Index

Symbol and Numerics

&& operator, conditional rendering with, 288–289
+ (addition) operator, 83, 86–87
&& (And) operator, 87–88
* (asterisk), 228, 700
 (backslash), 70, 71
` (backtick character), 53, 70–71
: (colon), 375
+ (concatenation operator), 71
{} (curly braces), 78
$ (dollar sign), 685–686
-- (decrement) operator, 86–87
/ (division) operator, 83, 86–87
{{}} (double curly braces), 372
“ (double quote), 71
== (equality) operator, 85–86
** (exponentiation) operator, 86–87
> (greater than) operator, 85–86
>= (greater than or equal to) operator, 85–86
++ (increment) operator, 86–87, 97
!= (inequality) operator, 85–86
< (less than) operator, 85–86
<= (less than or equal to) operator, 85–86
* (multiplication) operator, 83, 86–87
! (negation operator), 302
! (Not) operator, 87–88
|| (Or) operator, 87–88
. (period), 79
% (remainder) operator, 86–87
? (question mark), 700
“ (quotation mark), 70
’ (single quote), 71
[] (square braces), 375
!== (strict inequality) operator, 85–86
=== (strict equality) operator, 85–86
- (subtraction) operators, 83, 86–87
. . . (three dots), 122–123
=== (triple equals operator), 132
`200` status code, 217
`301` status code, 217
`400` status code, 218
`401` status code, 218
`404` status code, 218
`500` status code, 218

A

`abort` event, 187
above the fold content, 238
absolute path, 622
abstraction, 168
acceptance testing, 547
access tokens
  about, 750–751
  using, 751–753
Access-Control-Allow-Methods HTTP headers, 634
Access-Control-Allow-Origin HTTP headers, 634
`activeElement` property, 250
`addEventListener()` method, 184, 186–190, 193–194, 199, 200–202, 256, 328, 413, 415
adding
  app server, 723–724
  elements to the DOM, 254–255
  global styles, 292–293
  local styles, 293
  salt to hashing, 738
  server modules, 723–724
  style, 509–510
  style to components, 387–393, 447–449
  workspaces, 669
addition (+) operator, 83, 86–87
`addJS()` method, 643
`addListener()` method, 649
`addNumbers()` function, 140
Advanced Research Projects Agency Network (ARPANET), 628

COPYRIGHTED MATERIAL

after() method, 256
afterUpdate() method, 473
alert() method, 37, 199, 248, 372–373
analyzing complete Express server, 707–712
And (&&) operator, 87–88
anonymous functions
  declaring, 150
  writing as arrow functions, 152
API server
  Express, 709
  testing, 711–712
APIs
  about, 563
  built-in browser, 242–243
  custom, 243
  selecting, 352–353
app() function, 189–190
append() method, 251, 254–255, 256
app.get() method, 703
application generator tool (Express), 715–720
application-level middleware, 703
apply() method, 157
app.post() method, 703
app.put() method, 703
apps
  adding app server, 723–724
  configuring, 349
  mounting multiple, 349–350
  Svelte, 427–430
app.use() method, 703, 717–718
Apress, 62
arguments
  about, 140
  dynamic, 375
  passing
    about, 145–148, 584
    to directives, 375
    to event handlers, 329–330
    functions as, 146–147
    to listeners, 642–644
    to transitions, 470
  using return value as, 149
arguments object, 146
arithmetic operators, 86
ARPANET (Advanced Research Projects Agency Network), 628
Array() constructor, 108
array literal notation, 109
array methods
  looping with, 117–121
  updates and, 442–445
Array.map() method, 147, 291
arrays
  about, 79, 105–107
  accessing elements, 110–111
  creating, 108–110
  deleting elements, 111–112
  destructuring, 122
  filtering, 121
  looping with array methods, 117–121
  mapping, 119–121
  modifying, 111
  programming with array methods, 112–116
  reducing, 118–119
  slicing, 116
  splicing, 116
  spreading, 122–123
  using v-for directive with, 382–383
arrow functions
  defining methods using, 332–333
  knowing limits of, 154
  simplifying, 152–153
  using as event handlers, 643–644
  writing anonymous functions as, 152
ASCII, 605–606
assignment operators, 85
assignments, triggering reactivity with, 442
associativity, 84
asterisk (*), 228, 700
async functions
  about, 206–210
  catching exceptions with, 659
  converting promise chains to, 208–209
  unblocking code with, 238

asynchronicity, of database operations, 693
asynchronous JavaScript
  about, 197–198
  callback function, 200–202
  events, 199–200
  promises, 202–210
  reading, 198–199
  using AJAX, 210–222
  writing, 197–222
asynchronous JavaScript and XML (AJAX)
  about, 210–211
  calling other Response methods, 213–214
  fetch init object, 215–216
  getting data with Fetch API, 211–213
  getting JSON data, 222
  handling fetch() errors, 214–215
  Hypertext Transfer Protocol (HTTP), 216–218
  making requests with CORS, 218–220
  sending JSON data, 222
  working with JSON data, 220–222
asynchronous requests, waiting for, 479–481
attaching modifiers, to event listeners, 464–465
attributes
  different in JSX, 284–285
  event, 328
  style, 293–294
attributes property, 255
AudioTrack interface, 242
authentication. *See* registration and authentication
automated build toolchain, 514
automating build script, 513–522
#await block, 479–481

B

b (backspace), 71
Babel, transpilation with, 283
backpressure, 598–599
backslash (\), 70, 71
backspace (b), 71
backtick character (`), 53, 70–71
bang (converting) operator, 76
base classes, 169–170
Base64, 607, 750
bcrypt library, 738–739
before() method, 256
beforeUpdate() method, 473
Berners-Lee, Tim, 383
bidirectional connection, 667
bigInt data type, 74–75
:bind directive, creating two-way bindings with, 466–468
bind() method, 158, 161–162, 308, 331
binding
  data to templates, 398–400
  expressions, 372
  forms to events and data, 419–420
  Node.js, 563, 564
  select inputs, 467–468
Blink, 235
blob() method, 213–214
block comments, 51–52
block-scoped variables, 80
blur() method, 248
body property, 250
body-parser, 706
bookLoad(), 210
Boole, George, 75
Boolean data type, 75–76
Boolean() function, 141
Bootstrap, 292
bootstrapping, with vue-create, 344–346
break statement, 102–103
breakpoints
  defined, 660
  setting, 544–546, 669–671
browser chrome, 234
browser engine, 235
browser windows, running code in, 35–40
Buffer.alloc() method, 607–609, 610
Buffer.compare() method, 610
Buffer.concat() method, 610
Buffer.from() method, 609, 610
Buffer.isBuffer() method, 610

buffers
about, 603–604
content, 605–606
creating, 607–609
decoding, 606–607
encoding compared with decoding, 604
iterating over, 611
methods, 609–611
build script, automating, 513–522
built-in browser APIs, 242–243
built-in components, in JSX, 284–285
built-in directives, 373–375
built-in elements, 451–452
built-in events, triggering, 193–194
built-in matchers (Jest), 551
built-in middleware, 704–705
built-in objects, methods of, 142
byte, 605

C

`call()` method, 157
callback functions, 117–118, 146, 200–202
callback pattern, Node.js, 584–586
callbacks, opening files with, 614–616
calling
functions, 117, 140, 372
`Response` methods, 213–214
camelCase
in JavaScript, 32
in JSX, 284
`Canvas API`, 243
`capture` option, 188
carriage return (r), 71
case-sensitivity, of JavaScript, 31
`Cat()` function, 128
`catch()` function, 203, 657–658
catching
exceptions
with `async` functions, 659
with promises, 656–659
Promise rejection, 657–658
Chacon, Scott, 62
chaining functions, 121, 165–166
chaining streams, 600–602
Chakra, 237
`change` event, 187
changing
array lengths, 111
reactive data, 400–403
`charAt` function, 72
Cheat Sheet (website), 6
child, passing functions to, 159–165
`children` property, 250, 255
children props, composition using, 323–324
Chrome, debugging in, 542–546
Chrome DevTools, debugging in, 667–672
chunking, 588
class fields syntax, 175
`class` keyword, 169–170
classes
about, 167–168
abstraction, 168
base, 169–170
components of, 307–308
constructors, 171–173
creating
methods in, 173–174
objects with, 128–129
defining event handlers in, 330–332
derived, 170–171
encapsulation, 168
inheritance, 169
methods, 173–177
polymorphism, 169
practicing with, 177–179
properties, 173–177
`classList` property, 255
`clearInterval()` method, 248
`clearTimeout()` method, 248
`click` event, 187
client, 16–17
`close()` method, 248
closing files, using Promises, 616–617
`cls` command, 687

