

LEARNING MADE EASY



HTML, CSS, & JavaScript[®]

ALL-IN-ONE

for
dummies[®]
A Wiley Brand



5
Books
in one!

Paul McFedries

Author of *Web Coding & Development*
All-in-One For Dummies

Index

SYMBOLS

- + (addition operator), 430, 613–614
- & (ampersand), 109
- && (AND operator), 447–448, 450–451, 462
- * (asterisk), 233, 243
- \ (backslash), 17, 102, 422
- ` (back ticks), 585, 623
- { } (braces), 27, 459, 483, 646
- : (colon), 253
- + (concatenation), 436–437
- (decrement operator), 433
- / (division operator), 434–435
- . (dot), 240–242, 516
- :: (double colon), 278
- " (double quotation marks), 421
- // (double-slash), 408
- (em dash), 108–109
- == (equality operator), 439–440, 646
- = (equals sign), 413
- >= (greater than or equal operator), 442
- > (greater-than symbol), 82, 109, 245, 441
- # (hashtag symbol), 104, 242–243
- (hyphen), 104
- === (identity operator), 444, 646
- ++ (increment operator), 430–431
- != (inequality operator), 440–441
- <= (less than or equal operator), 442
- < (less-than symbol), 82, 109
- (minus sign), 420
- % (modulus operator), 324, 435
- * (multiplication operator), 433
- ! (NOT operator), 449
- || (OR operator), 448–449, 450–451, 462
- + (plus sign), 246–247, 420
- ; (semicolon), 36, 109
- ' (single quotation marks), 421
- / (slash), 17
- . . . (spread operator), 572–573
- [] (square brackets operator), 662
- === (strict equality operator), 444
- !== (strict inequality operator), 444
- (subtraction and negation operators), 431–432
- ? : (ternary operator), 446–447
- ~ (tilde), 245–246
- _ (underscore), 104, 417
- absolute positioning, 688–690
- accessibility semantics, 191–198
 - adding form field labels, 192
 - ARIA roles, 192–198
- Accessible Rich Internet Applications (ARIA) roles, 192–198
- accordions, 166
- :active pseudo-class, 268–269
- activeElement property, 524
- ad requirements, 51
- adaptive layouts, 779–791
 - container queries, 783–789
 - media queries, 780–783
 - user preference queries, 789–791
- add() method, 541
- addEventListener() method, 664
- addition operator (+), 430, 613–614
- <address> tag, 110–111
- administration interface, 51
- Adobe Fonts, 368
- ::after pseudo-element, 279–281, 685
- age, 608–610
- alert() method, 36, 481
- align-content property, 762
- aligning, 394–396
 - flex containers
 - centering element horizontally and vertically, 713–714

NUMBERS

0-width (ch), 324

A

- <a> tag, 102–103, 128
- <abbr> tag, 110
- absolute measurement units, 323–324

- aligning (*continued*)
 - flex items along primary axis, 709–710
 - flex items along secondary axis, 711–712
 - rows or columns along secondary axis, 716–718
- grid properties, 761–764
- typography
 - aligning text, 394
 - hyphenating text, 395
 - indenting paragraph first line, 396
- align-items property, 762
- align-self property, 712
- alt text, 125, 190
- ampersand (&), 109
- anchors, 104
- AND operator (&&), 447–448, 450–451, 462
- Android, 60
- anonymous functions,
 - 495–498
 - assigning to variables, 496–497
 - replacing function call with, 497–498
- :any-link pseudo-class, 271
- aperture, 371–372
- Apple Safari, 40, 60, 293, 631
- application programming interface (API), 522
- arguments
 - defined, 489
 - making argument optional, 492–493
 - used with Date object, 600
- ARIA (Accessible Rich Internet Applications) roles, 192–198
- arithmetic assignment operators, 436
- arithmetic operators, 429–430

- array literals, 557
- arrays, 553–582
 - declaring, 554–555
 - defined, 554
 - iterating, 559–569
 - every() method, 562–563
 - find() method, 568–569
 - forEach() method, 560–561
 - for...of loop, 561–562
 - map() method, 565–566
 - reduce() method, 566–567
 - some() method, 564–565
 - manipulating, 570–582
 - concat() method, 571–572
 - join() method, 573–574
 - length property, 570–571
 - pop() method, 574–575
 - push() method, 575–576
 - reverse() method, 576
 - shift() method, 576–577
 - slice() method, 577–578
 - sort() method, 578–580
 - splice() method, 580–582
 - unshift() method, 582
 - multidimensional arrays, 569–570
 - populating with data, 555–559
- arrow functions, 498–500
- <article> tag, 91
- ascenders, 372
- <aside> tag, 92–93
- assistive technology (AT), 185–187
 - for auditory disabilities, 186
 - for cognitive disabilities, 187
 - for motor disabilities, 187
 - ::before pseudo-element and, 283
 - for visual disabilities, 186

- asterisk (*), 233, 243
- Atom, 44
- attribute selectors, 247–249
- audio, HTML, 137–141
 - <audio> tag, 140–141
 - third-party audio, 137–139
 - web accessibility, 189–191
 - web audio formats, 139
- <audio> tag, 140–141
- auditory disabilities, 186
- auto keyword, 744–745
- auto-fill keyword, 744
- auto-fit keyword, 773
- autoplay attribute, 135, 140
- axes, grid, 751–753
- Axis-Praxis, 387

B

- tag, 99
- back ticks (`), 585, 623
- background, 346–354
 - applying color to, 346–347
 - background-image property, 347–352
 - shorthand property, 352–354
- background-image property, 347–352
- backslash (\), 17, 102, 422
- Bandcamp, 138
- bandwidth, 50, 184, 386
- baseline, 371
- baseline value, 129
- ::before pseudo-element, 281–283
- block boxes, 208, 233
- block dimension, 219
- block scope, 459, 504–505
- block statement, 459
- block variables, 646
- block-level elements, 83, 676

- <blockquote> tag, 100–101
- blur method, 666–667
- body property, 524
- body section, 20
- <body> tag, 20, 35, 80, 408–409
- Boolean literals, 423
- border property, 159–160
- border-radius property, 218, 220–221
- borders
 - box model, 207, 215–223
 - applying, 215–218
 - colors for, 359
 - rounding, 218–223
 - color of, 216
 - table, 159–161
- bottom value, 129
- box model, 205–233
 - adding padding, 213–215
 - block and inline boxes, 208
 - borders, 215–223
 - applying, 215–218
 - rounding, 218–223
 - changing display type for box, 209–210
 - components of, 206–207
 - dimensions, 228–233
 - converting inline element to block-level element, 233
 - minimum or maximum, 231–232
 - element box, 210–211
 - inspecting, 297–298
 - margins, 223–228
 - collapsing, 225–228
 - negative margins, 225
 - whitespace, 211–213
- box-sizing property, 232–233
-
 tag, 22, 86
- braces ({}), 27, 459, 483, 646
- Brackets website, 44

- break mode, 634–637
- break statement, 476–478
- breakpoint, 635–636
- broad selectors, 313
- browser market share, 60–61
- browser window events, 548
- browsers. *see* web browsers
- BrowserStack, 62
- bulleted lists, 105–107
- <button> tag, 167
- buttons, form, 167–168

C

- calculations, 333–336
 - calc() function, 333–334
 - clamp() function, 335–336
 - max() function, 335
 - min() function, 334–335
- callback function, 497–498, 547, 560
- calling functions, 484–488
 - after page is loaded, 486–487
 - parsed <script> tag, 484–485
 - in response to events, 487–488
- Canva, 124
- capitalization, 389
- captions
 - figures with, 127–128
 - for tables, 150–151, 162
- caption-side property, 162
- carriage return, 22
- cascade
 - algorithm, 318–319
 - defined, 26
 - overview, 309–310
- cascading style sheets (CSS), 205–396
 - applying styles, 306–319

- cascade, 309–310
- cascade algorithm, 318–319
- declaration types, 310–311
- inheritance, 306–309
- origin types, 311
- source code order, 317–318
- specificity, 313–317
- weight hierarchy, 311–313
- basics of, 24–32
 - adding styles to pages, 28–32
 - cascading, 26
 - rules and declarations, 26–27
 - sheets, 25–26
 - styles, 24–25
- box model, 205–233
 - adding padding, 213–215
 - block and inline boxes, 208
 - borders, 215–223
 - changing display type for box, 209–210
 - components of, 206–207
 - dimensions, 228–233
 - element box, 210–211
 - margins, 223–228
 - whitespace, 211–213
- calculating with, 333–336
 - calc() function, 333–334
 - clamp() function, 335–336
 - max() function, 335
 - min() function, 334–335
- colors, 337–362
 - background, 346–354
 - border colors, 359
 - gradients, 354–358
 - specifying, 338–343
 - text, 344–346
 - web development tools, 359–362
- commenting code, 288–290

