  multiplication and division
    assignment, 18
decrement operator (--) , 17, 64
deleating events, 131–134
De Morgan’s law, 28
descendants, 114
descendant selectors, 123
destructuring assignment, 69
distanceTo method, 98–100
div element, 135, 148
division assignment operator (//=), 18
document object, 113
Document Object Model (DOM), 114–118
  DOM API, 115–116
  element identifiers, 116–118
document.querySelectorAll
  method, 124
document.querySelector method, 160–161
dollar sign ($), 292
domain, 284
DOM API, 115–116, 129
dot notation, 48–49
double equals (==) operator, 31
double quotation marks ("), 19
doubler function, 88
drawCircle function, 149–150
draw function, 163
drawing context, 143
draw method, 192
drum sounds, 224–232
  drum loop, 230–232
  hi-hat synthesis, 224–226
  kick synthesis, 228
  making drumbeat, 237–240
  reverb, 228–230
  snare synthesis, 226–227
dynamically typed programming
  language, 77
dynamic visualizations, 256
E
  eighth notes, 217
  elements
    for basic visualization, 304–305
    defined, 112
    game, 193–197
      ball, 195–197
      paddles, 194
  else if clause, 61–62
  em element, 128
  empty strings, 31–32
  encapsulation, 190–191
  enter, element, 277
  Entity superclass, 194
  equality, with coercion, 31–32
  error messages, 14–15
  escape sequences, 23–24
  evaluation, 10
  event-based programming, 127–139
    event handlers, 128–134
    event bubbling, 130
    event delegation, 131–134
    keyboard events, 137–138
    mouse movement events, 134–136
  exit, element, 277
  expressions, 10
  Extensible Markup Language
    (XML), 256
F
  false value, 32–33
  fetching data, 300–304
  fillRect method, 143, 161–162
  fillStyle property, 143
filtering data by license, 319–322
filter method, 86
find array method, 85–86
first-class citizens, 78
flat, 215
flexbox layouts, 311–312
floating-point numbers, 11–12
FM (frequency modulation), 240
FM synthesis, 240–241
focus, 137
for...in loop, 70–71
for loop, 65–66
for...of loop, 67–69
forward slashes (//), 165, 292
frequencies, character, 280–282
frequency modulation (FM), 240
frontend code, 303
function declarations, 74, 79–80
function expressions (function literal), 80–82
function keyword, 74, 80
functions, 73–91
arrow functions, 82–83
declaring and calling, 74–78
parameter types, 77
return values, 75–76
side effects, 77–78
defined, 5
function expressions, 80–82
higher-order functions, 85–90
array methods taking callbacks, 85–87
custom functions taking callbacks, 88–89
functions returning functions, 89–90
methods vs., 94
passing as arguments, 78–79
rest parameters, 84–85
fundamental, tone, 214

Game class, 198–201
game elements, 193–197
GameView class, 191–193
Scores and Computer classes, 197–198
starting game, 201
Pong, 159–202
bouncing, 165–166
complete code, 185–188
computer control, 180–182
drawing ball, 161–162
game loop, 163–165
game over, 182–184
overview, 159–160
paddles, 166–175
refactoring, 162–163
scoring points, 176–180
setup, 160–161
game loop, 163–165
game over, 182–184
GameView class, 191–193
g element, 258–259
getContext method, 143
getElementById method, 117
Git, 330
GitHub, 331
GitHub Search API, 299–328
adding interactivity, 319–324
animating changes, 322–324
filtering data by license, 319–322
basic visualization, 304–309
creating elements, 304–305
drawing axes, 305–307
drawing bars, 308–309
complete code, 325–328
fetching data, 300–304
improving visualization, 310–319
color-coding bars, 314–318
labeling left axis, 318–319
showing repository info, 310–314
setting up, 300
Glitch, 331
Google Chrome. See Chrome
greater than (>), 30
greater than or equal to (≥), 30
grouping elements, 258–259

G

gain nodes, 208
Game class, 198–201
Game creation projects
object-oriented Pong, 189–202
design, 190–191
file structure, 191
H

handlers, event, 128–134
  event bubbling, 130
  event delegation, 131–134
  song writing, 236–237
harmonics, 214
hash mark (#), 122–123, 215
hasOwnProperty method, 105
head element, 113, 119
hex colors (hexadecimal color syntax), 260–261
hidden licenses, 319–322
higher-order functions, 85–90
  array methods taking callbacks, 85–87
  custom functions taking callbacks, 88–89
  functions returning functions, 89–90
hi-hat synthesis, 224–226
:hover pseudo-class, 133
href attribute, 119
HTML (HyperText Markup Language), 111–125, 332
  creating HTML document, 112–113
CSS, 120–124
    Link element, 120
    rulesets, 121
    selectors, 121–124
defined, 5–6
DOM, 114–118
    DOM API, 115–116
    element identifiers, 116–118
    nested relationships, 114
    script elements, 118–119
    using CSS selectors in JavaScript, 124–125