code
documenting, 51–54
editing in Sources panel, 544
running
in browser windows, 35–40
on command line, 570
in console, 33–34
unblocking with async and defer, 238
writing through testing, 552
code listing
adding styles, 509–510
App.js with map rendering, 528–529
assigning arrays to trigger reactivity, 444–445
asynchronous JavaScript user interface, 200
automatically refreshing stock price, 478–479
binding
attributes to data, 398–399
class methods before passing them as attributes, 330–331
inputs to data, 466–467
methods in constructor, 331–332
select inputs, 467–468
bookstore and shoppingCart, 160–161
breaking index.js into modules, 506
calculating dog's age in human years, 549
calling shoppingCart.displayCart() from bookstore, 162
catching
errors with fetch(), 214
exceptions with async/await, 659
Promise rejections, 658
thrown exceptions, 656
changing maximum number of listeners, 646–647
class component, 307
clearing intervals in componentWillUnmount(), 316
commenting templates, 453–454
compiled JSX, 282–283
components
created using Composition API, 352
created using Options API, 351
with a <slot>, 384
composing ErrorMessage components, 322–323
conditional rendering
with &&, 288–289
with conditional operator, 290
with element variables, 287–288
of elements, 455
inside in expressions, 455–456
conditionally defining functions using expressions, 151
configuring
Babel presets, 524
Webpack to use babel-loader, 523
connecting
to MongoDB in app.js, 732
to MongoDB server, 690
constructor for vehicles, 135
controlling styles programmatically, 392
converting buffers to strings, 618
copying
files using writable streams, 596
files with readFile() and writeFile(), 620
counter/svelte, 430
counting with v-for, 380
creating
and accessing ref objects, 403
and appending element nodes, 254
App.js, 525
classes with public, private, and static fields, 178–179
components that use props.children, 323
conflicting styles, 388
context, 490
database configuration object, 690
duplex streams with pipe(), 598
global styles within scoped styles, 448–449
JavaScript modules, 577
lists from arrays, 291
lists from arrays of objects, 456–457
methods in classes, 173
new post route, 752
new posts, 753
numbered lists from arguments, 146

code listing *(continued)*
objects using `Object.create()`, 130
persistent state, 318–319
promise chain, 202
reactive state objects, 400
reactive state with `<script setup>`, 401
readable stores, 484
static members, 176
streaming counters, 593
style modules, 295
transform streams, 600
uncontrolled inputs, 338
and using CSS modules, 390–391
writable stores, 484
CSS Modules file, 296
default `main.js`, 348
default vue-create `index.html` file, 347
defining
custom directives in `directives` object, 377
event handler functions using arrow function syntax, 333
functions using function expressions, 151
multiple props, 441
named event handler functions, 329
named slots, 461
props, 440
props using `defineProps()`, 396–397
props without assigning them to objects, 397
and using props with `setup()`, 397–398
`v-play` directive, 376
demonstrating JavaScript expressions in templates, 372
deriving class from base class, 172
`detectEdgeCollision` function, 533
detecting
collisions, 532
keyboard combinations, 418
displaying
bookstore and cart, 163–164
readable store's value, 489
enabling Reactivity Transform, 404–405
`ErrorButton` component, 367
Express API server, 707–708
extending
classes to create specific classes, 321
`EventEmitter`, 641
`React.PureComponent`, 314
fetching
data inside `componentDidMount()`, 312–313
lists using `onMounted()` lifecycle method, 364
files
containing modules, 224
containing named exports and default export, 226
that export several modules, 576
finding out whether items are directories, 624
finished `server.js` file, 724
finished `userModel` module, 734
firing built-in events, 192–193
flashlight component, 302
forwarding events, 465
function component example, 305
function components with timer, 319–320
functions
that takes arrays, 147
that throws errors, 655
for validating tokens, 752
without context, 156
Gamelet.js with comments, 51–52
generated HTML, 715
generated HTML templates, 504
`generateMap()` function, 508
generating HTML lists from arrays, 120–121
getting
absolute path to files, 622
data `onMount()`, 476–477
file information, 624–625
and using `unsubscribed()` function, 486
values of uncontrolled inputs from `Window` object, 338–339
handling
errors, 367–368
forwarded events, 466
Hello Express, 698

high-level view of Bing.com, 299
HTML contact form for testing application server, 662
HTML document, 39
HTML page, 48–49
importing
  core modules using ECMAScript modules, 566
  modules, 576
  modules into HTML files, 230
  style modules, 295
  and using components, 370–371
  and using core modules, 565
  and using CSS Modules, 296
`index.html` file for Svelte app, 427
`index.js` Route file, 719
`index.pug` template, 719
inserting documents into collections, 692
iterating over buffers, 611
It's Alive!, 630
JavaScript files, 38, 505
JavaScript Name Creator program, 30
JSON data, 221
JSON document, 675–676
`layout.pug` template, 720
listening for
  chunks, 591
  and handling click events, 161
  key events, 464
logging
  before and after updates, 473
  and displaying squares of numbers, 474
  references to tacos with events, 640–641
  `req.params`, 701
  requests, 649
  requests with middleware, 702
  in users, 743
looping over arrays, 382–383
`main.js` file for Svelte app, 428
method handlers receiving event objects, 415–416
mounting multiple apps, 350
`moveBall()` function, 507
named function module, 225
naming slots, 386
nested callbacks, 201
nested `setTimeout` calls, 207
Node.js web server, 568–569
objects with nested properties, 130–131
`obstacle`'s component, 526
overriding methods in classes, 174
`package.json` with ECMAScript modules enabled, 580
passing
  arguments to listeners with `emit()`, 642
  arguments using inner functions, 330
  content to slots, 460
  key attributes, 457–458
  multiple props, 441
  objects into function, 143
  objects to functions, 156–157
  props, 301, 441
  three aliases and objects to `v-for`, 382
  by value, 143
posts API requests, 729
Posts model, 734
printing properties of objects, 131
Pug template, 714
pushing data into streams, 592
putting together app components, 267–268
random movie picker app, 153
random number guessing game, 101
random number guessing game with `do...while`, 102
reading
  directory files and logging filenames, 623
  files asynchronously and logging it, 617
  from files at higher level in file structure, 621
  files synchronously, 619
README file for Gamelet, 54
registering 11 event listeners, 645
removing event listeners, 647, 648
rendering
  lists of posts, 434
  the root component, 524
  with state data, 311

code listing *(continued)*
replacing promise chains with async functions, 209
request for testing user registration, 740
responding to requests with requests, 631
rewriting `getSquare` using async/await, 208
root component, 428–429
`routes/user.js` file with login route, 747–749
routing with JavaScript, 258–259
sample `.env` file, 582
sample HTML page, 496
sample JSON arrays of objects, 290–291
`scripts` object with complete build script, 521, 522
seeing `Error` properties, 653
selecting
elements with `querySelectorAll()` method, 253
with `getElementById()` method, 252
server without streaming, 588–589
serving
buffers as images, 606–607
static HTML files from Node.js, 713
setting
application `config` options to specify global error handler, 349
event listeners, 188–189
global properties, 349
state in constructor, 310
up Express, 724
sign-up route, 739
simple components with pure HTML templates, 370
simple counter component written in Svelte, 439
simple React component, 282
simple streaming server, 590
simple web application for learning about debugging, 660–661
single-file component using `setup` attribute, 360–361
single-file component's script element, 358
single-file Todo app, 359
skeleton
for `routes/posts.js`, 726
for `routes/user.js`, 727
specifying
attribute values using expressions, 286
fallback content for slots, 386
starter template for single-file component, 358
starting
`package.json` file, 499
timers on mounts, 472–473
static navigation bar mockup, 266
static search box mockup, 267
streaming
in flowing mode, 594–595
in paused mode, 595
subscribing to stores, 485
supporting both callbacks and promises, 586
synchronous JavaScript user interface, 199
synchronous program, 198
targeting
content to named slots, 460
named slots, 386–387
testing
for collisions with obstacles, 511
dog-age calculator, 549
for edge collisions, 510
map component, 554
text expressions escape HTML, 458
unhandled promise rejection, 656–657
updating data with 1-way binding, 419
`useGitHubRepos` custom hook, 324–325
user API requests, 728
Users Route file, 718–719
using
$ syntax, 488
array methods, 443
arrow functions as event handlers, 643
`#await` block, 480–481
callback functions, 147
class, 129
computed properties, 406–407
constructor function, 128
controlled inputs, 336–337

custom elements, 452–453
custom hooks, 325
different transitions for entering and leaving, 470
dynamic loading, 229
element variables, 287
environment variables, 583
error-handling middleware, 704
event bubbling to detect events on multiple elements, 191
`express.static()`, 707
`finally()` block, 658–659
`for...in` to display properties of objects, 99
`fs.open()` and `fs.close()`, 616
`fs.open()` and `fs.close()` with Promises, 616–617
`getContext()`, 490–491
global and scoped styles, 389–390
@html to render HTML in text expressions, 459
inline event handlers, 412
inline JavaScript, 36–37
JavaScript modules, 578
method handlers, 413
middleware to modify `res` objects, 703
multiple conditions with &&, 289
`PassThrough` stream, 599
promises to load data with `XMLHttpRequest()`, 203–204
props to create more reusable components, 322
reactive context, 491–492
reactive state objects, 401
reactive statements, 445–446
Reactivity Transform, 405
`React.memo()` top optimize rendering of function components, 314
`refs` inside templates, 403–404
`removeBook` as event handler, 162
router-level middleware, 703
routes in `app.js`, 728
same function in multiple objects, 159
scoped CSS, 388–389
shorthand version of `v-bind`, 399
slots to render children, 460
`start()` and `stop()` with stores, 489
state in class components, 308
state in function components, 306
stores to provide and update preferences, 486–487
style attribute, 294
`switch` statement, 94
template literal string, 71
this `in class` constructor, 172
`this` in event handlers, 642–643
`tick()` to wait for state change to be applied, 474–475
transform involving user input, 601–602
transform streams, 600
`updateOne()` with upsert option, 695–696
`useEffect` to perform side effects, 317
`v-for` with objects, 381
`v-if`, `v-else-if`, and `v-else`, 379
`v-model` directive, 420
viewing properties of `SyntheticBaseEvent` object, 329
`VolumeSlider` written using function components, 334
watching values to trigger side effects, 409
wrapping
components around other components, 324
statements and function calls in arrow functions, 416
writing log messages to files, 597
coding, with Git, 55–62
collections
creating, 682
inserting documents into, 691–692
collisions
detecting, 531–534
testing for, 510–511
colon (:), 375
combining operators, 89
command line
running code on, 570
setting environment variables from, 582
command-line debugger, 663–667