- cascading style sheets
 - (continued)*
 - non-obvious code, 288–289
 - stylesheet sections, 289–290
- CSS resets, 290–291
- debugging, 292–306
 - adding class to element, 304–305
 - adding element declaration to inspector stylesheet, 302–304
 - adding inline declaration to element, 302
 - disabling declaration, 301
 - displaying web development tools, 292–294
 - editing property value, 299–301
 - inspecting elements, 294–299
 - simulating pseudo-class state, 305–306
- general discussion of, 8
- measurement units, 321–333
 - absolute, 323–324
 - relative, 324–331
 - viewport, 332–333
- modifying with JavaScript, 540–544
 - adding class to element, 541–544
 - changing element styles, 540–541
- pseudo-classes, 251–278
 - child-related pseudo-elements, 255–260
 - form element state pseudo-classes, 264–267
 - functional pseudo-classes, 271–278
 - matching elements by user action, 268–270
 - matching links, 270–271
 - overview, 251–253
 - styling elements with, 253–255
 - type-related pseudo-classes, 260–264
- pseudo-elements, 278–286
 - `::before`, 281–283
 - `::after`, 279–281
 - `::first-letter`, 283–284
 - `::first-line`, 285–286
 - working with, 278–279
- reasons to learn, 8–9
- selectors, 235–249
 - attribute selectors, 247–249
 - combinators, 243–247
 - examples of, 238–239
 - selector lists, 239
 - standard selectors, 240–243
 - web page relationships, 236–238
- typography, 363–396
 - alignment, 394–396
 - alternative numeric glyphs, 391–392
 - capitalization, 389
 - font shorthand property, 392
 - font stack, 365–366
 - generic font families, 366–367
 - line height, 376, 380–382, 384
 - line length, 376, 382–384
 - links, 392–393
 - small caps, 390–391
 - spacing between letters, 390
 - system font families, 367–368
 - text styles, 384–389
 - type size, 376–380, 384
 - typefaces, 364–365
 - web font families, 368–375
 - validating code, 62–63, 69–72
 - online, 69–71
 - in text editor, 72
 - web page basics, 10–14
 - launching new text file, 11
 - saving HTML files, 11–12
 - test driving web pages, 12–14
- case-sensitivity, 644–645
- centering elements, 713–714
- ch (0-width), 324
- change event, 667–668
- character entities, 109
- character references, 109
- `charAt()` method, 591–593
- checkboxes, 543
 - adding in forms, 171–173
 - coding, 654–656
 - checkbox state, 655–656
 - referencing checkboxes, 654
- checked property, 655
- `:checked` pseudo-class, 265
- child combinator, 245, 255
- child elements, 530–534
- `childElementCount` property, 524
- `childNodes` property, 530
- child-related pseudo-elements, 255–260
 - `:first-child` pseudo-class, 255–256
 - `:last-child` pseudo-class, 256–257
 - `:nth-child()` pseudo-class, 257–259
 - `:nth-last-child()` pseudo-class, 259

- `:only-child`
 - pseudo-class, 260
- children property, 524
- Chrome, 40, 60, 292, 630
- circles, 230
- `cite` attribute, 101
- `<cite>` tag, 111–112
- `clamp()` function, 335–336, 779
- class selector, 240–242
- classes
 - adding to elements, 541–544
 - specifying elements by name of, 526
- `clear()` method, 626
- clearing floats, 681–682
- client-side programming
 - language, 404
- Cloudflare Pages, 49
- cm (centimeters), 323
- `<code>` tag, 112–113
- codecs, 133
- CodePen, 45
- coding, 39–46, 59–75
 - choosing text editor, 42–45
 - creating local folders, 40–42
 - root folder, 41
 - subfolders, 41–42
 - cross-browser testing, 60–62
 - browser market share, 60–61
 - testing online, 61–62
 - testing suite, 61
 - setting up test browsers, 40
 - tools for, 44–45
 - validation, 62–75
 - CSS code, 69–72
 - HTML code, 65–69
 - JavaScript code, 72–75
 - overview, 62–65
 - WebDev Workbench, 45–46
- cognitive disabilities, 187
- collapsing margins, 225–228
- collections of elements, 527–529
- colon (:), 253
- color contrast, 200–202
- color picker, 178
- color property, 344
- colors, 337–362
 - background, 346–354
 - applying, 346–347
 - background shorthand property, 352–354
 - background-image property, 347–352
 - border colors, 359
 - gradients, 354–358
 - conic gradient, 358
 - linear gradient, 354–355
 - radial gradient, 356–357
 - specifying, 338–343
 - color keywords, 338–339
 - `hsl()` function, 341–343
 - RGB codes, 340–341
 - `rgb()` function, 339–340
- text, 344–346
 - `color` property, 344
 - text decorations, 344
 - text shadow, 345–346
- web development tools, 359–362

- combinators, 243–247
- child combinator, 245
- descendant combinator, 244
- next-sibling combinator, 246–247
- subsequent-sibling combinator, 245–246
- commenting code, 288–290, 644
- HTML, 95–96
- in JavaScript, 408
- non-obvious code, 288–289
- stylesheet sections, 289–290
- comparison expressions, 439–447
- comparison operators, 439
- data conversion and, 442–443
- equality operator (`==`), 439–440
- greater than or equal operator (`>=`), 442
- greater-than symbol (`>`), 441
- inequality operator (`!=`), 440–441
- less than or equal operator (`<=`), 442
- strict equality operator (`===`), 444
- strict inequality operator (`!==`), 444
- ternary operator (`?:`), 446–447
- using strings in, 445–446
- Comparium, 62
- completion, code, 43
- compound statement, 459
- computed styles, 298–299
- `concat()` method, 571–572
- concatenation (+), 436–437
- conic gradient, 358
- Console, 632–634
- displaying in various browsers, 632–633
- executing code in, 633–634
- logging data to, 633
- `console.log()` method, 633
- `console.table()` method, 633
- `const` keyword, 414–415
- constructors, 555
- `@container` at-rule, 787
- container collapse, 683–685
- container queries, 783–789
- container query units, 789
- containers, 133

- content
 - box components, 207
 - tags dividing, 16
- contenteditable
 - attribute, 267
- context, stacking, 698–701
- continue statement, 479
- contrast ratio, 361–362
- controls attribute, 135, 140
- counters, 371
- cPanel tool, 56
- cross-browser testing code,
 - 60–62
 - browser market share, 60–61
 - testing online, 61–62
 - testing suite, 61
- CSS (cascading style sheets),
 - 205–396
 - applying styles, 306–319
 - cascade, 309–310
 - cascade algorithm, 318–319
 - declaration types, 310–311
 - inheritance, 306–309
 - origin types, 311
 - source code order, 317–318
 - specificity, 313–317
 - weight hierarchy, 311–313
 - basics of, 24–32
 - adding styles to pages, 28–32
 - cascading, 26
 - rules and declarations, 26–27
 - sheets, 25–26
 - styles, 24–25
 - box model, 205–233
 - adding padding, 213–215
 - block and inline boxes, 208
 - borders, 215–223
 - changing display type for box, 209–210
 - components of, 206–207
 - dimensions, 228–233
 - element box, 210–211
 - margins, 223–228
 - whitespace, 211–213
 - calculating with, 333–336
 - calc() function, 333–334
 - clamp() function, 335–336
 - max() function, 335
 - min() function, 334–335
 - colors, 337–362
 - background, 346–354
 - border colors, 359
 - gradients, 354–358
 - specifying, 338–343
 - text, 344–346
 - web development tools, 359–362
 - commenting code, 288–290
 - non-obvious code, 288–289
 - stylesheet sections, 289–290
 - CSS resets, 290–291
 - debugging, 292–306
 - adding class to element, 304–305
 - adding element declaration to inspector stylesheet, 302–304
 - adding inline declaration to element, 302
 - disabling declaration, 301
 - displaying web development tools, 292–294
 - editing property value, 299–301
 - inspecting elements, 294–299
 - simulating pseudo-class state, 305–306
 - general discussion of, 8
 - measurement units, 321–333
 - absolute, 323–324
 - relative, 324–331
 - viewport, 332–333
 - modifying with JavaScript, 540–544
 - adding class to element, 541–544
 - changing element styles, 540–541
 - pseudo-classes, 251–278
 - child-related pseudo-elements, 255–260
 - form element state pseudo-classes, 264–267
 - functional pseudo-classes, 271–278
 - matching elements by user action, 268–270
 - matching links, 270–271
 - overview, 251–253
 - styling elements with, 253–255
 - type-related pseudo-classes, 260–264
 - pseudo-elements, 278–286
 - ::before, 281–283
 - ::after, 279–281
 - ::first-letter, 283–284
 - ::first-line, 285–286
 - working with, 278–279
 - reasons to learn, 8–9
 - selectors, 235–249
 - attribute selectors, 247–249
 - combinators, 243–247
 - examples of, 238–239
 - selector lists, 239
 - standard selectors, 240–243
 - web page relationships, 236–238
 - typography, 363–396
 - alignment, 394–396
 - alternative numeric glyphs, 391–392

