Index

Symbols

> (direct descendant selector), 160
$.each() utility, 289
$.getJSON() function, 77, 248, 339
$.grep() function, 55–56, 80
$.map() function, 55–56, 80, 290
_ (underscore character), 274

A

<a> element, 208
activityId attribute, 336, 337
"add" event, 335
addEventListener method, 165
Adobe Illustrator, 188
:after pseudoselector, 160
AJAX (Asynchronous JavaScript and XML), retrieving data using, 75–87
  creating chart, 85–87
  first level of data, 77–80
  processing response, 80–81, 84–85
  second level of data, 81–82
  source data format and structure, 76–77
animate() function, 213
animation variable, 210
annotating sparklines, 101–105
  adding chart, 102–103
  adding primary annotation, 103–104
  overview, 101
  preparing data, 102
  preparing HTML markup, 102
  providing additional information, 105
any() function, 292
APIhub, 87
APIs, repositories for, 87
app/ folder, 299
append() method, 226
app.js file, 340
application frameworks, 297
apply() function, 190–191, 212, 278
arc() function, 255, 259
arguments object, 216
arrays
  combining, 278–280
  extracting elements by position, 275–277
  finding elements in, 281–282
  generating, 282–283
  overview, 275
  removing invalid data values, 280–281
Asynchronous JavaScript and XML. *See* AJAX (Asynchronous JavaScript and XML)
attr() function, 226
attrTween() function, 263, 264, 265
authorization_token object, 328
autoscale option, 27

B

Backbone.js library, 297, 322, 323
  authorizing users, 328–330
  collections
    filtering, 335–336
    paging, 330–334
    supporting run models outside of, 342–344
  creating router, 340–342
  fine-tuning the app, 349–351
  history feature, 341
  responses
    accepting, 330
    parsing, 336
  retrieving details, 336–340
  views, 307–311
    allowing users to change, 345–349
    dynamically updating, 334–335
  background
    of bubble charts, 35–38
    of heat maps, 127
background-color property, 153

Data Visualization with JavaScript
© Stephen A. Thomas

bar charts, 6–15
  colors, 13–14
  defining data to display, 7–8
  `<div>` element to hold, 7
  drawing, 8–9
  dummy HTML elements in, 14–15
  horizontal axis, fixing, 10–11
  JavaScript required, 6–7
  styling, 12–13
  vertical axis, fixing, 9–10
`barWidth` property, 11
`:before` pseudoselector, 160
`bind()` function, 322
Bostock, Mike, 253
bower tool, 301
bower.json file, 299
Brewer, Cynthia, 191
browser, managing data in. *See* data management, in browser
Bruls, Mark, 120
bubble charts, 34–41
  background of, 35–38
  colors in, 38–40
  defining data to display, 34–35
  legends in, 40–41
  plotting data in, 36–37
  when to use, 34, 46
`buildLabelAnimation()` function, 216
bullets, 163

C

C++ library, 151
`calculateNewPath()` function, 263, 264, 265
`call()` function, 332
`<canvas>` interface, 141, 143
Cascading Style Sheets. *See* CSS (Cascading Style Sheets)
CDNs (content distribution networks), 49
`"change"` event, 335
charts. *See* bar charts; bubble charts; interactive charts; line charts; radar charts; scatter charts; sparklines; traditional chart types; tree maps
Charts view, 322–324
checkboxes, 52
Chroma.js library, 191
`chroma.scale()` function, 191
Chronoline.js library, building timelines with
  drawing timeline, 150
  libraries required, 148–149
  preparing data, 149–150
  setting options for data, 150–153
`<circle>` element, 235, 236, 237, 243
`<cite>` element, 156, 164, 165
`classList` interface, 165
`className` interface, 165, 309
`click` event, 208, 345, 346
`clickable` option, 206
`clicked()` method, 346
`clickedTag` variable, 144
`clickNode` event, 137
clicks on sparklines, responding to, 110–115
CloudFlare CDN, 49
CodeKit, 175
`collection` property, 337
collections
  finding elements in, 290–291
  iteration utilities, 289–290
  overview, 288–289
  rearranging, 292–293
  testing, 292
`color()` function, 260
colors
  in bar charts, 13–14
  in bubble charts, 38–40
  in SVGs, 191–192
  in tree maps, 123–124
`compact()` function, 280–281
composite charts, 105–109
`composite` option, 106
`concat()` method, 211
content distribution networks (CDNs), 49
context, maps for, 197–201
Continental font, 180
continuous data. *See* line charts
`Control` object, 207
cost of site, 49
`count` parameter, 331
`count` property, 128
`countby()` function, 293
`_createButton()` function, 208
Creative Commons licenses, 35