commands
  read-evaluate-print loop (REPL), 573–575
  rerunning in console, 34–35
comma-separated value (CSV) files, 109–110
comments, documenting Svelte with, 453–454
CommonJS module, 575–576
compare() method, 610
comparing objects, 132–134
comparison operators, 85–86
compilation, 11–12
complex data type, 78–79
complex queries, using operators, 685–686
component hierarchy, designing, 265–266
component lifecycle
  about, 309–316
  fetching data in Svelte, 476–481
  following, 362–366
  Svelte lifecycle, 471–476
componentDidMount() method, 309, 312–313, 316–317, 318
componentDidUpdate() method, 313, 315
components
  adding style to, 387–393, 447–449
  building with, 437–449
  error handling in, 366–368
  identifying contents of, 439–447
  naming, 362
  Vue, 357–368
  writing, 437–439
componentWillUnmount() method, 315–316, 319–320, 365–366
composing, with slots, 383–387, 459–461
composition
  of components, 321–325
  with custom hooks, 324–325
  using children props, 323–324
  using explicit props, 322–323
Composition API, 352
compression, 706
computed() function, 406
computing properties, 405–408
concat() method, 72, 112
concatenation operator (+), 71
concatenation operators, 86–87
conditional code, creating with return, 149–150
conditional operators, conditional rendering with, 289–290
conditional rendering
  about, 88
  with JavaScript, 378
  JSX, 286–290
  using directives, 378–379
configuring
  apps, 349
  dev server, 515–518
  directories, 722–723
  Git, 57–60
  Jest, 547–548
  Visual Studio Code (VS Code), 43–50
  Webpack for React, 522–524
connecting
  to databases, 731–733
  with interfaces, 242
  to MongoDB server, 689–691
Connection HTTP headers, 634
console
  rerunning commands in, 34–35
  running code in, 33–34
console property, 248
console.log() method, 593
const, creating constants with, 66–68
constants, creating with const, 66–68
constructing store, 483–490
constructor function, 108, 128
constructor() method, 309–310
constructors, 171–173
content, buffer, 605–606
Content-Length HTTP headers, 634
content-type HTTP headers, 634
context
  functions and, 155–165
  setting of functions, 157–158
continue statement, 102–103
controlled inputs

defined, 335
using, 335–337
converting
to Boolean, 75–76
nested callbacks to async functions, 207–208
promise chains to async functions, 208–209
to React, 522–534
to streams, 589–590
between strings and numbers, 74
UI to React, 524–529
`cookie-parser`, 706
copying
JavaScript code, 15–22
objects, 132–134
static assets, 519–520
core modules, 564–566
`cors`, 706
CORS, making requests with, 218–220
`createApp()` function, 348, 349
`createAttribute()` method, 251
`createComment()` method, 251
`createElement()` method, 251, 254–255
`createEvent()` method, 251
`createInterface()` method, 641
`createReadStream()` method, 593–594
`createRoot()` method, 278–279
`createServer()` method, 630–631
`createTextNode()` method, 251
`createUser()` method, 687–688
creating
about, 495
arrays, 108–110
basic routes, 725–728
buffers, 607–609
collections, 682
with components, 437–449
conditional code with return, 149–150
constants with `const`, 66–68
creating modules, 506–512
custom events, 194, 640–641
databases, 681–682
directories, 722–723
duplex streams, 598–599
elements to the DOM, 254–255
expressions, 82
forms with React, 324–339
IDs, 682–683
instances, 170
lists, 290–291
loops, 97–103, 456–458
managing dependencies with npm, 497–503
methods in classes, 173–174
modules, 506–512
Node.js modules, 575–580
objects
about, 127–130
with class, 128–129
with `Object.create()` method, 130
using constructor functions, 128
using literal notation, 127
options request, 635
projects in Visual Studio Code (VSC), 25–27
React components, 297–325
React UI, 265
reactive data, 432–434
reactive statements, 445–446
readable stores, 484
readable streams, 592–595
reasons for tools, 496–497
requests with CORS, 218–220
scaffolding in Svelte, 425–426
schema with Mongoose, 730–734
source maps, 542–544
static version in React, 266–268
strings with template literal notation, 70–71
style modules, 295
transition animations, 468–469
two-way bindings
with `:bind` directive, 466–468
with `v-model`, 420
uncaught Promise rejection, 656–657
unidirectional transitions, 470
user model, 733–734
variables with `let`, 63–66

creating *(continued)*
web servers, 630–638
writable stores, 484
writable streams, 596–597
writing
dev script, 505–506
files, 503–505
cross-site scripting (XSS), 218, 745–746
CSS
global, 387–388
modules, 390–391
scoped, 388–389
`v-bind` directive in, 391–393
CSS Modules file, 296
CSV (comma-separated value) files, 109–110
curly braces ({}), 78
cursor objects, 693–694
custom APIs, 243
custom directives, 376–377
custom elements, 452–453
custom events, creating, 640–641
custom hooks, composition with, 324–325

D

dashes, in JavaScript, 32
data
about, 63
arrays, 79
binding
forms to events and, 419–420
to templates, 398–400
complex data type, 78–79
creating
constants with `const`, 66–68
variables with `let`, 63–66
deleting, 696
fetching in Svelte, 476–481
getting
about, 222, 693–694
with Fetch API, 211–213
to node modules, 580–584
objects, 78–79
reactivity and, 395–410
refreshing, 477–479
returning, 149–150
sending, 222
types of, 68–77
updating, 695–696
using in React, 327–339
variable scope, 80
data flow, implementing inverse, 269–270
`data()` method, 350
data storage, networking and, 239–240
data types
in NoSQL databases, 676–677
props, 300–301
recognizing, 300–304
in relational databases, 676
databases
about, 673
connecting to, 731–733
creating, 681–682
MongoDB, 674–681, 688–696
Mongosh, 681–688
Date HTTP headers, 634
`db` command, 686
`db.collectiondeleteMany()` command, 687
`db.collection.deleteOne()` command, 687
`db.collection.drop()` command, 687
`db.collection.find()` command, 687
`db.collection.find()` method, 684
`db.collection.insertMany()` command, 687
`db.collection.insertOne()` command, 686
`db.collection.updateMany()` command, 687
`db.collection.updateOne()` command, 687
`db.dropDatabase()` command, 687
`dblclick` event, 187
debugging. *See* error handling and debugging
declaring
anonymous function, 150
variables, 64
`decodeURI()` function, 141
`decodeURIComponent()` function, 141

decoding
  buffers, 606–607
  encoding compared with, 604
decrement (--) operator, 86–87
deep copy, 123
default exports, 225–226
default parameters, setting, 147–148
defaultMaxListeners, finding value of, 644–645
defer, unblocking code with, 238
defineProps() function, 396–397
defining
  event handlers in class components, 330–332
  function expressions, 150–152
  functions, 140
  methods using arrow syntax, 332–333
  modules, 224
  props
    about, 440
    with <script setup>, 396–397
    with setup() function, 397–398
DELETE method, 217
delete operator, 112
deleteMany() method, 696
deleteOne() method, 696
deleting
  array elements, 111–112
  data, 696
  object properties, 138
dependencies
  about, 497
  installing, 709
  managing with npm, 497–503
  npm, 501
deployment, 518–519
derived classes
  about, 170–171
  overriding methods in, 174
describe() function, 549, 550
designing
  component hierarchy, 265–266
  elements, 298–299
  templates, 451–461
destructuring arrays, 122
detecting collisions, 531–534
determining location, of states, 269
dev scripts, writing, 505–506
dev server, configuring, 515–518
developers, Node.js for, 561–562
development environment, starting, 22–29
devicePixelRatio property, 248
directives
  built-in, 373–375
  conditional rendering using, 378–379
  creating two-way binding with :bind directive, 466–468
  custom, 376–377
  dynamic arguments, 375
  listening for events with on: directive, 463–466
  modifiers, 375–376
  passing arguments to, 375
  shorthand names, 375
  using transition animations, 468–470
directories
  configuring, 722–723
  creating, 722–723
  finding, 624
  getting information, 623–625
  listing files in, 623
_dirname property, 594
disk space, troubleshooting, 50
dispatchEvent() method, 192, 194, 256
dispatching
  events programmatically, 192–194
  synthetic events, 328–329
displayBookstore() method, 163–164
displayCart() method, 161
displayWeatherForecast() function, 149
dist folder, 520–522
division (/) operator, 83, 86–87
Document interface
  about, 242, 249
  adding elements to the DOM, 254–255
  creating elements to the DOM, 254–255
  element methods, 256–257

Document interface *(continued)*
element nodes, 255–256
family tree, 249
methods, 250–251
properties, 249–250
selecting
element nodes, 251
with getElementById() method, 252
using selectors, 252–253
visualizing, 249
documenting
code, 51–54
Svelte with comments, 453–454
documents
finding, 684–686
inserting into collections, 691–692
listing, 682–683
dollar sign ($), 685–686
doLogin() function, 200–202
DOM
adding elements to the, 254–255
creating elements in the, 254–255
updating, 354–355
doMath() function, 118
doProcessing() function, 208
dot notation, 79, 130–131
double curly braces ({{}}), 372
double quote (“), 71
do...while loops, 101–102
drink() method, 173–174
drive() method, 155, 156, 157
duplex streams, creating, 598–599
dynamic arguments, 375
dynamic modules, loading, 229
dynamically typed language, 68