- capitalization, 389
 - font shorthand property, 392
 - font stack, 365–366
 - generic font families, 366–367
 - line height, 376, 380–382, 384
 - line length, 376, 382–384
 - links, 392–393
 - small caps, 390–391
 - spacing between letters, 390
 - system font families, 367–368
 - text styles, 384–389
 - type size, 376–380, 384
 - typefaces, 364–365
 - web font families, 368–375
 - validating code, 62–63, 69–72
 - online, 69–71
 - in text editor, 72
 - web page basics, 10–14
 - launching new text file, 11
 - saving HTML files, 11–12
 - test driving web pages, 12–14
 - CSS declaration, 26
 - CSS Flexible Box Layout Module standard, 727
 - CSS Grid, 737–764
 - alignment properties, 761–764
 - assigning grid items, 750–761
 - to columns and rows, 750–757
 - to named grid areas, 757–761
 - defined, 299
 - Flexbox and, 705
 - fluid layouts with, 773–776
 - overview, 738
 - setting up grid container, 739–740
 - specifying grid rows and columns, 740–750
 - explicit grid, 741–745
 - gutters, 749–750
 - implicit grid, 745–749
 - CSS Validation Service, 69–71
- ## D
- Dailymotion, 131
 - data
 - comparison operators and data conversion, 442–443
 - logging to Console, 633
 - populating arrays with, 555–559
 - web storage
 - adding, 623–625
 - getting, 625
 - removing, 626
 - data types, 419
 - databases, 51
 - date argument, 600
 - Date() function, 601
 - Date object, 599–611
 - arguments used with, 600
 - extracting information about date, 601–605
 - performing date calculations, 608–611
 - setting date, 605–608
 - working with, 600–601
 - date picker, 178–179
 - dates and times, 599–611
 - arguments used with Date object, 600
 - extracting information about date, 601–605
 - performing date calculations, 608–611
 - setting date, 605–608
 - working with Date object, 600–601
 - dd argument, 600
 - debugger statement, 637
 - debugging CSS, 72, 292–306
 - adding class to element, 304–305
 - adding element declaration to inspector stylesheet, 302–304
 - adding inline declaration to element, 302
 - disabling declaration, 301
 - displaying web development tools, 292–294
 - editing property value, 299–301
 - inspecting elements, 294–299
 - box model, 297–298
 - computed styles, 298–299
 - layout, 299
 - styles, 295–297
 - simulating pseudo-class state, 305–306
 - debugging JavaScript code, 627–648
 - common errors, 644–646
 - with Console, 632–634
 - displaying in various browsers, 632–633
 - executing code in, 633–634
 - logging data to, 633
 - error messages, 647–648
 - error types, 628–629
 - logic errors, 629
 - runtime errors, 628–629
 - syntax errors, 628
 - monitoring script values, 640–643
 - pausing code, 634–637
 - stepping through code, 637–640
 - strategies for, 643–644
 - tools for, 630–632
 - debugging tool tab, 632

- decision-making, 462–466
 - multi-block statements, 463
 - AND operator (&&), 462
 - OR operator (||), 462
 - switch statement, 464–466
- declaration block, 27
- declarations, 26–27
 - adding inline declaration to element, 302
 - adding to inspector stylesheet, 302–304
 - declaration types, 310–311
 - disabling, 301
 - weight and, 311–313
- declaring
 - arrays, 554–555
 - JSON variables, 620–621
 - variables
 - with `const`, 414–415
 - with `let`, 412–413
- decrement operator (`--`), 433
- dedicated servers, 51
- default font size, 768
- `:default` pseudo-class, 265
- deletion, 580
- descendant combinator, 244
- descenders, 372
- design software, 44
- devices, 767
- `<dfn>` tag, 113–114
- Dijkstra, Edsger, 630
- dimensions, box model, 228–233
 - converting inline element to block-level element, 233
 - minimum or maximum height or width, 231–232
- direction, `flex`, 706–709
- directory, 54–55
- `:disabled` pseudo-class, 266
- `display` property, 209–210
- `display` type, 209–210

- `<div>` tag, 94
- division operator (`/`), 434–435
- `<dl>` tag, 114–115
- Doctype declaration, 19
- `<!DOCTYPE html>` tag, 18–19
- document events, 547
- document object, 37
- `document.write()` statement, 481
- DOM (Document Object Model)
 - overview, 522–525
 - traversing, 529–535
 - getting children of parent element, 530–534
 - getting parent of child element, 534
 - getting siblings of element, 534–535
- domain name, 50–51
- dot (`.`), 240–242, 516
- double colon (`::`), 278
- double quotation marks (`"`), 421
- double-slash (`//`), 408
- `do...while` loops, 474–475
- drop shadow
 - on buttons, 168
 - for text, 345
- dynamic viewport, 777
- dynamism
 - `<script>` tag, 34–35
 - defined, 8, 33
 - displaying message to user, 35–36
 - writing text to page, 37

E

- ECMAScript, 64
- ECMAScript 6, 407
- e-commerce, 52
- Edge, 40, 60, 293, 631

- editable elements, 267
- edit-save-reload workflow, 12–14, 40
- elements, 26
 - adding class to, 304–305
 - adding inline declaration to, 302
 - adding to pages, 535–538
 - defined, 15–17
 - floating, 677–685
 - clearing, 681–682
 - container collapse, 683–685
 - creating pull quote, 679–680
 - inserting text or HTML into, 538–539
 - inspecting, 294–299
 - box model, 297–298
 - computed styles, 298–299
 - layout, 299
 - styles, 295–297
 - positioning, 685–695
 - absolute positioning, 688–690
 - fixed positioning, 690–692
 - relative positioning, 687–688
 - sticky positioning, 692–695
 - removing, 539
 - specifying, 525–529
 - by class name, 526, 527
 - collections of elements, 527–529
 - `id` attribute, 525
 - by tag name, 526
 - stacking, 696–701
 - layering elements with `z-index` property, 696–698
 - stacking context, 698–701
- tables, HTML, 149–154
 - captions, 150–151
 - header column, 151–152

- header row, 149–150
- table footer, 152–154
- em dash (–), 108–109
- em measurement unit, 324, 325–329
- tag, 98
- email addresses, 51
- email type, 169
- embedding internal style sheets, 28–31
- emphasizing text, 16–17, 98–99
- :enabled pseudo-class, 266
- end tag, 16
- endsWith() method, 588–589
- entity names, 82, 109
- equality operator (==), 439–440, 646
- equals sign (=), 413
- errors, 628–629
 - common, 644–646
 - error messages, 647–648
 - logic errors, 629
 - runtime errors, 628–629
 - syntax errors, 628
- escape sequences, 422
- ESLint, 74
- events
 - calling function in response to, 487–488
 - defined, 547
 - form events, 664–668
 - blur events, 666–667
 - change event, 667–668
 - focus() method, 665
 - monitoring focus events, 666
 - getting data about, 549–551
 - preventing default event action, 551–552
 - types of, 547–549
- every() method, 562–563

- ex (x-height), 324
- explicit grid, 741–745
- explicit label, 192
- exponential notation, 420
- expressions, 427–455
 - comparison expressions, 439–447
 - comparison operators, 439
 - data conversion, 442–443
 - equality operator (==), 439–440
 - greater than or equal operator (>=), 442
 - greater-than symbol (>), 441
 - inequality operator (!=), 440–441
 - less than or equal operator (<=), 442
 - strict equality operator (===), 444
 - strict inequality operator (!==), 444
 - ternary operator (? :), 446–447
 - using strings in, 445–446
- logical expressions, 447–451
 - logical operators, 447
 - NOT operator (!), 449
 - AND operator (&&), 447–448, 450–451
 - OR operator (||), 448–449, 450–451
- numeric expressions, 429–436
 - addition operator (+), 430
 - arithmetic assignment operators, 436
 - arithmetic operators, 429–430
 - decrement (--) operator, 433
 - division (/) operator, 434–435

- increment (++) operator, 430–431
- modulus (%) operator, 435
- multiplication (*) operator, 433
- subtraction and negation (–) operators, 431–432
- order of precedence, 452–455
- string expressions, 436–438
- structure of, 428–429
- external JavaScript files, 408–410
- external style sheet, 31–32
- eye, 371

F

- fat arrow, 499
- faux small caps, 391
- field type, 651–652
- <fieldset> tag, 173
- <figcaption> tag, 127–128
- figures, 127–128
- file formats
 - extensions, 12
 - fonts, 373
 - web media, 133
- file picker, 179
- filenames, 12
- find() method, 568–569
- Firefox, 40, 60, 293, 630
- :first-child pseudo-class, 255–256
- ::first-letter pseudo-element, 283–284
- ::first-line pseudo-element, 285–286
- :first-of-type pseudo-class, 261
- fit-content keyword, 229
- fixed positioning, 690–692
- fixed-width layout, 766