Data Visualization with JavaScript
© Stephen A. Thomas

CSS (Cascading Style Sheets)
  fixing timeline problems using, 159–160
  styling map fonts using, 184–185
  styling SVG elements using, 187–188
  styling table in legend, 193
  transitions, 165–166
`css()` function, 57
`cx` attribute, 238
`cy` attribute, 238

D

`d3.csv()` function, 256
D3.js library
  adapting traditional chart type
    adding data to chart, 231
    answering users' questions, 231–232
    controlling chart's dimensions, 226–228
    creating stage for visualization, 226
    drawing chart framework, 228–230
    overview, 224
    preparing data, 225
    setting up web page, 225–226
  creating force-directed network graph
    adding force direction to graph, 240–242
    adding interactivity, 242–245
    creating stage for visualization, 235
    drawing graph's edges, 237–238
    drawing graph's nodes, 235–237
    overview, 232
    positioning elements, 238–239
    preparing data, 233–234
    setting up page, 234–235
  creating scalable map, 245–249
  creating unique visualization
    adding interactivity, 262–267
    coloring area, 259–262
    creating scales, 254–256
    creating stage for visualization, 253
    drawing visualization, 258–259
    overview, 252–253
    preparing data, 253
    retrieving data, 256–258
    setting up page, 253
  overview, 223–224
  website for, 224
`d3.min()` function, 228
`d3.nest()` operator, 257, 258
`d3.svg.arc()` function, 255
dashboards, sparklines as, 115–117
`data()` function, 236
data management, in browser, 269–270
  arrays
    combining, 278–280
    extracting elements by position, 275–277
    finding elements in, 281–282
    generating, 282–283
    overview, 275
    removing invalid data values, 280–281
  collections
    finding elements in, 290–291
    iteration utilities, 289–290
    overview, 288–289
    rearranging, 292–293
    testing, 292
  enhancing objects
    cleaning up object subsets, 285–286
    keys and values, 283–285
    overview, 283
    updating attributes, 286–288
  overview, 269–270
  using functional programming
    evaluating performance, 273–274
    fixing performance problem, 274–275
    implementing Fibonacci algorithm, 273
    overview, 270–271
    starting with imperative approach, 271–272
`data-` prefix, 99
`data` property, 24, 329, 330
data-driven web applications. *See* web applications, data-driven
Data.gov website, 87
`data-id` attribute, 346, 347
`dataset` object, 129

Data Visualization with JavaScript
© Stephen A. Thomas

date ranges, 156
DAY_IN_MILLISECONDS variable, 152
<dd> function, 158
decimal point, dropping, 29
defaultPixelsPerValue option, 94
defaults() function, 286–288
defer() function, 322
Deferred object, 79, 80, 81, 83, 86
description list (<dl>), 157, 163, 166, 319
details() method, 341, 348, 350
details variable, 146
Details view, 317–318
difference() function, 279
direct descendant selector (>), 160
disableTooltips option, 103
display property, 54, 72, 111, 164
display:none property, 163
<dl> element, 157, 163, 166, 319
done() function, 80, 82
draw() method, Flotr object
  bubble charts, 36
  line charts, 16
  pie charts, 23, 24
  radar charts, 44
  scatter charts, 26–27
drawMap() function, 323
Dream, Tim, 139
<dt> function, 158
dummy data set, use with line charts, 18–19
dummy HTML elements, in bar charts, 14–15
duration property, 211
.dx function, 257
.dy function, 257

E

.each() function, 57, 67, 70, 83, 85, 309
easing function, 166
el property, 313
em units, 54, 161
enableTagOptions parameter, 99
enter() function, 237, 238
ev.data.node.id property, 137
events property, 345
every() function, 292
excanvas.min.js library, 6, 7, 49
exit() function, 238
.extend() function, 62, 63, 97–98, 329