E

ECMA International, 14–15
ECMAScript, 14–15, 575
editing code, in Sources panel, 544
effects
performing on unmounting, 319–320
running, 317–319
Eich, Brendan, 14
element methods, Document interface, 256–257
element nodes
Document interface, 255–256
selecting, 251
element variables, conditional rendering with, 287–288
elements, designing, 298–299
embedded script, 237
emit() method, 640
emitting once, 649
enabling reactivity, 304
encapsulation, 168
encodeURI() function, 141
encodeURIComponent() function, 141
encoding, decoding compared with, 604
encoding argument, 620
end() method, 636, 702
end-to-end testing, 547
engines
browser, 235
JavaScript, 237
rendering, 235–238
template, 713–714
.env, setting environment variables with, 582–583
environment variables, 581–583
equality (==) operator, 85–86
equals() method, 610
error handling and debugging
in Chrome, 542–546
in components, 366–368
debugging Node.js programs, 660–672
error types, 651–652
exceptions *versus* errors, 654
handling exceptions, 655–659
Node.js’s Error object, 653–654
Error object (Node.js), 653–654
error-handling middleware, 703–704

ErrorMessage component, 322–323
errors
  exceptions compared with, 654
  handling with async/await, 210
  linting, 537–540
  troubleshooting, 50
  types of, 651–652
error.stack, reading, 654
ES modules, 577–579
ESbuild, 515
escaping characters, 70
ESLint
  installing, 536–537
  integrating into build script, 540–541
  running, 537
eval() function, 141
event attributes, 185, 328
event handlers
  about, 435–436
  basic, 464
  defining in class components, 330–332
  passing
    arguments to, 329–330
    functions from function components, 333–334
  properties of, 185–186
  in React, 328–334
  using arrow functions as, 643–644
event listeners
  attaching modifiers to, 464–465
  removing, 194
event modifiers
  about, 416–417
  detecting combinations, 419
  key modifiers, 417–418
event object, 190
event property, 248
event targets, selecting, 186
EventEmitter
  about, 640
  creating custom events, 640–641
  extending, 641
  passing arguments to listeners, 642–644
event.preventDefault(), 416–417
events
  about, 181, 639
  asynchronous JavaScript and, 199–200
  binding forms to data and, 419–420
  dispatching programmatically, 192–194
  emitting once, 649
  event loop, 183–184
  EventEmitter class, 640–644
  forwarding, 465–466
  handling multiple, 466
  JavaScript runtime model, 182–183
  listening for, 184–195
  listening for with on: directive, 463–466
  maxListeners, 644–647
  removing listeners, 647–648
  responding to, 411–420
  setting listeners with v-on, 411–416
  Synthetic, 328–329
  using event modifiers, 416–419
  using in React, 327–339
EventTarget interface, 242
every() method, 112
exceeding maximum listeners for emitters, 645–646
exceptions
  catching
    with async functions, 659
    with promises, 656–659
  errors compared with, 654
  handling, 655–659
excluding files, from linting, 541
exit command, 687
expect() function, 549, 551
expectations, 551
expiration, of tokens, 744–745
exponentiation (**) operator, 86–87
exporting
  modules, 224–226
  props, 440–442
  renaming exports, 228

Express
  analyzing complete server, 707–712
  installing, 697–698
  installing dependencies, 709
  installing Express API server, 709
  middleware, 701–707
  server-side routing with, 699–701
  serving views, 712–715
  using application generator tool, 715–720
express() method, 698, 703, 704–705
expressions
  about, 30, 81
  building, 82
  class, 170
  in JSX, 285–286
  text, 458–459
  watch, 546
express.Router() method, 725–728
express.static() method, 706–707
extending EventEmitter, 641

F

fallback content, specifying, 385–386
fat arrow syntax, 119
Fetch API
  about, 242
  getting data with, 211–213
fetch() errors
  about, 218
  handling, 214–215
fetch init object, 215–216
fetch() method, 211, 215–216, 222, 407, 655
fetching data, in Svelte, 476–481
fields
  defining, 174–177
  returning, 684
File API, 243
file descriptor, 615
file system
  getting file and directory information, 623–625
  importing fs module, 613–614
  reading files, 614–619
  using paths, 621–623
  writing files, 619–620
files
  closing using Promises, 616–617
  excluding from linting, 541
  getting information, 623–625
  listing in directories, 623
  opening using Promises, 616–617
  reading, 614–619
  writing, 503–505, 619–620
  writing to using Promises, 616–617
filter() method, 112, 121
filter object, 684
filtering arrays, 121
finally() block
  about, 210
  using, 658–659
find() method, 684, 685, 694
finding
  directories, 624
  documents, 684–686
  value of defaultMaxListeners, 644–645
findOne() method, 694
finishing login route, 746–753
fixing linting errors, 537–540
flow
  about, 91
  choosing a path, 91–96
  creating loops, 97–103
flowing mode, 594–595
focus() method, 194, 248
following component lifecycle, 362–366
for loops, 97–98
forEach() method, 112, 693–694
for...in loops, 98–99
form feed (f), 71
forms
  binding to events and data, 419–420
  creating with React, 324–339
forms property, 250
for...of loops, 100

forwarding events, 465–466
fs module
  importing, 613–614
  reading readable streams from, 593–594
fs.open() method, opening files with, 614–616
fs.read() method, reading from files with, 614–617
fs.readdir() method, 623
fs.readFile() method, 671–672
fs.stat() method, 624–625
fs.write() method, writing to disks with, 619–620
fs.writeFile() method, 620
fullScreen property, 248
function bodies, writing, 148
function calls, stacking, 182
function components
  about, 304–306
  passing event handler functions from, 333–334
  using lifecycle in, 316–320
function definition, 140
function expressions, defining, 150–152
function keyword, 332–333
functions
  about, 139–141
  addNumbers(), 140
  anonymous, 150, 152
  app(), 189–190
  async, 206–210
  Boolean(), 141
  callback, 117–118, 146, 200–202
  calling, 117, 140, 372
  Cat(), 128
  catch(), 203, 657–658
  chaining, 165–166
  charAt, 72
  computed(), 406
  context and, 155–165
  createApp(), 348, 349
  declaration scope and hoisting, 150
  declaring anonymous, 150
  decodeURI(), 141
  decodeURIComponent(), 141
  defineProps(), 396–397
defining function expressions, 150–152
describe(), 549, 550
displayWeatherForecast(), 149
doLogin(), 200–202
doMath(), 118
doProcessing(), 208
encodeURI(), 141
encodeURIComponent(), 141
eval(), 141
expect(), 549, 551
generateMap(), 508–509
getContext(), 490–491
getData(), 655–656
getLocalWeather(), 145, 148, 149
getStateInfo(), 226, 227–228
getUserById(), 657
h(), 354–355
handleClick(), 188
handleEvent(), 648
import(), 229
isFinite(), 141
isNaN(), 141
it(), 550–551
knowing limits of arrow functions, 154
loadBook(), 208
mount(), 348, 349
moveBall(), 507–508
naming, 144–145
number, 73
Number(), 141
onMount(), 472–473, 476–477, 478
onMouseMove(), 190
parseFloat(), 73, 86, 141
parseInt(), 73, 86, 141, 142
passing
  arguments, 145–148
  to child, 159–165
  between objects, 158–159
  objects to, 156–157
  from one objects to another, 158–159
  as props, 330–334

functions *(continued)*
by reference, 143
by value, 142–143
passive, 188
pickAMovie(), 153
postData(), 414
reactive(), 360, 391–393, 400–403
ref(), 391–393, 400–404
removeBook(), 161–162
replace, 72
res.render(), 719
returning data, 149–150
reverseOrder(), 253
router.get(), 719
setting context of, 157–158
setup(), 358–361, 377, 396, 397–398, 401
showLogInScreen(), 88
signal, 188
simplifying arrow functions, 152–153
split(), 72, 109–110, 142
squareIt(), 117–118
start(), 488–490
as stateless, 305
stop(), 488–490
string, 71–72
String(), 139–140, 141
substring, 72
super(), 171, 172, 175, 176, 307
test(), 549, 550–551
then(), 203
toLowerCase, 72
toUpperCase, 73
trim(), 142
uneval(), 141
using
methods of built-in objects, 142
top-level, 141
watch(), 408–410
window.alert(), 414
writing
about, 144–154
anonymous functions as arrow functions, 152
function bodies, 148
methods, 154–155
function-scoped variables, 80

G

Gecko, 235
generateMap() function, 508–509
generating tokens, 744–746
geolocation property, 245
get() method, 698, 699, 701
GET method, 17, 217
getAttribute() method, 256
getContext() function, 490–491
getData() function, 655–656
getDerivedStateFromProps() method, 309, 310, 313
getElementById() method, 186, 251, 252
getElementsByClassName() method, 251, 256, 511
getElementsByTagName() method, 251, 256
getLocalWeather() function, 145, 148, 149
getMacListeners() method, 645
getProducts() method, 168
getSnapshotBeforeUpdate() method, 313, 314–315
getStateInfo() function, 226, 227–228
getUserApproval() function, 602
getUserById() function, 657
Git, coding with, 55–62
GitHub, 62
.gitignore file, 276
global CSS, 387–388
global styles, adding, 292–293
globals, using in templates, 373
global-scoped variables, 80
Google Chrome, 15
graphing operator, 84
greater than (>) operator, 85–86
greater than or equal to (>=) operator, 85–86
Grunt, 518–519
Gulp, 518–519