- flex containers, 705–719
 - adding gaps between items, 718–719
 - aligning flex items along primary axis, 709–710
 - aligning flex items along secondary axis, 711–712
 - aligning rows or columns along secondary axis, 716–718
 - centering element horizontally and vertically, 713–714
 - flex direction, 706–709
 - overview, 706
 - setting up, 706
 - wrapping flex items, 714–716
- flex items, 719–735
 - changing order of, 734–735
 - flex shorthand property, 730–731
 - growing, 720–722
 - laying out content columns, 731–734
 - shrinking, 722–726
 - suggesting size, 726–730
- flex shorthand property, 730–731
- flex-basis property, 726
- Flexbox, 299, 703–735
 - CSS Grid and, 705
 - flex containers, 705–719
 - adding gaps between items, 718–719
 - aligning flex items along primary axis, 709–710
 - aligning flex items along secondary axis, 711–712
 - aligning rows or columns along secondary axis, 716–718
 - centering element horizontally and vertically, 713–714
 - flex direction, 706–709
 - overview, 706
 - setting up, 706
 - wrapping flex items, 714–716
 - flex items, 719–735
 - changing order of, 734–735
 - flex shorthand property, 730–731
 - growing, 720–722
 - laying out content columns, 731–734
 - shrinking, 722–726
 - suggesting size, 726–730
 - flex shorthand property, 730–731
 - flex-basis property, 726
- Fluid layouts, 769–779
 - cclamp() function, 779
 - with CSS Grid, 773–776
 - with Flexbox, 770–773
 - making typography fluid, 778
 - viewport units, 776–778
- focus event, 666
- focus() method, 665
 - :focus pseudo-class, 269
- Font Playground, 387
- font shorthand property, 392
- Font Squirrel, 373
- font stack, 365–366
- @font-face at-rule, 374
- fonts, 325–326. *see also* typography
 - <footer> tag, 93
 - for . . . loop, 634
 - for loops, 470–474, 529
 - forcing inheritance, 308–309
 - forEach() method, 560–561
- foreground, 346
- forgiving selector lists, 272
- form events, 548
- <form> tag, 167
- formats
 - images, 124–125
 - web audio, 139
 - web videos, 133–134
- forms, 163–180
 - building, 166–180
 - adding form buttons, 167–168
 - checkboxes, 171–173
 - pickers, 178–180
 - radio buttons, 173–175
 - selection lists, 175–177
 - setting up form, 167
 - text-based fields, 168–171
- flow, JavaScript, 457–480
 - code looping, 466–480
 - avoiding infinite loops, 480
 - controlling loop execution, 476–479
 - do . . . while loops, 474–475
 - for loops, 470–474
 - while loops, 467–470
 - if . . . else statements, 460–461
 - making multiple decisions, 462–466
 - multi-block statements, 463
 - AND operator (&&), 462
 - OR operator (||), 462
 - switch statement, 464–466
- fluid layouts, 769–779
 - cclamp() function, 779
 - with CSS Grid, 773–776
 - with Flexbox, 770–773
 - making typography fluid, 778
 - viewport units, 776–778
- focus event, 666
- focus() method, 665
 - :focus pseudo-class, 269
- Font Playground, 387
- font shorthand property, 392
- Font Squirrel, 373
- font stack, 365–366
- @font-face at-rule, 374
- fonts, 325–326. *see also* typography
 - <footer> tag, 93
 - for . . . loop, 634
 - for loops, 470–474, 529
 - forcing inheritance, 308–309
 - forEach() method, 560–561
- foreground, 346
- forgiving selector lists, 272
- form events, 548
- <form> tag, 167
- formats
 - images, 124–125
 - web audio, 139
 - web videos, 133–134
- forms, 163–180
 - building, 166–180
 - adding form buttons, 167–168
 - checkboxes, 171–173
 - pickers, 178–180
 - radio buttons, 173–175
 - selection lists, 175–177
 - setting up form, 167
 - text-based fields, 168–171

- Local Lane, 166
- overview, 163–165
- processing form data, 649–672
 - coding checkboxes, 654–656
 - form events, 664–668
- HTMLFormElement object, 650
- HTMLInputElement object, 650
- keyboard shortcuts, 668–670
- programming text fields, 651–653
- radio buttons, 657–660
- selection lists, 660–664
- pseudo-classes targeting state, 264–267
- Server Street, 165
- web accessibility with, 192
- forms property, 524
- for . . . of loop, 561–562
- fr unit, 742–743, 773
- FTP client, 45, 56
- FTP support, 52
- function (local) scope, 505–506
- functional pseudo-classes, 271–278
 - :has(), 276–278
 - :is(), 272–273
 - :not(), 273–275
 - selector lists, 272
 - :where(), 275
- functions, 36, 481–512
 - anonymous functions, 495–498
 - assigning to variables, 496–497
 - replacing function call with, 497–498
 - arrow functions, 498–500

- calling, 484–488
 - after page is loaded, 486–487
 - parsed `<script>` tag, 484–485
 - in response to events, 487–488
- defined, 482
- passing values to, 488–493
 - making argument optional, 492–493
 - multiple values, 490–491
 - single value, 489–490
- recursive functions, 508–510
- returning values from, 494–495
- running in future, 500–504
 - interval, 502–504
 - timeout, 501–502
- structure of, 482–483
- variable scope, 504–507
 - block scope, 504–505
 - function scope, 505–506
 - global scope, 507

G

- gap shorthand property, 749
- generic font families, 366–367
- getAttribute() method, 545
- getDate() method, 602
- getDay() method, 602, 604
- getFullYear() method, 602
- getHours() method, 602
- getItem() method, 625
- getMilliseconds() method, 602
- getMinutes() method, 602
- getMonth() method, 602, 604
- getSeconds() method, 602
- getTime() method, 602

- GIF (Graphics Interchange Format), 124
- GIPHY, 124
- GitHub, 4
- GitHub Pages, 49
- global (page-level) scope, 507
- global variables, 646
- Google Chrome, 40, 60, 292, 630
- Google Fonts, 368–370, 373, 387–388
- gradients, 354–358
 - conic gradient, 358
 - linear gradient, 354–355
 - radial gradient, 356–357
- Graphics Interchange Format (GIF), 124
- greater than or equal operator (`>=`), 442
- greater-than symbol (`>`), 82, 109, 245, 441
- grid
 - alignment properties, 761–764
 - assigning grid items, 750–761
 - to columns and rows, 750–757
 - to named grid areas, 757–761
 - overview, 738
 - specifying rows and columns, 740–750
 - explicit grid, 741–745
 - gutters, 749–750
 - implicit grid, 745–749
 - grid cell, 750
 - grid container, 739–740
 - grid-template-columns property, 746–748
 - grid-template-rows property, 745–746
 - growing flex items, 720–722
 - gutters, 718–719, 749–750

H

`<h1>` tag, 25, 83–84, 187–188

hard disk, 55–56

`:has()` functional pseudo-class, 276–278, 317

hashtag symbol (#), 104, 242–243

head property, 524

head section, 19

`<head>` tag, 20

header column, 151–152

header row, 149–150

`<header>` tag, 88–89

headings, 83–85, 187–188

height, boxes, 229

height attribute, 135

hexadecimal code, 340–341

hexadecimal integer values, 421

hh argument, 600

hidden type, 170

hierarchies, object, 523–524

Hopper, Grace, 630

horizontal alignment, 154–157

horizontal radius, 222

horizontal rule, 17–18, 86–87

`:hover` pseudo-class, 269–270

`<hr>` tag, 17–18, 86–87

href (hypertext reference), 15, 102

hs1() function, 341–343, 790

HTML (HyperText Markup Language), 79–202

adding structure with, 14–24

elements and tags, 15–17

style vs., 23–24

tag attributes, 17–18

text, 21–23

web page title, 20–21

basics of web pages, 10–14

launching new text file, 11

saving HTML files, 11–12

test driving web pages, 12–14

building lists, 105–108

bulleted lists, 105–107

numbered lists, 107–108

commenting code, 95–96

creating links, 102–104

internal links, 103–104

overview, 102–103

embedding audio snippets, 137–141

own audio, 139–141

third-party audio, 137–139

embedding videos, 131–137

own videos, 132–137

third-party videos, 131–132

forms, 163–180

building, 166–180

Local Lane, 166

overview, 163–165

Server Street, 165

general discussion of, 8

inserting images, 124–130

figures with captions, 127–128

image formats, 124–125

overview, 125–126

turning images into link, 128

vertical alignment, 129–130

inserting into element, 538–539

inserting special characters, 108–109

reasons to learn, 8–9

semantic page tags, 87–95

`<article>` tag, 91

`<aside>` tag, 92–93

`<div>` tag, 94

`<footer>` tag, 93

`<header>` tag, 88–89

`<main>` tag, 90–91

`<nav>` tag, 89–90

`<section>` tag, 91–92

`` tag, 95

structure tags, 80–87

headings, 83–85

horizontal rule, 86–87

line breaks, 85–86

paragraphs, 81–82

tables, 143–162

adding more rows, 148

elements, 149–154

one-row table, 146–147

overview, 143–145

styling, 154–162

text tags, 98–101, 109–121

`<abbr>` tag, 110

adding quotations, 100–101

`<address>` tag, 110–111

`<cite>` tag, 111–112

`<code>` tag, 112–113

`<dfn>` tag, 113–114

`<dl>` tag, 114–115

emphasizing text, 98–99

`<kbd>` tag, 116

`<mark>` tag, 116–117

marking important text, 99

nesting tags, 100

`<pre>` tag, 117–118

`<s>` tag, 118–120

`<sub>` tag, 120

`<sup>` tag, 120–121

tweaking attributes with JavaScript, 544–546

validating code, 62–63, 65–69

online, 65–67

in text editor, 67–69

web accessibility, 181–202

accessibility semantics, 191–198

assistive technology, 185–187

color contrast, 200–202

keyboard support, 198–200

- media, 189–191
- page structure, 187–188
- reasons for, 182–183
- text, 188–189
- validating, 202
- who needs accessibility, 184–185

`<html lang="en">` tag, 19

HTML viewer tab, 631

HTMLFormElement object, 650

HTMLInputElement object, 650

HTMLOptionElement object, 661

HTMLSelectElement object, 660–661

HyperText Markup Language.
see HTML

hypertext reference (`href`), 15, 102

hyphen (–), 104

hyphenating text, 395

hypothetical main size, 726

I

`<i>` tag, 98–99

iconoclasts, 405

`id` attribute, 525

`id` selector, 242–243

identity operator (`===`), 444, 646

`if . . . else` statements, 460–461

`if` statements, 458–460

`<iframe>` tag, 132

images, 124–130

- clearing floats for, 681–682
- figures with captions, 127–128
- image formats, 124–125
- mobile-first design, 797
- overview, 125–126
- responsive layouts with, 791–794