I

id attribute, 116
identifiers, 13
ID selector, 122
if...else statement
    chained, 61–63
    keydown handler including, 138
    overview, 59–60
if statement, 58–59, 179–180
incrementing, 16–19
  addition and subtraction
    assignment, 18
  multiplication and division
    assignment, 18
increment operator (++), 17
indentation, 40, 113
indexes
    in arrays, 38–39
    finding index of element in array, 45
    in strings, 21–22
indexOf method, 45
infinite loops, 65
inheritance, 97–101
initialization, 13, 178–179
inline element, 122
innerText property, 117
input element, 148
instanceof keyword, 100
instances, creating, 94–97
interactivity, 319–324
  animating changes, 322–324
  with canvas elements, 147–151
  filtering data by license, 319–322
  visualizations, 256

J

JavaScript
  adding interactivity to SVG
    graphics format, 266–268
  bindings, 12–16
    constants, 14–15
    naming conventions, 15–16
    variables, 13–14
  Booleans, 26–30
    comparison operators, 29–30
  classes, 93–108
    creating instances and, 94–97
    inheritance, 97–101
    prototype-based inheritance, 101–108
  coercion, 30–35
    equality with, 31–32
    truthiness, 32–35
  compound data types, 37–56
    arrays, 38–47
JavaScript (continued)
compound data types (continued)
nesting objects and arrays, 52–56
objects, 47–52
conditionals, 57–63
escape sequences, 23–24
expressions and statements, 10
frameworks and libraries, 332
functions, 73–91
arrow functions, 82–83
declaring and calling, 74–78
function expressions, 80–82
higher-order functions, 85–90
passing as arguments, 78–79
rest parameters, 84–85
incrementing and decrementing, 16–19
addition and subtraction
assignment, 18
multiplication and division
assignment, 18
loops, 63–71
numbers and operators, 10–12
floating point, 11–12
order of operations, 11
strings, 19–23
finding length of, 20–21
getting character from, 21
getting multiple characters from, 21–22
joining, 20
trimming whitespace from, 22
template literals, 24–26
undefined and null, 26
using CSS selectors in, 124–125
writing code, 3–7
using JavaScript Console, 4–5
using text editor, 5–6
JavaScript clock, 220–221
JavaScript Console, 4–5
Jest, 333
join-by-index mode, 276
joining strings, 20
join method, 45–46, 271–273, 276–279
JSON (JavaScript Object Notation), 55
JSON API, 300
JSON.stringify method, 55–56
K
keyboard events, 137–138
keydown events, 137–138
key function, 275–276
keys, object
defined, 47
getting, 49–50
getting values and, 50–51
key-value pair, 47
keywords, 13
kick synthesis, 228
L
labeled axes, 288–291
label element, 148–149
left-aligned text, 177
length property, 20–21
less than (<), 30
less than or equal to (≤), 30
let keyword, 13, 16
licenses
filtering data by, 319–322
names, 315–318
li element, 131
link element, 120
listener, event, 129
literals
defined, 10
nesting with, 52–53
logical operators, 27–28
lookahead time, 221
looping variable, 65–66
loops, 63–71
for...in loop, 70–71
for loop, 65–66
for...of loop, 67–69
while loop, 63–64
M
makeAppender function, 90
map array method, 86–87
matching elements, 121
Math.hypot method, 98
MDN (Mozilla Developer Network)
Web Docs, 147, 263, 292
metadata, 113
methods
   for arrays, 41–47
      adding element to array, 42
      combining arrays, 44–45
      finding index of element in array, 45
      removing element from array, 43–44
      turning array into string, 45–46
   chaining, 130, 269
   defined, 21
   functions vs., 94
   overriding, 107–108
middle C, 215
monophonic synth, 216
monospace, 177
mouse movement events, 134–136, 169
move method, 95–97
moveTo method, 146
Mozilla Developer Network (MDN)
  Web Docs, 147, 263, 292
multidimensional arrays, 39
multiplication assignment operator (*=), 18
music projects, 205–252
   making drum sounds, 224–232
      drum loop, 230–232
      hi-hat synthesis, 224–226
      kick synthesis, 228
      reverb, 228–230
      snare synthesis, 226–227
   sampling, 232–234
   song writing, 235–252
      adding bass lines, 240–242
      adding chords, 242–244
      complete code, 247–252
      event handling, 236–237
      getting organized, 235–236
      making drumbeat, 237–240
      playing tunes, 244–246
Tone.js library, 210–217
   generating tone with, 210–211
   playing more notes in sequence, 215–216
   playing multiple notes at once, 216–217
Tone.Synth options, 211–215
Tone.js transport, 217–224
Tone.Loop, 218–220
Tone.Part, 223–224
Tone.Sequence, 221–222
Web Audio API, 206–209
   generating tone with, 207–209
   setting up, 206–207
mutations, 41