H

h() function, 354–355
handleClick() function, 188
handleEvent() function, 648
handler functions, specifying, 329
handling
  authentication, 741–743
  dependencies with npm, 497–503
  errors in components, 366–368
  errors with async/await, 210
  exceptions, 655–659
  fetch() errors, 214–215
  multiple events, 466
hanging request, 702
hasAttribute() method, 256
hashes, 736–737
hashing
  adding salt to, 738
  saving and, 738–739
head property, 250
heaping objects, 183
helmet, 706
herd() method, 176
high-level languages, 10–11
History interface, 257–259
hoisting function declaration, 150
hot reloading, 277–278, 515
HTML
  importing modules into, 229–230
  including JavaScript files in, 37–40
  JSX as not, 282
  running inside script elements, 36–37
  writing output with JSX, 283–284
  writing templates, 369–371
HTML event attributes
  about, 185
  running JavaScript from, 35–36
HTTP headers, 633–634
HTTP request, 632–633
HTTP server, 561
Hypertext Transfer Protocol (HTTP)
  about, 216–217
  request method, 217
  status codes, 217–218

I

icons, explained, 5–6
id property, 255
identifying
  component contents, 439–447
  render blocking, 237–238
  state, 268
  types of streams, 592
IDs, creating, 682–683
if...else statements, 91–93
ImageCapture API, 243
images property, 250
immutable values, 67
implementing
  inverse data flow, 269–270
  user registration, 734–741
implicit headers, 637
import() function, 229
import statement, 227–228
importing
  fs module, 613–614
  module objects, 228–229
  modules, 227–228
  modules into HTML, 229–230
imports, renaming, 228
includes() method, 112, 610
increasing maximum number of listeners, 646–647
increment (++) operator, 86–87, 97
incrementCount() method, 308
IndexedDB API, 243
index.js, refactoring, 506–507
indexOf() method, 72, 112
inequality (!=) operator, 85–86
inheritance
  about, 169
  in object-oriented programming, 321–322

initializing
projects
about, 497–499
with Vite, 271–278
reactive data, 400–403
variables, 64
inline handlers
about, 411–413
choosing between method handlers and, 414–416
inline styling, 293
`innerHeight` property, 248
`innerHTML` property, 255
`innerWidth` property, 248
`input` event, 187
input types, using `v-model` directive with various, 420
inserting documents into collections, 691–692
Inspect interface (Chrome DevTools), 672
installing
dependencies, 709
ESLint, 536–537
Express, 697–698
Express API server, 709
Git, 56–57
Jest, 547–548
Live Server, 47–50
module bundler, 514–515
MongoDB, 677–680
Mongoose, 731–733
Node.js, 43
Node.js driver, 689
Visual Studio Code (VSC), 23–24
Volar, 346
Vue DevTools, 353–355
instances, creating, 170
integrating ESLint into build script, 540–541
integration testing, 547
interfaces
connecting with, 242
Web, 241–243
interfacing, with web browsers, 234–235
Internet, 15–16
invoking functions, 117
`isFinite()` function, 141
`isNaN()` function, 141
`it()` function, 550–551
iterating, over buffers, 611

J

JavaScript. *See also* asynchronous JavaScript
about, 9–12
conditional rendering with, 378
copying, 15–22
expressions in JSX, 285–286
history of, 12–15
including files in HTML, 37–40
reading, 15–22
rerunning commands in console, 34–35
running
code in browser window, 35–40
code in console, 33–34
from HTML event attributes, 35–36
single-threadedness of, 183–184
starting development environment, 22–29
testing, 535–555
using in templates, 371–373
writing programs in, 29–32
JavaScript engine, 237
JavaScript Object Notation (JSON)
getting data, 222
sending data, 222
working with, 220–222
JavaScript runtime model, 182–183
JavaScriptCore, 237
Jest
how it works, 550–551
installing and configuring, 547–548
`join()` method, 112, 121, 622
`json()` method, 213, 606–607
JSX
conditionally rendering, 286–290
creating lists, 290–291

fundamentals of, 281–282
JavaScript expressions in, 285–286
as not HTML, 282
styling React apps and components, 292–296
transpiling with Babel, 283
using built-in components, 284–285
writing, 281–296
writing HTML output with, 283–284
as XML, 282–283

K

key modifiers, 417–418
`keydown` event, 187, 418
keys, specifying, 383
`keyup` event, 187, 418
keywords, 30

L

`language` property, 244
`lastIndexOf()` method, 112
launching VS Code terminal, 272–274
left-associativity, 84
`length()` method, 610
`length` property, 110
less than (<) operator, 85–86
less than or equal to (<=) operator, 85–86
`let`, creating variables with, 63–66
levels, of programming language, 10
lexical scoping, 332
libuv library, 563–564
lifecycle
component, 309–316
Svelte, 471–476
`limit()` method, 685
limitations, of `reactive()`, 402–403
limiting lists, 685
line comments, 51
linters, using, 536–541
`listen()` method, 698
listeners
passing arguments to, 642–644
removing, 647–648
setting with `v-on`, 411–416
listening
for events, 184–195
for events with `on:` directive, 463–466
on multiple targets, 190–191
listing
documents, 682–683
files in directories, 623
lists
creating, 290–291
limiting, 685
rendering engine, 379–383
sorting, 684–685
literal notation, creating objects using, 127
Live Server, installing, 47–50
`loadBook()` function, 208
loading
dynamic modules, 229
without streams, 588–589
local styles, 293
`localStorage` property, 248
local-*versus*-global installs, 503
`location` property, 248
log points, setting, 671–672
logical operators, 87–88
login route
finishing, 746–753
testing, 749–750
loops
with array methods, 117–121
creating, 97–103, 456–458
event, 183–184
loosely typed language, 68

M

machine code, processor-specificity of, 10
macOS
installing MongoDB on, 680
starting MongoDB on, 681
`main.js` file, 348

map() method, 112, 119–121
mapping arrays, 119–121
Markdown files, 53–54
matches() method, 256
maxListeners, 644–647
messages, queuing, 183
metadata, in package.json file, 500
method handlers
  about, 413–414
  choosing between inline handlers and, 414–416
methods
  addEventListener(), 184, 186–190, 193–194, 199, 200–202, 256, 328, 413, 415
  addJS(), 643
  addListener(), 649
  after(), 256
  afterUpdate(), 473
  alert(), 37, 199, 248, 372–373
  append(), 251, 254–255, 256
  app.get(), 703
  apply(), 157
  app.post(), 703
  app.put(), 703
  app.use(), 703, 717–718
  array, 112–116, 117–121
  Array.map(), 147, 291
  before(), 256
  beforeUpdate(), 473
  bind(), 158, 161–162, 308, 331
  blob(), 213–214
  blur(), 248
  Buffer, 609–611
  Buffer.alloc(), 607–609, 610
  Buffer.compare(), 610
  Buffer.concat(), 610
  Buffer.from(), 609, 610
  Buffer.isBuffer(), 610
  of built-in objects, 142
  call(), 157
  clearInterval(), 248
  clearTimeout(), 248
  close(), 248
  compare(), 610
  componentDidMount(), 309, 312–313, 316–317, 318
  componentDidUpdate(), 313, 315
  componentWillUnmount(), 315–316, 319–320, 365–366
  concat(), 72, 112
  console.log(), 593
  constructor(), 309–310
  createAttribute(), 251
  createComment(), 251
  createElement(), 251, 254–255
  createEvent(), 251
  createInterface(), 641
  createReadStream(), 593–594
  createRoot(), 278–279
  createServer(), 630–631
  createTextNode(), 251
  createUser(), 687–688
  creating in classes, 173–174
  defined, 140
  defining, 174–177
  defining using arrow syntax, 332–333
  DELETE, 217
  deleteMany(), 696
  deleteOne(), 696
  dispatchEvent(), 192, 194, 256
  displayBookstore(), 163–164
  displayCart(), 161
  Document interface, 250–251
  drink(), 173–174
  drive(), 155, 156, 157
  element, 256–257
  emit(), 640
  end(), 636, 702
  equals(), 610
  every(), 112
  express(), 698, 703, 704–705
  express.Router(), 725–728
  express.static(), 706–707
  fetch(), 211, 215–216, 222, 407, 655
  filter(), 112, 121