- delivering images
 - responsively, 792–794
- making images
 - responsive, 792
- turning into links, 128
- vertical alignment, 129–130
- web accessibility, 190

images property, 525

`` tag, 125–126, 190, 274–275, 544

implicit grid, 740, 745–749

implicit label, 192

implicit return, 499–500

! important annotation, 310–311

`in` (inches), 323

`includes()` method, 588–589

increment operator (`++`), 430–431

indenting lines, 396, 643

index number, 554

`indexOf()` method, 589–590

inequality operator (`!=`), 440–441

infinite loops, 480

infinite recursion, 511–512

inheritance, 306–309

initial containing block, 332

inline boxes, 208, 233

inline dimension, 219

inline elements, 83, 676

inline quotations, 101

inline styles, 28

`innerHTML` property, 537–538

`<input>` tag, 167

`:in-range` pseudo-class, 266

insertion, 581

inspecting elements, 294–299

- box model, 297–298
- computed styles, 298–299
- layout, 299
- styles, 295–297

inspector stylesheet, 302–304

integers, 419, 612

interfaces, 522

internal links, 103–104

internal style sheet, 28–31

internet service provider (ISP), 48

interval, 502–504

`:invalid` pseudo-class, 266

`:is()` functional pseudo-class, 272–273, 275, 317

italics, 386

iterating arrays, 528, 559–569

- `every()` method, 562–563
- `find()` method, 568–569
- `forEach()` method, 560–561
- `for . . . of` loop, 561–562
- `map()` method, 565–566
- `reduce()` method, 566–567
- `some()` method, 564–565

J

JavaScript, 32–37, 399–672

- adding comments to code, 408
- arrays, 553–582
 - declaring, 554–555
 - defined, 554
 - iterating, 559–569
 - manipulating, 570–582
 - multidimensional arrays, 569–570
 - populating with data, 555–559
- assumptions with, 405–407
 - browsers with JavaScript turned off, 405–407
 - handling very old browsers, 407
- basics of web pages, 10–14
 - launching new text file, 11
 - saving HTML files, 11–12
 - test driving web pages, 12–14

- JavaScript (*continued*)
 - controlling flow of, 457–480
 - code looping, 466–480
 - `if . . . else` statements, 460–461
 - making multiple decisions, 462–466
 - true/false decisions with `if` statements, 458–460
 - creating external JavaScript files, 408–410
 - dates and times, 599–611
 - debugging code, 627–648
 - common errors, 644–646
 - with Console, 632–634
 - error messages, 647–648
 - error types, 628–629
 - monitoring script values, 640–643
 - pausing code, 634–637
 - stepping through code, 637–640
 - strategies for, 643–644
 - tools for, 630–632
 - defined, 33
 - difficulty of, 402
 - displaying message to user, 35–36
 - expressions, 427–455
 - comparison expressions, 439–447
 - logical expressions, 447–451
 - numeric expressions, 429–436
 - order of precedence, 452–455
 - string expressions, 436–438
 - structure of, 428–429
 - functions, 481–512
 - anonymous functions, 495–498
 - arrow functions, 498–500
 - calling, 484–488
 - defined, 482
 - passing values to, 488–493
 - recursive functions, 508–510
 - returning values from, 494–495
 - running in future, 500–504
 - structure of, 482–483
 - variable scope, 504–507
 - general discussion of, 8
 - inabilities of, 404
 - Math object, 611–615
 - converting between strings and numbers, 612–614
 - properties and methods, 614–615
 - objects, 513–552
 - building reactive pages with events, 546–552
 - creating, 519–521
 - defined, 514–515
 - Document Object Model, 522–525
 - manipulating elements, 535–539
 - manipulating properties of, 515–518
 - modifying CSS with JavaScript, 540–544
 - object methods, 518–519
 - specifying elements, 525–529
 - traversing DOM, 529–535
 - tweaking HTML attributes with JavaScript, 544–546
 - Web APIs, 521–522
- processing form data, 649–672
 - coding checkboxes, 654–656
 - form events, 664–668
- HTMLFormElement object, 650
- HTMLInputElement object, 650
- keyboard shortcuts, 668–670
- programming text fields, 651–653
- radio buttons, 657–660
- selection lists, 660–664
- programming language, 401–402
- reasons for using, 403
- reasons to learn, 8–9
- requirements for, 404–405
- `<script>` tag, 34–35
- storing user data in browser, 617–626
- strings, 583–599
 - determining length of, 586–587
 - methods that extract substrings, 591–599
 - searching for substrings, 587–590
 - string templates, 584–586
- validating code, 64, 72–75
 - online, 73
 - in text editor, 74–75
- variables, 411–425
 - declaring with `const`, 414–415
 - declaring with `let`, 412–413
 - HTML keywords and, 424–425
 - literal data types, 419–423
 - naming, 417–419
 - reserved words, 423–424
 - storing value in, 413–414
 - using in statements, 415–417
- web coding, 400
- writing text to page, 37
- `join()` method, 573–574
- JPEG (Joint Photographic Experts Group), 124
- JSHint, 73

- JSON (JavaScript Object Notation), 618–623
 - converting JavaScript object to, 621–622
 - converting JSON string to JavaScript object, 622–623
 - declaring and using variables, 620–621
 - syntax for, 619–620
- JSONLint validation tool, 621
- `justify-content` property, 716–717, 762
- justifying text, 395
- `justify-items` property, 762

K

- `<kbd>` tag, 116
- kerning, 391
- keyboard events, 548
- keyboard shortcuts, 668–670
- keyboard support, 198–200
 - adding element to tab order, 199
 - removing element from tab order, 200
- keywords
 - defined, 99
 - JavaScript, 401
 - variables and HTML keywords, 424–425

L

- `<label>` tag, 170–171, 192
- LambdaTest, 62
- landmark ARIA roles, 194
- landmarks, 188
- language, 15
- large viewport, 777
- `:last-child` pseudo-class, 256–257

- `lastIndexOf()` method, 589–590
- `lastModified` property, 525
- `:last-of-type` pseudo-class, 262
- layering elements, 696–698
- layouts, 675–800
 - with CSS Grid, 737–764
 - alignment properties, 761–764
 - assigning grid items, 750–761
 - overview, 738
 - setting up grid container, 739–740
 - specifying grid rows and columns, 740–750
- Flexbox, 703–735
 - CSS Grid and, 705
 - flex containers, 705–719
 - flex items, 719–735
 - overview, 704
- floating elements, 677–685
 - clearing, 681–682
 - container collapse, 683–685
 - creating pull quote, 679–680
- inspecting, 299
- page flow, 676–677
- positioning elements, 685–695
 - absolute positioning, 688–690
 - fixed positioning, 690–692
 - relative positioning, 687–688
 - sticky positioning, 692–695
- responsive layouts, 765–800
 - adaptive layouts, 779–791
 - defined, 766–769
 - fluid layouts, 769–779
 - with images, 791–794

- mobile-first web development, 795–800
- stacking elements, 696–701
 - layering elements with `z-index` property, 696–698
 - stacking context, 698–701
- `<legend>` tag, 173
- length, string, 586–587
- length property, 570–571
- less than or equal operator (`<=`), 442
- less-than symbol (`<`), 82, 109
- `let` keyword, 412–413
- letter-spacing property, 390
- `` tag, 105–106
- licensing, typeface, 373
- ligatures, 391
- line breaks, 85–86
- line height, 376, 380–382, 384
- line length, 376, 382–384
- line numbers, 43, 753–757
- linear gradient, 354–355
- lines, grid, 751–753
- `:link` pseudo-class, 271
- `<link>` tag, 31–32, 318
- links, 102–104
 - internal links, 103–104
 - overview, 102–103
 - pseudo-classes, 270–271
 - styling, 392–393
 - turning images into, 128
 - to web font families, 368–370
- links property, 525
- linters, 64, 74
- Linux (Unix) computers, 56
- listener, event, 547
- lists, 105–108
 - bulleted lists, 105–107
 - numbered lists, 107–108

- literal data types, 419–423
 - Boolean literals, 423
 - numeric literals, 419–421
 - string literals, 421–423
- Living Standard, 62
- load-time errors, 628
- local (function) scope, 505–506
- local folders, 40–42
 - root folder, 41
 - subfolders, 41–42
- Local Lane route, 166
- local variables, 646
- localStorage, 618
- location property, 525
- logic errors, 629
- logical expressions, 447–451
 - logical operators, 447
 - NOT operator (!), 449
 - AND operator (&&), 447–448, 450–451
 - OR operator (||), 448–449, 450–451
- logical operators, 447
- logical properties, 219–220
- loop attribute, 135, 140
- loops and looping, 466–480
 - avoiding infinite loops, 480
 - controlling loop execution, 476–479
 - do...while loops, 474–475
 - for loops, 470–474
 - populating arrays using, 558–559
 - while loops, 467–470
- lowercase letters, 19

M

- Macs, 630
 - TextEdit on, 11
 - web browsers, 13
 - web development tools for, 292