N
   named function expressions, 81
   naming conventions, 15–16
      classes, 95
      constructors, 102
   nested relationships, 114
   nesting, 52–56
      elements, 112
      exploring nested objects in console, 54–55
      with literals, 52–53
      printing nested objects with
         JSON.stringify, 55–56
      with variables, 53–54
   new keyword, 96
   nice method, 291
   Node.js, 303, 330
   NodeList, 124–125
   nodes, 206
   notes
      playing in sequence, 215–216
      playing multiple notes at once, 216–217
   not operator (!), 28
   null value, 26
   number literals, 10
   numbers, 10–12
      floating point, 11–12
      order of operations, 11

O
   Object.assign method, 51–52
   Object constructor, 105
   Object.entries method, 71
   Object.keys method, 49–50
   object literal, 47
   object literal shorthand syntax, 238
object-oriented Pong, 189–202
  design, 190–191
  file structure, 191
  Game class, 198–201
  game elements, 193–197
    ball, 195–197
    paddles, 194
  GameView class, 191–193
  Scores and Computer classes, 197–198
  starting game, 201
object-oriented programming, 93
Object.prototype, 105–106
objects, 47–52
  accessing values in, 48
  creating, 47–48
  nesting
    exploring nested objects in
      console, 54–55
    with literals, 52–53
    printing nested objects with
      JSON.stringify, 55–56
    with variables, 53–54
  setting values to, 49
  working with, 49–52
    combining objects, 51–52
    getting object’s keys, 49–50
    getting object’s keys and
      values, 50–51
offsetX and offsetY properties, 150
open source code owners, 304
open source repositories, 299
operands, 27
operators, 10–12
  defined, 10
    floating-point numbers, 11–12
    order of operations, 11
  optional chaining operator, 313
  order of operations, 11
  or operator (||), 27–28, 33–34
oscillators, 208
outlined rectangles, drawing, 144–145
overriding methods, 107–108
overtones, 214
P
  paddles, Pong, 166–175
    bouncing near ends of, 174–175
detecting collisions, 170–173
  elements, 194
  moving with player input, 169–170
parameters, 74
  arguments vs., 75
  functions, 77
  rest parameters, 84–85
parent elements, 114
parentheses (), 79
path element, 261–264
paths, drawing shapes using, 145–147
p element, 112
PEMDAS rule, 11
pentatonic scale, 219
percent (%) characters, 301
period (.), 123
pipe character (|), 241–242
placeholder syntax ({$}), 24–25
player input, moving paddles with,
  169–170
polymorphism, 191
polyphonic synth, 216–217
Pong, 159–202
  bouncing, 165–166
  complete code, 185–188
  computer control, 180–182
  drawing ball, 161–162
  game loop, 163–165
  game over, 182–184
object-oriented Pong, 189–202
  design, 190–191
  file structure, 191
  Game class, 198–201
  game elements, 193–197
  GameView class, 191–193
  Scores and Computer classes, 197–198
  starting game, 201
overview, 159–160
paddles, 166–175
  bouncing near ends of,
    174–175
detecting collisions, 170–173
  moving with player input,
    169–170
refactoring, 162–163
scoring points, 176–180
setup, 160–161
pop method, 43
postfix incrementing and
decrementing, 17
prefix incrementing and
decrementing, 17
prepended elements, 275
printing nested objects, 55–56
projects
data visualizations
D3 library, 255–297
GitHub Search API, 299–328
game creation
object-oriented Pong, 189–202
Pong, 159–188
making music
generating sound, 205–252
song writing, 235–252
properties, defined, 20
prototype-based inheritance, 101–108
comparing constructors and
classes, 104–105
exploring Object.prototype,
105–106
overriding method, 107–108
using constructors and prototypes,
102–104
walking the prototype chain,
106–107
[[Prototype]] property, 55, 103
prototypes, 101–104
pseudo-classes, 133, 266
push method, 42
Python, 334

Q
quadraticCurveTo method, 147
querySelectorAll method, 124
question mark (?), 301
queues, 43

R
range, 284
rate limiting, 78
React, 332
real-time updates, 273–275
receivers, 94
rectangles, drawing outlined, 144–145
refactoring, 162–163
regular expressions, 292–295
relationships, 97
relative position, 262–263
rel attribute, 119
release, ADSR, 211–212
repository info, 310–314
rest parameters, 84–85
rests, 222
return keyword, 89
return values
functions, 75–76
methods, 21
with side effects, 77–78
reverb, 228–230
RGBA color, 150
RGB color, 261
right-aligned text, 177
rotation, 258
rulesets, 121
Rust, 334–335