find(), 684, 685, 694
findOne(), 694
focus(), 194, 248
forEach(), 112, 693–694
fs.open(), 614–616
fs.read(), 614–617
fs.readdir(), 623
fs.readFile(), 671–672
fs.stat(), 624–625
fs.write(), 619–620
fs.writeFile(), 620
GET, 17, 217
get(), 698, 699, 701
getAttribute(), 256
getDerivedStateFromProps(), 309, 310, 313
getElementById(), 186, 251, 252
getElementsByClassName(), 251, 256, 511
getElementsByTagName(), 251, 256
getMacListeners(), 645
getProducts(), 168
getSnapshotBeforeUpdate(), 313, 314–315
getUserApproval(), 602
hasAttribute(), 256
herd(), 176
includes(), 112, 610
incrementCount(), 308
indexOf(), 72, 112
join(), 112, 121, 622
json(), 213, 606–607
lastIndexOf(), 112
length(), 610
limit(), 685
listen(), 698
map(), 112, 119–121
matches(), 256
mongoose.model(), 730–731
mongoose.Schema(), 730–731
mounting, 309–313
move(), 169
naming, 115
next(), 693–694, 702
Object.create(), 130
on(), 640, 642–643, 649
onBeforeMount(), 366
onBeforeUnmount(), 366
onBeforeUpdate(), 366
once(), 649
onDestroy(), 473, 485–486
onErrorCaptured(), 366
onMounted(), 362–366
onUnmounted(), 365–366
onUpdated(), 365
open(), 248
operate(), 158–159
overriding in derived classes, 174
.parse(), 291
PATCH, 217
patch(), 699
pipe(), 590, 592, 598–599, 600
play(), 376
pop(), 112, 113–116, 442–445
POST, 217
post(), 699
prepend(), 251, 254–255, 256
preventDefault(), 195, 375–376
prompt(), 199, 248
properties and, 173–177
push(), 112, 113–116, 442–445, 592
pushState(), 258
PUT, 217
put(), 699
querySelector(), 251, 256
querySelectorAll(), 251, 256
read(), 595
Readable(), 592
readdir(), 624
readFile(), 585, 617–618
readFileSync(), 618–619
reduce(), 111, 112, 118–119
remove(), 256
removeAllListeners(), 648
removeEventListener(), 194
removeListener(), 647–648

methods *(continued)*
render(), 278–279, 307, 309, 310–311, 313–314, 354–355
replaceName(), 602
replaceState(), 258
resolve(), 622
of response object, 636–637
reverse(), 113
Router(), 703
routing, 699–701
scroll(), 248
scrollTo(), 248
set(), 484, 486, 487, 488–490
setHeader(), 637–638
setInterval(), 248, 479
setState(), 308
setTimeout(), 207–208, 248
sharpen(), 126
shift(), 113, 114–115
shouldComponentUpdate(), 313–314
slice(), 72, 110, 113, 116
some(), 113
sort(), 113, 684–685
splice(), 113, 116
stringify(), 222
subscribe(), 483, 484, 485, 487
testEdgeCollision(), 510–511
text(), 214
tick(), 474–476
toArray(), 693–694
toJSON(), 610
toString(), 86, 113, 605, 606–607, 609, 610
unmounting, 315–316
unshift(), 113
unsubscribe(), 485–486, 487
update(), 484, 486, 487
updateMany(), 695–696
updateOne(), 695–696
updating, 313–315
useEffect(), 316–317, 318–319, 320, 362
useState(), 305–306, 316
Window interface, 247–248
writable(), 484
write(), 596–597, 598–599, 610, 637
writeHead(), 636, 637–638
writing, 154–155
XMLHttpRequest(), 203–204
Microsoft, 14
middleware, 701–707
minification, 519
modes, read, 594–595
modifiers
attaching to event listeners, 464–465
directive, 375–376
modifying
arrays, 111
objects, 130–131
React projects, 277–278
module bundler, installing, 514–515
module objects, importing, 228–229
modules. *See also* Node.js
about, 223
creating, 506–512
CSS, 390–391
defining, 224
exporting, 224–226
importing
about, 227–228
into HTML, 229–230
loading dynamic, 229
packages compared with, 580
style, 295
MongoDB
about, 674
connecting to, 681–682
connecting to server, 689–691
installing, 677–680
relational compared with NoSQL databases, 674–677

shell commands, 686–687
starting, 680–681
using from Node.js, 688–696
Mongoose
creating schema with, 730–734
installing, 731–733
mongoose.model() method, 730–731
mongoose.Schema() method, 730–731
Mongosh
connecting to MongoDB, 681–682
creating
collections, 682
databases, 681–682
IDs, 682–683
finding documents, 684–686
listing documents, 682–683
MongoDB shell commands, 686–687
users and roles, 687–688
monitoring scripts, 569
morgan, 706
mount() function, 348, 349
mounting
about, 472–473
methods, 309–313
multiple apps, 349–350
root components, 348–350
mousedown event, 187
mouseenter event, 187
mouseleave event, 187
mousemove event, 187
mouseout event, 187
mouseover event, 187
mouseup event, 187
mousewheel event, 187
move() method, 169
moveBall() function, 507–508
Mozilla Developer Network, 417
multiple style blocks, 389–390
multiplication (*) operator, 83, 86–87

N

named exports, 225
naming
components, 362
constants, 67–68
functions, 144–145
methods, 115
slots, 386–387
variables, 66
NaN data type, 77
Navigator interface
about, 242, 243
properties of-245, 244
quirks of, 244
negation operator (!), 302
Netscape, 14
networking
about, 239, 627
creating web servers, 630–638
data storage, 239–240
with Node.js, 627–638
security for, 628–630
troubleshooting, 50
new line (n), 71
next() method, 693–694, 702
Node.js
about, 559
callback pattern, 584–586
components of, 562–564
core modules, 564–566
creating modules, 575–580
debugging programs, 660–672
Error object, 653–654
getting data to modules, 580–584
how it works, 560–562
installing, 43
networking with, 627–638
uses for, 566–567
using MongoDB from, 688–696
using read-evaluate-print loop (REPL), 570–575
working with, 567–570

Node.js driver, installing, 689
node_modules folder
  about, 275, 347
  using, 502–503
non-simple request, 219–220
NoSQL databases, 674–677
Not (!) operator, 87–88
notation, for square brackets, 131
npm
  managing dependencies with, 497–503
  updating, 503
number data type, 73–74
Number() function, 141
number functions, 73
number inputs, 467
numbers
  converting between strings and, 74
  number inputs and, 467
  using v-for directive for, 380

O

Object.create() method, creating objects with, 130
object-oriented programming, inheritance in, 321–322
objects
  about, 78–79, 125–127
  comparing, 132–134
  copying, 132–134
  creating
    about, 127–130
    with class, 128–129
    with Object.create() method, 130
    using constructor functions, 128
    using literal notation, 127
  deleting properties of, 138
  event, 190
  heaping, 183
  modifying, 130–131
  passing functions between, 158–159
  passing to functions, 156–157
  prototypes, 134–137
  request, 632–633
  response, 633–636
  using v-for directive with, 380–382
on: directive, listening for events with, 463–466
on() method, 640, 642–643, 649
onBeforeMount() method, 366
onBeforeUnmount() method, 366
onBeforeUpdate() method, 366
once() method, 649
once option, 188
onClick attribute, 328
onclick property, 185
onDestroy() method, 473, 485–486
onErrorCaptured() method, 366
onLine property, 244
onMount() function, 472–473, 476–477, 478
onMounted() method, 362–366
onMouseMove() function, 190
onUnmounted() method, 365–366
onUpdated() method, 365
open() method, 248
opening
  array elements, 110–111
  files using Promises, 616–617
  files with fs.open() and callbacks, 614–616
operate() method, 158–159
operational errors, 651, 652
operators
  about, 30, 81
  arithmetic, 86
  assignment, 85
  combining, 89
  comparison, 85–86
  complex queries using, 685–686
  concatenation, 86–87
  delete, 112
  logical, 87–88
  precedence of, 83–84
  short-circuit, 88

ternary, 92–93
types of, 83–89
opinionated code formatter, 44
optimizing
automating build script, 513–522
converting to React, 522–534
Options API, 350–351
options request, creating, 635
Or (||) operator, 87–88
outer function, 146
overriding methods in derived classes, 174

P

package management, 497
`package.json` file
about, 276
components of, 500–502
setting module type in, 579–580
`package-lock.json` file, 276
packages, modules compared with, 580
parameters
for `addEventListener()` method, 186–190
of functions, 140
path, 701
rest, 145–146
setting default, 147–148
parentheses, operators and, 84–85
`.parse()` method, 291
`parseFloat()` function, 73, 86, 141
`parseInt()` function, 73, 86, 141, 142
parsing `Response`, 212–213
passing
arguments
about, 145–148, 584
to directives, 375
to event handlers, 329–330
to listeners, 642–644
to transitions, 470
callback functions to array methods, 117–118
event handler functions from function components, 333–334
functions
as arguments, 146–147
to child, 159–165
between objects, 158–159
as props, 330–334
objects to functions, 156–157
props, 396–398, 440–442
by reference, 143
by value, 142–143
passing arguments, 140
`passive` function, 188
`PassThrough` stream, 599
password security, 736
`patch()` method, 699
`PATCH` method, 217
paths
choosing, 91–96, 454–456
parameters, 701
patterns, 700
regular expressions, 700
string, 699
using, 621–623
patterns, path, 700
paused mode, 595
`pdfViewerEnabled` property, 244
performing effects on unmounting, 319–320
period (.), 79
permission issues, troubleshooting, 50
`permissions` property, 245
personally identifiable information (PII), 744
`pickAMovie()` function, 153
PII (personally identifiable information), 744
`pipe()` method, 590, 592, 598–599, 600
`play()` method, 376
polyfill, 211
polymorphism, 169
`pop()` method, 112, 113–116, 442–445
popping, 113–114
`position` argument, 620
`post()` method, 699
`POST` method, 217