F

fail() method, 80
fetch() method, 332, 333, 334, 336, 337, 338, 339
fetchGps() method, 344
fetchMore() function, 332, 333
fib() function, 271, 273, 274, 275
Fibonacci numbers, 271, 273, 278
fill attribute, 192
fillColor parameter, 93
filter() function, 290, 291
filter function parameter, 35
filter option, 38
find() function, 144, 290
findWhere() function, 291
first() function, 276
flatten() function, 281
float property, 69
Flot library, 49–50. *See also* interactive charts
  creating static graphs, 51
  and flotData, 84–85
  installing in Yeoman application, 301
  and legends, 53
  plot() function, 55, 58–59, 61–62, 63, 68–69, 71
  and plothover events, 71–74
  selection plug-in, 60–64
  and zooming in on charts, 60
Flotr object, 16, 23, 24, 26–27, 36, 44
Flotr2 library, 6. *See also* bar charts; bubble charts; line charts; pie charts
  including in web pages, 6–7
  setting up the <div> element, 7
  working around bugs, 14–15
Flotr.draw() function, 8, 15
fonts
  for maps, 180–186
    adding legend, 185–186
    combining multiple countries into single map, 182–183
    displaying single country, 181
    including in page, 180–181
    varying countries based on data, 183–185
  used by TimelineJS, 175–177
force layout tool, 240
force object, 240, 241

Data Visualization with JavaScript
© Stephen A. Thomas

"ForceAtlas2, A Graph Layout Algorithm for Handy Network Visualization" (Jacomy and Venturini), 135
forceAtlas2 plug-in, 135
force-directed network graphs, 135–136
  adding force direction to, 239–242
  adding interactivity, 242–245
  creating stage for visualization, 235
  drawing edges, 237–238
  drawing nodes, 235–237
  overview, 232
  positioning elements, 238–239
  preparing data, 233–234
  setting up page, 234–235
forEach() function, 158, 193, 195
format parameter, 78
from property, 62
functional programming, data management using
  evaluating performance, 273–274
  fixing performance problem, 274–275
  implementing Fibonacci algorithm, 273
  overview, 270–271
  starting with imperative approach, 271–272

G

<g> (group) element, 227, 229, 253
GeoJSON, 247
get() method, 150
getData() function, 116
.getJSON() function, 77, 78, 79, 81
Ghory, Imran, 120
Google Maps, 197
gps property, 323, 339
graphs. *See* network graphs
gravity parameter, 135
.grep() function, 74, 84, 144
grid lines, removing from line charts, 18
grid option, 23, 38, 69, 70, 71
group (<g>) element, 227, 229, 253
GROUP BY operation, 257
groupby() function, 292, 293
grunt command, 300
Gruntfile.js file, 299

H

handleClick function, 263
heat maps, 125–130
  background image, 127
  defining data to display, 127
  drawing, 129
  formatting data, 128–129
  HTML to hold, 128
  JavaScript required, 126
  overview, 125
  z-index property, adjusting, 130
height property, 166
height variable, 226–227
hierarchy variable, 258
history feature, Backbone.js, 341
horizontal axis
  in bar charts, fixing, 10–11
  in scatter charts, clarifying, 29–30
horizontalLines property, 18, 23
hoverable property, 71
HTML
  dummy elements, in bar charts, 14–15
  embedding SVG markup within, 189
  preparing for building timelines, 154
  semantic, creating timeline in, 155–157
HTML canvas feature, 6, 49
HTML5 Word Cloud project, 139
Hubble's law, 224
Huizing, Kees, 120
humanize() function, 320

I

id attribute, 58, 77, 80, 92, 188, 309
<iframe> element, 111, 169–170
indexOf() function, 195, 281–282
individual list item, 155
information dashboards, sparklines as, 115–117
initial() function, 277
initialize() method, 214, 309, 318, 323, 329, 337, 338
<input> element, 52, 53, 57
interactive charts, 47–88
  legends in, 53–54
  retrieving data using AJAX, 75–87
    creating chart, 85–87
    first level of data, 77–80