- <main> tag, 90–91
- makeDate callback function, 607
- map() method, 565–566
- margin property, 224–225
- margins, box model, 207, 223–228
 - collapsing, 225–228
 - negative margins, 225
- <mark> tag, 116–117
- markup, 15
- Markup Validation Service, 65–66
- matches-none pseudo-class, 273–275
- Math object, 611–615
 - converting between strings and numbers, 612–614
 - addition operator (+), 613–614
 - parseFloat() function, 613
 - parseInt() function, 612–613
 - properties and methods, 614–615
- max() function, 335
- max-content keyword, 229
- max-height property, 231–232
- max-width property, 231–232
- measurement units, 321–333
 - absolute, 323–324
 - container query units, 789
 - relative, 324–331
 - em measurement unit, 325–329
 - rem unit, 329–331
 - using percentages, 325
 - viewport, 332–333
- @media at-rule, 780
- media queries, 780–783
- messages, 35–36
- <meta> tag, 20, 108, 798
- methods, 36

- Microsoft Edge, 40, 60, 293, 631
- middle value, 129
- min() function, 334–335
- min-content keyword, 229
- minmax() function, 744
- minus sign (–), 420
- Mixcloud, 138
- mm (millimeters), 323
- mm argument, 600
- mobile-first CSS, 798
- mobile-first web development, 795–800
 - defined, 795–796
 - mobile-first CSS, 798
 - non-mobile breakpoint, 799–800
 - prioritizing in, 796–797
 - scaling content, 798
 - testing width, 797
- Modern Font Stacks, 368
- modular scale, 379–380
- modules, 62
- modulus operator (%), 324, 435
- month argument, 600
- month picker, 179–180
- motor disabilities, 187
- mouse events, 548
- Mozilla Firefox, 40, 60, 293, 630
- MP3, 139
- MPEG-4, 133, 139
- ms argument, 600
- mth argument, 600
- multi-block statements, 463
- multidimensional arrays, 569–570
- multiplication operator (*), 433

N

- named grid areas, 758–761
- namespace, 495

- naming variables, 417–419
- narrow selectors, 313
- nav element, 188
- <nav> tag, 89–90
- negative margins, box model, 225, 228
- nested blocks, 643
- nesting tags, 100
- Netlify, 49
- network performance, 768
- network tab, 632
- next sibling, 534
- next-sibling combinator, 246–247
- non-identity operator, 444
- non-mobile breakpoint, 799–800
- Normalize.css, 291
- <noscript> tag, 405
- :not() functional pseudo-class, 273–275, 317
- NOT operator (!), 449
- Notepad, 11, 42
- Notepad++, 44
- Nova, 44, 56
- no-width layout, 766
- :nth-child() pseudo-class, 257–259
- :nth-last-child() pseudo-class, 259
- :nth-last-of-type() pseudo-class, 263
- :nth-of-type() pseudo-class, 262–263
- null string, 421
- number type, 169
- numbered lists, 107–108
- numbers, converting between strings and, 612–614
- numeric expressions, 429–436
 - addition operator (+), 430
 - arithmetic assignment operators, 436

- arithmetic operators, 429–430
- decrement (--) operator, 433
- division (/) operator, 434–435
- increment (++) operator, 430–431
- modulus (%) operator, 435
- multiplication (*) operator, 433
- subtraction and negation (-) operators, 431–432
- numeric literals, 419–421
- numeric styles, 391–392

O

- object literal, 520
- object methods, 518–519
- objects, 513–552
 - building reactive pages with events, 546–552
 - event types, 547–549
 - events defined, 547
 - getting data about event, 549–551
 - preventing default event action, 551–552
 - creating, 519–521
 - defined, 514–515
 - Document Object Model, 522–525
 - elements
 - manipulating, 535–539
 - specifying, 525–529
 - manipulating properties of, 515–518
 - changing value of property, 517–518
 - referencing property, 516–517
 - some properties are objects, 517
- modifying CSS with JavaScript, 540–544
 - adding class to element, 541–544
 - changing element styles, 540–541
- object methods, 518–519
- traversing DOM, 529–535
 - getting children of parent element, 530–534
 - getting parent of child element, 534
 - getting siblings of element, 534–535
- tweaking HTML attributes with JavaScript, 544–546
- Web APIs, 521–522

- oblique value, 386
- Ogg, 133, 139
- tag, 107
- one-directional containers, 706
- online resources, 4
 - cross-browser testing sites, 62
 - text editors, 44
 - web hosts, 53
- online validation
 - CSS, 69–71
 - HTML, 65–67
 - JavaScript, 73
- :only-child pseudo-class, 260
- :only-of-type pseudo-class, 263–264
- open source typefaces, 373
- Opera, 60
- operands, 428
- operating system, 51
- <option> tag, 176
- :optional pseudo-class, 266
- OR operator (||), 448–449, 450–451, 462

order of precedence, 452–455
order property, 734–735
origin types, 311
outline property, 218
:out-of-range
 pseudo-class, 266

P

<p> tag, 22, 81–82
padded cells, 161–162
padding, box model, 207,
 213–215
padding property, 161,
 214–215
page flow, 676–677
page footers, 93
page headers, 88–89
page-level (global) scope, 507
paragraphs, 81–82
parent elements, 530–534
parentheses, 244, 455, 645
parse() method, 622
parseFloat() function, 613
parseInt() function, 612–613
parsing, 484–485
password type, 169
pausing code, 634–637
pc (picas), 323
percentages, 229
 for margins, 224
 for padding values, 214
 relative measurement
 units, 325
photo editors, 44
pickers, 178–180
plus sign (+), 246–247, 420
PNG (Portable Network
 Graphics), 125
polluting namespace, 496
pop() method, 574–575
position property, 686
positioning elements, 685–695
 absolute positioning,
 688–690
 fixed positioning, 690–692
 relative positioning, 687–688
 sticky positioning, 692–695
post-decrement form, 434
poster attribute, 134
post-increment operator, 432
<pre> tag, 117–118
pre-decrement form, 434
pre-increment operator, 432
previews, code, 43
previous sibling, 534
primary axis
 aligning flex items along,
 709–710
 defined, 706
 shrinking flex items and,
 725
programming language, 33,
 401–402
progressive enhancement, 796
prompt() method, 416–417
properties, 515–518
 changing value of, 517–518
 CSS, 26–27
 editing value, 299–301
 objects as, 517
 referencing, 516–517
property access operator, 516
providers, web hosts, 48–49
 finding free hosting provider,
 48–49
 signing up with commercial
 hosting provider, 49
 using existing internet
 provider, 48
pseudo-classes, 251–278
 child-related pseudo-
 elements, 255–260
 : first-child pseudo-
 class, 255–256
 : last-child pseudo-class,
 256–257
 : nth-child() pseudo-
 class, 257–259
 : nth-last-child()
 pseudo-class, 259
 : only-child pseudo-
 class, 260
 form element state pseudo-
 classes, 264–267
 functional pseudo-classes,
 271–278
 : has(), 276–278
 : is(), 272–273
 : not(), 273–275
 selector lists, 272
 : where(), 275
 matching elements by user
 action, 268–270
 : active pseudo-class,
 268–269
 : focus pseudo-class, 269
 : hover pseudo-class,
 269–270
 matching links, 270–271
 overview, 251–253
 simulating pseudo-class
 state, 305–306
 styling elements with,
 253–255
 type-related pseudo-classes,
 260–264
 : first-of-type
 pseudo-class, 261
 : last-of-type
 pseudo-class, 262
 : nth-last-of-type()
 pseudo-class, 263
 : nth-of-type()
 pseudo-class, 262–263
 : only-of-type
 pseudo-class, 263–264

- pseudo-elements, 278–286
 - ::before, 281–283
 - ::after, 279–281
 - ::first-letter, 283–284
 - ::first-line, 285–286
 - working with, 278–279
- pt (points), 323
- pull quote, 679–680
- push() method, 575–576
- px (pixels), 323–324

Q

- Q (quarter-milimeter), 323
- queries, 779–791
 - container queries, 783–789
 - media queries, 780–783
 - user preference queries, 789–791
- quotations, 100–101, 584–585, 645

R

- radial gradient, 356–357
- radio buttons, 657–660
 - adding in forms, 173–175
 - radio button state, 658–660
 - referencing, 657–658
- range syntax, 783
- RE (regressive enhancement), 795
- reactive pages, 546–552
 - event types, 547–549
 - events defined, 547
 - getting data about event, 549–551
 - preventing default event action, 551–552
- read only properties, 517–518
- :read-only pseudo-class, 266
- :read-write pseudo-class, 267

- recursive functions, 508–510
- reduce() method, 566–567
- referencing
 - checkboxes, 654
 - properties, 516–517
 - radio buttons, 657–658
 - selection list options, 662
 - text fields by field type, 651–652
- regressive enhancement (RE), 795
- relative measurement units, 324–331
 - em measurement unit, 325–329
 - rem unit, 329–331
 - using percentages, 325
- relative positioning, 687–688
- rem unit, 324, 329–331
- remove() method, 542–543
- removeAttribute() method, 546
- removeItem() method, 626
- repeat() function, 743–744
- replacement, 581
- :required pseudo-class, 267
- reserved words, 401, 423–424
- resets, CSS, 290–291
- responsive layouts, 765–800
 - adaptive layouts, 779–791
 - container queries, 783–789
 - media queries, 780–783
 - user preference queries, 789–791
- defined, 766–769
- fluid layouts, 769–779
 - clamp() function, 779
 - with CSS Grid, 773–776
 - with Flexbox, 770–773
 - making typography fluid, 778
- viewport units, 776–778