postData() function, 414
Postman, 709
Posts model, 734
practicing, with classes, 177–179
precedence, of operators, 83–84
prefix notation, 97
prepend() method, 251, 254–255, 256
Prettier, 44–47
preventDefault() method, 195, 375–376
preventing
  default actions, 195
  render blocking, 237–238
primitive data types, 69
private members, 175–176
Pro Git, 62
programmer errors, 651, 652
programming
  with array methods, 112–116
  sign-up route, 735–736
  web browsers, 241–259
programming language, JavaScript as a, 10–12
projects
  creating in Visual Studio Code (VSC), 25–27
  initializing, 497–499
  initializing with Vite, 271–278
promises
  about, 202–203
  catching exceptions with, 656–659
  converting chains to async functions, 208–209
  writing, 203–206
prompt() method, 199, 248
properties
  scripts, 250
  scrollX, 248
  scrollY, 248
  stylesheets, 250
properties (props)
  activeElement, 250
  attributes, 255
  body, 250
  children, 250, 255
  classList, 255
  composition using explicit, 322–323
  computing, 405–408
  console, 248
  as a data type, 300–301
  defining
    about, 174–177, 440
    with <script setup>, 396–397
    with setup() function, 397–398
  devicePixelRatio, 248
  _dirname, 594
  Document interface, 249–250
  event, 248
  of event handlers, 185–186
  exporting, 440–442
  forms, 250
  fullScreen, 248
  geolocation, 245
  head, 250
  id, 255
  images, 250
  innerHeight, 248
  innerHTML, 255
  innerWidth, 248
  language, 244
  length, 110
  localStorage, 248
  location, 248
  methods and, 173–177
  Navigator interface, 244–245
  of objects, 138
  onclick, 185
  onLine, 244
  passing
    about, 396–398, 440–442
    functions as, 330–334
  pdfViewerEnabled, 244
  permissions, 245
  using, 440–442
  Window interface, 247, 248

props.children, 323
prototypes, 134–137
proxy objects, reactive() function and, 401–402
public folder, 275, 347
public members, 175
Pug, 714–715
push() method, 112, 113–116, 442–445, 592
pushing, 113–114
pushState() method, 258
put() method, 699
PUT method, 217

Q

querySelector() method, 251, 256
querySelectorAll() method, 251, 256
question mark (?), 700
queuing messages, 183
quotation mark ("), 70

R

React
  apps
    structure of, 274–276
    styling, 292–296
  components
    about, 297–298
    class components, 307–308
    composing, 321–325
    data types, 300–304
    designing elements, 298–299
    function components, 304–306
    lifecycle of, 309–316
    returning valid values, 300
    using lifecycle in function components, 316–320
  configuring Webpack for, 522–524
  converting to, 522–534
  converting UI to, 524–529
  creating forms with, 334–339
  event handling in, 328–334
  using data and events in, 327–339
  Vue compared with, 343–344
React library, 270
React Testing Library, 553–555
ReactDOM library, 270, 278–279
reacting, to state changes with watch(), 408–410
reactive data
  changing, 400–403
  creating, 432–434
  initializing, 400–403
reactive() function, 360, 391–393, 400–403
reactive shortcuts, 487–488
reactive statements, creating, 445–446
reactivity
  data and, 395–410
  with state, 301–304
  Svelte, 483–492
  triggering with assignments, 442
Reactivity Transform, 404–405
ReactJS
  about, 263–264
  building React UI, 265–270
  component-based nature of, 270–271
  declarative nature of, 271
  distilling thinking in, 264
  initializing projects with Vite, 271–278
  as JavaScript, 271
  ReactDOM, 278–279
  Virtual DOM (VDOM), 278–279
read() method, 595
readable stores, creating, 484
readable streams, creating, 592–595
readdir() method, 624
read-evaluate-print loop (REPL)
  about, 570
  commands, 573–575
  playing with Node.js, 570–573
readFile() method, 585, 617–618

readFileSync() method, 618–619
reading
  error.stack, 654
  files, 614–619
  JavaScript code, 15–22
  readable streams from fs module, 593–594
  semver, 501–502
  stack frames, 654
  synchronous code, 198–199
README file, 52–53
recognizing data types, 300–304
reduce() method, 111, 112, 118–119
reducer function, 118–119
reducing arrays, 118–119
ref() function, 391–393, 400–404
ref object, 403–404
refactoring index.js, 506–507
reference, passing by, 143
refresh tokens, sending, 745–746
refreshing data, 477–479
registration and authentication
  about, 721–722
  adding app and server modules, 723–724
  creating
    basic routes, 725–728
    and configuring directory, 722–723
    schema with Mongoose, 730–734
  finishing login route, 746–753
  generating and using tokens, 744–746
  handling authentication, 741–743
  implementing user registration, 734–741
  testing routes, 728–730
regular expressions, path, 700
relational databases, 674–677
remainder (%) operator, 86–87
Remember icon, 6
remove() method, 256
removeAllListeners() method, 648
removeBook() function, 161–162
removeEventListener() method, 194
removeListener() method, 647–648
removing
  event listeners, 194
  listeners, 647–648
renaming exports/imports, 228
render blocking, identifying and preventing, 237–238
render() method, 278–279, 307, 309, 310–311, 313–314, 354–355
rendering engine, 235–238
rendering lists, 379–383
REPL (read-evaluate-print loop)
  about, 570
  commands, 573–575
  playing with Node.js, 570–573
replace function, 72
replaceName() method, 602
replaceState() method, 258
request method, 217
request object, 632–633
requests, making with CORS, 218–220
rerunning commands, in console, 34–35
reset event, 187
resolve() method, 622
responding, to events, 411–420
response body, 635–636
Response methods
  calling, 213–214
  parsing, 212–213
response object
  about, 633–636
  methods of, 636–637
Response.blob(), 213–214
response.statusCode, 637
response.statusMessage, 637
Response.text(), 214
res.render() function, 719
REST client, setting up, 709–711
rest parameters, 145–146

restricted JavaScript, in templates, 372–373
`return` statement, 267, 329, 584
return value, using as an argument, 149
returning
  data, 149–150
  fields, 684
  valid values, 300
`reverse()` method, 113
`reverseOrder()` function, 253
right-associativity, 84
roles, working with, 687–688
root components, mounting, 348–350
`Router()` method, 703
`router.get()` function, 719
router-level middleware, 703–704
routes
  creating basic, 725–728
  testing, 728–730
routing methods, 699–701
running
  code
    in browser windows, 35–40
    on command line, 570
    in console, 33–34
  effects, 317–319
  ESLint, 537
  HTML inside script elements, 36–37
  JavaScript from HTML event attributes, 35–36
runtime environments, web browsers as, 561

S

salt, adding to hashing, 738
sandbox, 21, 561
saving, hashing and, 738–739
scaffolding Vue.js applications, 344–348
schema, creating with Mongoose, 730–734
scope, 80, 150
scoped CSS, 388–389
script elements, running HTML inside, 36–37
`<script setup>`, defining props with, 396–397
`<script>` element, 358, 446–447
scripts
  build, 513–522
  dev, 505–506
  monitoring, 569
  npm, 500
  in Svelte components, 439–440
`scripts` property, 250
`scroll()` method, 248
`scrollTo()` method, 248
`scrollX` property, 248
`scrollY` property, 248
SDLC (systems development life cycle), 513–514, 535
security
  networking and, 628–630
  password, 736
`select` event, 187
select inputs, binding, 467–468
selecting
  element nodes, 251
  event targets, 186
  with `getElementById()` method, 252
  between method and inline handlers, 414–416
  paths, 454–456
  using selectors, 252–253
selectors, selecting using, 252–253
`semver`, reading, 501–502
semver versioning, 502
sending refresh tokens, 745–746
server
  defined, 627
  JavaScript on the, 22
server modules, adding, 723–724
server-side routing, with Express, 699–701
server-side services, 22
services, value of, 21–22

serving
  static files, 706–707
  views, 712–715
set() method, 484, 486, 487, 488–490
setHeader() method, 637–638
setInterval() method, 248, 479
setState() method, 308
setTimeout() method, 207–208, 248
setting
  addEventListener() parameters, 186–190
  breakpoints, 544–546, 669–671
  context, 490–492
  context of functions, 157–158
  default parameters, 147–148
  environment variables from command line, 582
  environment variables with .env, 582–583
  listeners with v-on, 411–416
  log points, 671–672
  module type in package.json, 579–580
  watch expressions, 671
setup, of REST client, 709–711
setup() function, 358–361, 377, 396, 397–398, 401
shallow copy, 122–123
sharpen() method, 126
shell commands, MongoDB, 686–687
shift() method, 113, 114–115
shifting, 114–115
short-circuit operators, 88
shorthand names, directive, 375
shouldComponentUpdate() method, 313–314
show dbs command, 686
show roles command, 686
show users command, 686
showLogInScreen() function, 88
side effect, 317, 408
signal function, 188
sign-up route, programming, 735–736
simple request, 218–219
simplifying arrow functions, 152–153
single quote ('), 71
single-file components, 357–361
single-threadedness, of JavaScript, 183–184
slice() method, 72, 110, 113, 116
slicing arrays, 116
slots
  composing with, 383–387, 459–461
  naming, 386–387
software incompatibility, troubleshooting, 50
some() method, 113
sort() method, 113, 684–685
sorting lists, 684–685
source code, 6
source maps, building, 542–544
Sources panel
  about, 542
  editing code in, 544
special characters, 71
specifying
  fallback content, 385–386
  handler functions, 329
  keys, 383
speed, of Node.js, 566–567
SpiderMonkey, 237
splice() method, 113, 116
splicing arrays, 116
split() function, 72, 109–110, 142
spreading arrays, 122–123
square braces ([]), 375
square brackets, notation for, 131
squareIt() function, 117–118
src folder, 275–276, 347–348
stack frames, reading, 654
stacking function calls, 182
standard library, 564
start() function, 488–490
starting
  development environment, 22–29
  functions, 488–490
  MongoDB, 680–681
state
  defined, 301
  determining location of, 269