S
sampling, 232–234
Scalable Vector Graphics. See SVG
scaling
bars in D3, 284–288
SVG supporting, 258
scope, 67, 179
Scores class, 197–198
scoring points, 176–180
script elements, 6, 118–119
selections, 269–270
selectors
overview, 121–124
using in JavaScript, 124–125
self-documenting, 87
semicolon (;), 10
Set constructor, 315
setInterval function, 83
setTimeout function, 79, 81–82, 220
shadowing, 108
shapes, drawing using paths, 145–147
sharp, 215
shiftKey property, 267
shift method, 43–44
short-circuiting, 34
siblings, 114
sidebars, 310–314
side effects, 77–78
Simple Storage Service
(Amazon S3), 233
sine wave, 208
single quotation mark (’), 19
sixteenth notes, 217
skewing, 258
slice method, 21–22
slider, 148
snare synthesis, 226–227
song writing, 235–252
adding bass lines, 240–242
adding chords, 242–244
complete code, 247–252
event handling, 236–237
going get organized, 235–236
making drumbeat, 237–240
playing tunes, 244–246
span elements, 310–311
square brackets ([]), 21, 48–49
src attribute, 119
starting game, 201
state, 152
statements, 10
statically typed language, 77
static drawings, 142–147
outlined rectangles, 144–145
using paths, 145–147
static methods, 49, 198
static properties, 194
strict inequality (!=) operator, 32
str.includes(otherStr) method, 23
string literal, 19
strings, 19–23
finding length of, 20–21
getting character from, 21
getting multiple characters from,
21–22
joining, 20
trimming whitespace from, 22
turning array into, 45–46
strokeRect method, 144
strong elements, 122–123
str.padStart(num, char) method, 23
str.repeat(count) method, 23
str.toLowerCase() method, 23
subclasses, 97
subexpressions, 171
subtraction assignment operator ( -=), 18
superclass, 97
super keyword, 99
sustain, ADSR, 211–212
SVG (Scalable Vector Graphics)
graphics format, 256–268
adding interactivity with
JavaScript, 266–268
defining paths, 261–264
drawing circles, 260–261
grouping elements, 258–259
styling elements with CSS,
264–266
SvgPathEditor, 263
syntactic sugar, 101
synthesizer, 211

tags, 6, 112
template literals, 24–26
testing, 332–333
text editors, 5–6
this keyword, 95–97
3D graphics, 143
tic-tac-toe game, 39–41
timbre, 213
time, Web Audio API, 220–221
tone generation
with Tone.js library, 210–211
with Web Audio API, 207–209
Tone.js library, 210–217
generating tone with, 210–211
playing more notes in sequence,
215–216
playing multiple notes at once,
216–217
Tone.Synth options, 211–215
Tone.js transport, 217–224
Tone.Loop, 218–220
Tone.Part, 223–224
Tone.Sequence, 221–222
Tone.Sequence, 218–220
Tone.Part, 223–224
Tone.Sequence, 221–222
Tone.Synth options, 211–215
tooltips, 310
transformations, 258
transitions, 275–276, 296–297
translation, 258–259
transport, Tone.js. See Tone.js transport trim method, 22
triple equals (===) operator, 29
true constants, 15–16
true value, 32–33
trombone notes, 233
truthiness, coercion, 32–35
tune playing, 244–246
TypeScript language, 333–334

U
ul element, 131
unauthenticated APIs, 302–303
undefined behavior, 117
undefined value, 26
unshift method, 43
update, element, 277
update function, 153, 164
URL encoding system, 301
URLs, 300–301

V
valid identifiers, 48
values, object
accessing, 48
setting, 49
variables
naming conventions, 15–16
nesting with, 53–54
overview, 13–14
var keyword, 13
vector graphics, 256
visualizing data
D3 library, 255–297
advanced joins, 276–279
bar graphs, 279–297
data binding, 270–271
data joins, 271–273
real-time updates, 273–275
selections, 269–270
setup, 268
SVG graphics format, 256–268
transitions and key functions, 275–276
GitHub Search API, 299–328
adding interactivity, 319–324
basic visualization, 304–309
complete code, 325–328
fetching data, 300–304
improving visualization, 310–319
setting up, 300
VS Code (Visual Studio Code)
if statement in, 58–59
indentation in, 40
overview, 5–6
Vue.js, 332

W
W3C (World Wide Web Consortium), 206
walking the prototype chain, 106–107
Web Audio API, 206–209
generating tone with, 207–209
setting up, 206–207
web development, 332
while loop, 63–64
white noise, 224
whitespace, 22, 295
World Wide Web Consortium (W3C), 206

X
XML (Extensible Markup Language), 256

Z
zero-based indexing, 21, 38, 218