- with images, 791–794
 - delivering images responsively, 792–794
 - making images responsive, 792
- mobile-first web development, 795–800
 - defined, 795–796
 - mobile-first CSS, 798
 - non-mobile breakpoint, 799–800
 - prioritizing in, 796–797
 - scaling content, 798
 - testing width, 797
- rest parameters, 492
- reverse() method, 576
- RGB codes, 340–341
- rgb() function, 339–340
- root directory, 53
- root folder, 41
- :root pseudo-class, 253
- rounded corners, on buttons, 168
- rounding borders, box model, 218–223
- rows, table
 - adding more, 148
 - one-row table, 146–147
- royalties, 134
- rules, CSS, 26–27
- runtime errors, 628–629

S

- <s> tag, 118–120
- Safari, 40, 60, 293, 631
- Samsung Internet, 60
- sans-serif typefaces, 364–365
- Sauce Labs, 62
- saving HTML files, 11–12
- scalability, 52
- Scalable Vector Graphics (SVG), 125

- scaling content, 798
- screen magnifier, 186
- screen orientation, 768
- screen readers, 186
- <script> tag
 - calling function when tag is parsed, 484–485
 - creating external JavaScript files, 408–410
 - overview, 34–35
- search type, 170
- secondary axis
 - aligning flex items along, 711–712
 - aligning rows or columns along, 716–718
 - defined, 706
- section structure ARIA roles, 194–195
- <section> tag, 91–92
- <select> tag, 176
- selection lists, 175–177, 660–664
 - changing selected option, 664
 - getting selected list option, 662–663
 - HTMLOptionElement object, 661
 - HTMLSelectElement object, 660–661
 - referencing selection list options, 662
- selector lists, 239, 272
- selectors, 235–249
 - attribute selectors, 247–249
 - combinators, 243–247
 - child combinator, 245
 - descendant combinator, 244
 - next-sibling combinator, 246–247
 - subsequent-sibling combinator, 245–246
 - defined, 209
 - examples of, 238–239
 - selector lists, 239
 - specifying elements by, 527
 - standard selectors, 240–243
 - class selector, 240–242
 - id selector, 242–243
 - type selector, 240
 - universal selector, 243
 - web page relationships, 236–238
- self-hosting font files, 372–375
- semantic page tags, 87–95
 - <article> tag, 91
 - <aside> tag, 92–93
 - differentiating, 197–198
 - <div> tag, 94
 - <footer> tag, 93
 - <header> tag, 88–89
 - <main> tag, 90–91
 - <nav> tag, 89–90
 - <section> tag, 91–92
 - tag, 95
 - for web accessibility, 188
- semicolon (;), 36, 109
- serif typefaces, 364–365
- Server Street route, 165
- servers, 48
- server-side JavaScript tools, 404
- sessionStorage, 618
- setAttribute() method, 545
- setDate() method, 605
- setFullYear() method, 605
- setHours() method, 605
- setInterval() method, 502–504
- setItem() method, 623
- setMilliseconds() method, 605
- setMinutes() method, 605
- setMonth() method, 605
- setSeconds() method, 605
- setTime() method, 605
- setTimeout() method, 501–502
- shadow, text, 345–346
- shared servers, 51
- sheets, 25–26
- shift() method, 576–577
- shrinking flex items, 722–726
- siblings, element, 534–535
- single quotation marks ('), 421
- size
 - flex items, 726–730
 - grid rows and columns, 742–743
 - type, 376–380, 384
- sizes attribute, 793
- slash (/), 17
- slice() method, 577–578, 593–594
- small caps, 390–391
- small viewport, 777
- some() method, 564–565
- sort() method, 578–580
- SoundCloud, 138
- source code order, 317–318
- <source> tag, 136
- spacing between letters, 390
- tag, 95
- special characters, HTML, 108–109
- special keys, 668
- specificity, 275, 313–317
- specifying
 - colors, CSS, 338–343
 - elements, 525–529
 - by class name, 526

- collections of elements, 527–529
- id attribute, 525
- by selector, 527
- by tag name, 526
- spell checking, 43
- splice() method, 580–582, 598–599
- split() method, 594–596
- spread operator (. . .), 572–573
- square brackets operator ([]), 662
- src attribute, 132, 134, 136, 140, 141, 409
- srcset attribute, 793
- ss argument, 600
- stacking elements, 696–701
 - layering elements with z-index property, 696–698
 - stacking context, 698–701
- standard selectors, 240–243
 - class selector, 240–242
 - id selector, 242–243
 - type selector, 240
 - universal selector, 243
- start tag, 16
- startsWith() method, 588–589
- statements, 36
 - breaking up long, 437
 - using variables in, 415–417
- static fonts, 387
- statistics, website, 52
- stepping through code, 637–640
- sticky positioning, 692–695
- storage space, 49–50
- strict equality operator (===), 444
- strict inequality operator (!==), 444
- string expressions, 436–438
- string literals, 421–423
- stringify() method, 621
- strings, 583–599
 - converting between numbers and, 612–614
 - addition operator (+), 613–614
 - parseFloat() function, 613
 - parseInt() function, 612–613
 - determining length of, 586–587
 - methods that extract substrings, 591–599
 - searching for substrings, 587–590
 - string templates, 584–586
 - using in comparison expressions, 445–446
- tag, 99
- structure
 - defined, 8
 - of expressions, 428–429
 - of functions, 482–483
 - style vs., 23–24
- structure tags, 80–87
 - headings, 83–85
 - horizontal rule, 86–87
 - line breaks, 85–86
 - paragraphs, 81–82
- <style> tag, 29–30, 318
- styles, 306–319
 - adding with CSS, 28–32
 - external style sheet, 31–32
 - inline styles, 28
 - internal style sheet, 28–31
 - borders, 216
 - cascade, 309–310
 - cascade algorithm, 318–319
 - declaration types, 310–311
 - defined, 8, 24–25
 - of elements, 540–541
 - inheritance, 306–309
 - inspecting, 295–297
 - origin types, 311
 - source code order, 317–318
 - specificity, 313–317
 - structure vs., 23–24
 - weight hierarchy, 311–313
- stylesheets, 289–290
- styling
 - elements with pseudo-classes, 253–255
 - tables with HTML, 154–162
 - aligning text within cells, 154–158
 - borders, 159–161
 - location of captions, 162
 - padded cells, 161–162
 - <sub> tag, 120
 - subfolders, 41–42
 - Sublime Text, 44
 - subsequent-sibling combinator, 245–246
 - substr() method, 596–597, 598–599
 - substring() method, 597–599
 - substrings
 - methods that extract, 591–599
 - searching for, 587–590
 - subtitle class, 241
 - subtraction and negation operators (–), 431–432
 - <sup> tag, 120–121
 - SVG (Scalable Vector Graphics), 125
 - switch statement, 464–466
 - syntax, 401
 - highlighting, 42
 - JavaScript errors, 628
 - JSON, 619–620
 - system font families, 367–368

T

tab order

- adding element to, 199
- removing element from, 200

`tabindex="0"` attribute, 668

table footer, 152–154

tables, 143–162

- adding more rows, 148
- elements, 149–154
 - captions, 150–151
 - header column, 151–152
 - header row, 149–150
 - table footer, 152–154
- one-row table, 146–147
- overview, 143–145
- styling, 154–162
 - aligning text within cells, 154–158
 - borders, 159–161
 - location of captions, 162
 - padded cells, 161–162

tabs, 22

tags

- attributes, 17–18
- defined, 15–17
- specifying elements by name of, 526

target, 270

`:target` pseudo-class, 271

`<td>` tag, 147

tech support, 52

`tel` type, 169

templates

- grid, 741
- string, 584–586

ternary operator (`?:`), 446–447

testing code, 59–75

- cross-browser testing, 60–62
 - browser market share, 60–61
- testing online, 61–62
- testing suite, 61

testing suite, 61

testing width, 797

text

- adding, 21–23
- adding color to, 344–346
 - color property, 344
- text decorations, 344
- text shadow, 345–346
- aligning within table cells, 154–158
- inserting into element, 538–539
- mobile-first design, 797
- web accessibility, 188–189
- writing to pages, 37

text decorations, 344

text editors, 10–11

- choosing, 42–45
- validating code
 - CSS, 72
 - HTML, 67–69
 - JavaScript, 74–75

text fields, 651–653

- getting value of, 652
- referencing by field type, 651–652
- setting value of, 653

text files, 11

text processing, 43

text styles, 384–389

- italics, 386
- text weight, 384–386
- variable fonts, 386–389

text tags, 98–101, 109–121

- `<abbr>` tag, 110
- adding quotations, 100–101
- `<address>` tag, 110–111
- `<cite>` tag, 111–112
- `<code>` tag, 112–113
- `<dfn>` tag, 113–114
- `<dl>` tag, 114–115
- emphasizing text, 98–99