identifying, 268
reactivity with, 301–304
stateless, functions as, 305
statements
break, 102–103
continue, 102–103
if...else, 91–93
import, 227–228
in JavaScript, 30
return, 267, 329, 584
switch, 93–96
static assets, copying, 519–520
static component, 429
static files, serving, 706–707
static members, 176–177
static version, building in React, 266–268
status codes, HTTP, 217–218
stocking store, 483–490
stop() function, 488–490
stopping functions, 488–490
storage, data, 239–240
strategies, style, 296
Straub, Ben, 62
streaming
about, 587–588
chaining streams, 600–602
chunking, 588
converting to streams, 589–590
creating
duplex streams, 598–599
readable streams, 592–595
writable streams, 596–597
identifying types of streams, 592
loading without streams, 588–589
transforming streams, 599–600
viewing chunks, 590–591
strict equality (===) operator, 85–86
strict inequality (!==) operator, 85–86
string data type, 69–70
String() function, 139–140, 141
string functions, 71–72
string paths, 699
stringify() method, 222
strings
converting between numbers and, 74
creating with template literal notation, 70–71
userAgent, 245–246
using v-for directive for, 380
structure
of Express apps, 716–720
of React apps, 274–276
of Vue app, 347
style
adding, 509–510
adding to components, 387–393, 447–449
attributes, 293–294
React apps and components, 292–296
Vue, 350–353
style modules, creating, 295
style objects, 294
stylesheets property, 250
submit event, 187
subscribe() method, 483, 484, 485, 487
subscribing, to stores, 485
substring function, 72
subtraction (-) operators, 83, 86–87
super() function, 171, 172, 175, 176, 307
Svelte
about, 423–424
building scaffolding, 425–426
constructing store, 483–490
documenting with comments, 453–454
fetching data in, 476–481
getting context, 490–492
getting for VS Code extension, 426–427
lifecycle of, 471–476
playing with, 431–436
reactivity, 483–492
setting context, 490–492
stocking store, 483–490
Svelte apps, 427–430

Swagger, 709
switch statements, 93–96
Symbol data type, 77
Synthetic Events, dispatching, 328–329
SyntheticBaseEvent object, 328–329
system testing, 547
systems development life cycle (SDLC), 513–514, 535

T

tab (t), 71
targets, listening on multiple, 190–191
TCP/IP standard, 628
Technical Stuff icon, 6
template engines, 713–714
template literal notation, creating strings with, 70–71
templates
  binding data to, 398–400
  choosing a path, 454–456
  composing with slots, 459–461
  creating loops, 456–458
  designing, 451–461
  documenting Svelte with comments, 453–454
  elements as building blocks, 451–453
  Vue, 369–393
  writing text expressions, 458–459
ternary operator, 92–93
test() function, 549, 550–551
test specs, 550–551
test suites, 550
testEdgeCollision() method, 510–511
testing
  API server, 711–712
  for collisions, 510–511
  Git, 57–60
  JavaScript, 535–555
  login route, 749–750
  routes, 728–730
  user registration, 740–741
  writing code through, 552
  writing tests, 548–550
text expressions, writing, 458–459
text() method, 214
then() function, 203
thin clients, 17
third-party APIs, 243
third-party middleware, 705
this keyword, 154
three dots (. . .), 122–123
tick() method, 474–476
Tip icon, 5
toArray() method, 693–694
ToggleVisibility class, 330–331
toJSON() method, 610
tokens, generating and using, 744–746
toLowerCase function, 72
toolbox
  about, 41–42
  coding with Git, 55–62
  configuring Visual Studio Code, 43–50
  documenting code, 51–54
  installing Node.js, 43
top-level functions, 141
toString() method, 86, 113, 605, 606–607, 609, 610
toUpperCase function, 73
Transfer-Encoding HTTP headers, 634
transform streams, 599–600
transition animations
  creating, 468–469
  creating unidirectional transitions, 470
  passing arguments to transitions, 470
transpilation, with Babel, 283
triggering
  built-in events, 193–194
  custom events, 194
  reactivity with assignments, 442

trim() function, 142
triple equals operator (===), 132
troubleshooting, 50
two-way bindings
  creating with :bind directive, 466–468
  creating with v-model, 420
typeof operator, 68

U

unblocking code, with async and defer, 238
uncaught exception, 654
uncontrolled forms, 337–339
undefined data type, 77
underscores, in JavaScript, 32
uneval() function, 141
unidirectional connection, 667
unidirectional transitions, 470
unit testing
  about, 547
  configuring Jest, 547–548
  installing Jest, 547–548
  writing tests, 548–550
unmounting methods
  about, 315–316
  performing effects on, 319–320
unshift() method, 113
unshifting, 114–115
unsubscribe() method, 485–486, 487
unsubscribing, from stores, 485–486
update() method, 484, 486, 487
updateMany() method, 695–696
updateOne() method, 695–696
updating
  about, 6
  array methods and, 442–445
  data, 695–696
  DOM, 354–355
  npm, 503
updating methods, 313–315
use db command, 686
useEffect() method, 316–317, 318–319, 320, 362
user interface
  converting to React, 524–529
  defined, 234
user registration
  implementing, 734–741
  testing, 740–741
userAgent string, 245–246
userModel, creating, 733–734
users, working with, 687–688
useState() method, 305–306, 316
UTF-8, 605–606

V

V8 engine, 237, 563
values
  about, 30
  passing by, 142–143
  returning valid, 300
variable scope, 80
variables
  creating with let, 63–66
  declaring, 64
  environment, 581–583
  initializing, 64
  naming, 66
  using, 64–66, 67
v-bind directive, 391–393, 399–400
v-else directive, 379
v-else-if directive, 379
v-for directive
  about, 380
  using with arrays, 382–383
  using with objects, 380–382
viewing chunks, 590–591
views, serving, 712–715

v-if directive, 379
Virtual DOM (VDOM), 278–279
Visual Studio Code (VSC)
  about, 22–23
  configuring, 43–50
  getting Svelte for extension, 426–427
  installing, 23–24
  launching terminal, 272–274
  learning to use, 25–29
visualizing Document interface, 249
Vite, initializing projects with, 271–278
vite.config.js file, 276
v-model directive, 420
Volar, installing, 346
v-on, setting listeners with, 411–416
v-show directive, 378
Vue
  components, 357–368
  following component lifecycle, 362–366
  handling errors in components, 366–368
  installing Vue DevTools, 353–355
  mounting root components, 348–350
  naming components, 362
  React compared with, 343–344
  scaffolding Vue.js applications, 344–348
  single-file components, 357–361
  structure of app, 347
  styles, 350–353
  templates
    adding style to components, 387–393
    composing with slots, 383–387
    conditional rendering, 378–379
    directives, 373–377
    rendering lists, 379–383
    using JavaScript templates, 371–373
    writing HTML templates, 369–371
Vue DevTools, installing, 353–355
vue-create, bootstrapping with, 344–346
Vue.js applications, scaffolding, 344–348

W

Warning icon, 6
watch expressions, 546, 671
watch() function, reacting to state changes with, 408–410
Web APIs, 241–243
web application server, 627
web browsers
  about, 233–234, 241
  browser engine, 235
  History interface, 257–259
  HTML Document interface, 249–257
  interfaces, 241–243
  interfacing with, 234–235
  Netscape Navigator, 243–245
  networking, 239–240
  programming, 241–259
  rendering engine, 235–238
  as runtime environments, 561
  user agent string, 245–246
  Web APIs, 241–243
  Window interface, 246–248
web servers
  creating, 630–638
  defined, 627
Web Worker API, 243
WebGL API, 243
WebKit, 235
Webpack
  about, 515–516
  configuring for React, 522–524
websites
  Bootstrap, 292
  Cheat Sheet, 6
  ECMAScript, 15
  Git, 56
  GitHub, 62
  Mozilla Developer Network, 417

Node.js, 43
Postman, 709
Pro Git, 62
source code, 6
Swagger, 709
updates, 6
WebSocket, 667
`while` loops, 100–101
white space, JavaScript and, 31
`Window` interface
about, 246–247
methods, 247–248
properties, 247, 248
`window.alert()` function, 414
Windows
installing MongoDB on, 677–679
starting MongoDB on, 681
workspaces, adding, 669
`writable()` method, 484
writable streams, creating, 596–597
`write()` method, 596–597, 598–599, 610, 637
`writeHead()` method, 636, 637–638
writing
anonymous functions as arrow functions, 152
asynchronous JavaScript, 197–222
code through testing, 552
components, 437–439
dev scripts, 505–506
to disks with `fs.write()`, 619–620
files, 503–505, 619–620
to files using Promises, 616–617
function bodies, 148
functions, 144–154
HTML output with JSX, 283–284
HTML templates, 369–371
JavaScript programs, 29–32
JSX, 281–296
methods, 154–155
Node.js programs, 568–569
promises, 203–206
tests, 548–550
text expressions, 458–459

X

XML, JSX as, 282–283, 285
`XMLHttpRequest()` method, 203–204
XSS (cross-site scripting), 218, 745–746
XSS attacks, 745–746

Z

zero-based indexing, 79