`<kbd>` tag, 116

`<mark>` tag, 116–117

marking important text, 99

nesting tags, 100

`<pre>` tag, 117–118

`<s>` tag, 118–120

`<sub>` tag, 120

`<sup>` tag, 120–121

text type, 168

`<textarea>` tag, 170

text-based fields, 168–171

TextEdit, 11, 42

`text-transform` property, 389

`<tfoot>` tag, 152–153

`<th>` tag, 149, 151–152

tilde (`~`), 245–246

time, 601. *see also* dates and times

time type, 169

timeout, 501–502

title property, 525

`<title>` tag, 20–21

titles, web page, 20–21

top value, 130

Torvalds, Linus, 2

`<tr>` tag, 146–147

`<track>` tag, 191

tracks, grid, 751–753

traversing DOM, 529–535

- getting children of parent element, 530–534

- getting parent of child element, 534

- getting siblings of element, 534–535

true/false decisions, 458–460

truth table, 448, 449

two-directional containers, 706

type attribute, 136, 141, 168–170

type selector, 240

typefaces, 364–365

- type-related pseudo-classes, 260–264
 - :first-of-type pseudo-class, 261
 - :last-of-type pseudo-class, 262
 - :nth-last-of-type() pseudo-class, 263
 - :nth-of-type() pseudo-class, 262–263
 - :only-of-type pseudo-class, 263–264
- typography, 363–396
 - alignment, 394–396
 - aligning text, 394
 - hyphenating text, 395
 - indenting paragraph first line, 396
 - alternative numeric glyphs, 391–392
 - capitalization, 389
 - font shorthand property, 392
 - font stack, 365–366
 - generic font families, 366–367
 - line height, 376, 380–382, 384
 - line length, 376, 382–384
 - links, 392–393
 - making typography fluid, 778
 - small caps, 390–391
 - spacing between letters, 390
 - system font families, 367–368
 - text styles, 384–389
 - italics, 386
 - text weight, 384–386
 - variable fonts, 386–389
 - type size, 376–380, 384
 - typefaces, 364–365
 - web font families, 368–375

- linking to, 368–370
- self-hosting, 372–375

U

- tag, 105–106
- underscore (`_`), 104, 417
- unforgiving selector lists, 272
- Unicode string values, 445
- universal selector, 243
- Unix (Linux) computers, 56
- `unshift()` method, 582
- uploading
 - defined, 53
 - site files, 56–57
- uptime, 51
- URL fragment, 270
- URL property, 525
- `url` type, 169
- user action, 268–270
 - :active pseudo-class, 268–269
 - :focus pseudo-class, 269
 - :hover pseudo-class, 269–270
- user agent stylesheet, 290
- user preference queries, 789–791
- users, 35–36
- UX (user experience), 766

V

- :valid pseudo-class, 267
- validation, 62–75
 - CSS code, 69–72
 - online, 69–71
 - in text editor, 72
 - HTML code, 65–69
 - online, 65–67
 - in text editor, 67–69
 - JavaScript code, 72–75

- online, 73
 - in text editor, 74–75
- overview, 62–65
- web accessibility, 202
- value keyword, 209
- values
 - changing value of property, 517–518
 - monitoring script values, 640–643
 - passing to functions, 488–493
 - making argument optional, 492–493
 - multiple values, 490–491
 - single value, 489–490
 - returning from functions, 494–495
 - storing in variables, 413–414
- `var` keyword, 413
- variable fonts, 386–389
- Variable Fonts website, 387
- variable scope, 504–507
 - block scope, 504–505
 - function scope, 505–506
 - global scope, 507
- variables, 411–425
 - assigning anonymous function to, 496–497
 - declaring JSON variables, 620–621
 - declaring with `const`, 414–415
 - declaring with `let`, 412–413
 - HTML keywords and, 424–425
 - literal data types, 419–423
 - Boolean literals, 423
 - numeric literals, 419–421
 - string literals, 421–423

- variables (*continued*)
 - monitoring script values, 640–643
 - naming, 417–419
 - reserved words, 423–424
 - storing value in, 413–414
 - using in statements, 415–417
 - vertical alignment
 - cell text in tables, 157–158
 - for images, 129–130
 - vertical radius, 222
 - vertical-align property, 129
 - vh (viewport height), 332
 - <video> tag, 134–137
 - videos, 131–137
 - third-party videos, 131–132
 - <video> tag, 134–137
 - web accessibility, 189–191
 - web media file formats, 133
 - web video formats, 133–134
 - viewport height (vh), 332
 - viewport maximum (vmax), 332
 - viewport measurement units, 332–333, 776–778
 - viewport minimum (vmin), 332
 - viewport width (vw), 332
 - Vimeo, 131–132
 - :visited pseudo-class, 271
 - visual disabilities, 186, 386
 - vmax (viewport maximum), 332
 - vmin (viewport minimum), 332
 - VS (Visual Studio) Code, 44, 68–69, 72, 74
 - vw (viewport width), 332
- W**
- W3C (World Wide Web Consortium), 62
 - WAI (Web Accessibility Initiative), 192
 - Wakamai Fondue, 391
 - watch expression, 642–643
 - WAV format, 139
 - WAVE (Web Accessibility Evaluation Tool), 202
 - WCAG (Web Content Accessibility Guidelines), 361, 382
 - web accessibility, 181–202
 - accessibility semantics, 191–198
 - adding form field labels, 192
 - ARIA roles, 192–198
 - assistive technology, 185–187
 - for auditory disabilities, 186
 - for cognitive disabilities, 187
 - for motor disabilities, 187
 - for visual disabilities, 186
 - choosing accessible colors, 361–362
 - color contrast, 200–202
 - keyboard support, 198–200
 - adding element to tab order, 199
 - removing element from tab order, 200
 - media, 189–191
 - page structure, 187–188
 - using headings hierarchically, 187–188
 - using semantic sectioning elements, 188
 - reasons for, 182–183
 - benefits of, 183
 - human right, 182
 - text, 188–189
 - validating, 202
 - who needs accessibility, 184–185
 - Web Accessibility Evaluation Tool (WAVE), 202
 - Web Accessibility In Mind (WebAIM) Contrast Checker, 361
 - Web Accessibility Initiative (WAI), 192
 - web address, 54–55
 - Web APIs (Web Application Programming Interfaces), 521–522
 - web browsers
 - cross-browser testing code, 60–62
 - browser market share, 60–61
 - testing online, 61–62
 - testing suite, 61
 - displaying Console in, 632–633
 - handling very old browsers, 407
 - with JavaScript turned off, 405–407
 - loading HTML file on, 13–14
 - setting up test browsers, 40
 - web video formats supported by, 134
 - window size, 768
 - web coding, 400
 - Web Content Accessibility Guidelines (WCAG), 361, 382
 - web development tools, 624
 - colors in, 359–362
 - displaying, 292–294
 - displaying grid line numbers in, 756–757
 - displaying named areas in, 760–761
 - network speeds, 768
 - opening in browsers, 630–631

- web files, 57
- web font families, 368–375
 - linking to, 368–370
 - self-hosting, 372–375
- Web Hosting Talk, 53
- web hosts, 47–57
 - directory and web address, 54–55
 - finding, 52–53
 - making changes to web files, 57
 - making hard disk mirror web home, 55–56
 - providers, 48–49
 - finding free hosting provider, 48–49
 - signing up with commercial hosting provider, 49
 - using existing internet provider, 48
 - terminology, 49–52
 - uploading site files, 56–57
- Web Hypertext Application Technology Working Group (WHATWG), 62
- web pages
 - adding dynamism with JavaScript
 - defined, 33
 - displaying message to user, 35–36
 - <script> tag, 34–35
 - writing text to page, 37
 - adding element to, 535–538
 - adding structure with HTML, 14–24
 - for accessibility, 187–188
 - elements and tags, 15–17
 - style vs., 23–24
 - tag attributes, 17–18
 - text, 21–23
 - web page title, 20–21
- adding style with CSS, 24–32
 - adding styles to pages, 28–32
 - cascading, 26
 - rules and declarations, 26–27
 - sheets, 25–26
 - styles, 24–25
- basics of, 10–14
 - launching new text file, 11
 - saving HTML files, 11–12
 - test driving web pages, 12–14
- building reactive pages with events, 546–552
 - event types, 547–549
 - events defined, 547
 - getting data about event, 549–551
 - preventing default event action, 551–552
- calling functions after loaded pages, 486–487
- general discussion of, 7–8
- linking, 102–103
- reasons to learn, 8–9
- relationships of elements on, 236–238
- responsive layouts, 765–800
 - adaptive layouts, 779–791
 - defined, 766–769
 - fluid layouts, 769–779
 - with images, 791–794
 - mobile-first web development, 795–800

- web servers, 48
- web storage, 617–626
- adding data to, 623–625
- getting data from, 625
- JSON, 618–623
 - converting JavaScript object to, 621–622
 - converting JSON string to JavaScript object, 622–623
 - declaring and using variables, 620–621
 - syntax for, 619–620
- overview, 618
- removing data from, 626
- web storage tab, 632
- Web Video Text Tracks (WebVTT) file, 190–191
- WebAIM (Web Accessibility In Mind) Contrast Checker, 361
- WebDev Workbench, 4, 45–46
- WebM, 133, 139
- week picker, 180
- weight, text, 384–386
- weight hierarchy, 311–313
- WHATWG (Web Hypertext Application Technology Working Group), 62
- :where() functional pseudo-class, 275, 317
- while loops, 467–470
- white space, 22
- whitespace, 211–213
- widget ARIA roles, 196–197
- widgets, 797
- width, boxes
- borders, 216
- containing blocks, 229

width attribute, 135

Windows

Notepad on, 11

web browsers, 13

web development tools for,
292–293

WOFF2 files, 373

World Wide Web Consortium
(W3C), 62

wrapping flex items, 714–716

writing, text to pages, 37

writing mode, 219

X

x-height (ex), 324, 371

Y

YouTube, 132

yy argument, 600

yyyy argument, 600

Z

z-index property, 696–698

Zoom command, 186

zoom level, 768