Data Visualization with JavaScript
© Stephen A. Thomas

interactive charts, retrieving data using AJAX (*continued*)
processing response, 80–81, 84–85
second level of data, 81–82
source data format and structure, 76–77
selecting chart content, 48–59
adding controls, 57–58
data structure for interaction, 54–55
determining data based on interaction state, 55–56
`<div>` element to hold chart, 50
drawing chart, 51–52
JavaScript required, 49–50
preparing data, 50–51
responding to interaction controls, 58–59
tracking data values, 65–75
`<div>` element to hold chart, 66–67
drawing chart, 68–71
implementing interaction, 71–75
preparing data, 67–68
zooming, 59–65
drawing chart, 60–61
enabling interaction, 63–64
preparing data to support interaction, 61–62
preparing page, 60
preparing to accept interaction events, 62–63
`interpolatePathForNonRoot()` function, 263–264
`interpolatePathForRoot()` function, 263–264
`intersection()` function, 279
`invert()` function, 285
`isGpsActivity()` method, 339, 344
iteration utilities, 289–290

J

Jacomy, Mathieu, 135
JavaScript
building timelines with
adding interactivity, 163–167
adding styles, 161–163
creating timeline in semantic HTML, 155–157
fixing timeline problems with CSS, 159–160
including supporting content, 157–159
overview, 153–154
preparing HTML, 154
starting JavaScript execution, 154
using jQuery features, 159
for creating basic bar chart, 6–7
for selecting chart content, 49–50
JavaScript Object Notation (JSON), 283
jQuery, 49. *See also* sparklines
building timelines using, 159
using with TimelineJS, 170
Yeoman and, 301
JSON (JavaScript Object Notation), 283
JSONP (JSON with padding), 78, 79
JST tool, 311

K

`.key` property, 257
`keys()` function, 283–284

L

`<label>` element, 52
`Label` object, 214, 217
`label` property, 24, 43, 137
labeling
line charts, 20–21
maps
building label animation, 216–218
creating labels for stops, 214–216
incorporating label animation, 218–220
pie charts, 24–25
scatter charts, 28–29
language patterns. *See* word clouds
`last()` function, 276
`lastIndexOf()` method, 282
`:last-of-type` selector, 160
`L.control` object, 209
`L.DomUtil.create()` method, 208
`leaflet-bar` class, 208
Leaflet-based maps, 201–222
adding animation control, 207–210
adding routes to, 205–206

Data Visualization with JavaScript
© Stephen A. Thomas

adding titles, 221–222
animating routes, 211–213
building label animation, 216–218
creating labels for stops, 214–216
drawing base map, 203–205
incorporating label animation, 218–220
overview, 201
preparing animation, 210–211
preparing data, 201–202
setting up web page and libraries, 202–203
`leaflet-control-animate` class, 208
`leaflet-control-title` class, 221
`leaflet-label` class, 215, 220
`left` position, 162
legends
adding to maps, 185–186, 193–194
in bubble charts, 40–41
in interactive charts, 53–54
LESS CSS preprocessor, 175
`<li>` element, 155, 160, 161, 164, 165
line charts, 15–21
defining data to display, 15–16
dummy data set use with, 18–19
graphing multiple data sets, 17
graphing one data set, 16–17
labeling, 20–21
vs. pie charts, 21–22
readability of, 17–18
when to use, 15, 46
`<line>` element, 237, 239, 243
`lineColor` parameter, 93, 98
`listenTo()` function, 309
lists, ordered, 155
`list-style-type` property, 159
local variables, 57
`location` property, 198

M

`.map()` function, 26, 84, 86, 102–103, 106, 110, 141, 150, 190, 205, 290
Map view, 322–324
map-based visualization
for context, 197–201
Leaflet-based maps, 201–222
adding animation control, 207–210
adding routes to map, 205–206
adding title, 221–222
animating routes, 211–213
building label animation, 216–218
creating labels for stops, 214–216
drawing base map, 203–205
incorporating label animation, 218–220
overview, 201
preparing animation, 210–211
preparing data, 201–202
setting up web page and libraries, 202–203
map fonts, 180–186
adding legend, 185–186
combining multiple countries into single map, 182–183
displaying single country, 181
including in page, 180–181
varying countries based on data, 183–185
overview, 179–180
scalable map, 245–249
Scalable Vector Graphics (SVGs)
adding interactivity, 194–197
adding legend, 193–194
collecting data, 190–191
colors, 191–192
creating SVG map, 188
embedding map in page, 189
overview, 186–188
MapReduce, 290
maps, 35. *See also* heat maps; map-based visualization; tree maps
`Math.max()` function, 190, 212
`max` property, 129
`max-height` property, 163, 164, 166
`maxSpotColor` option, 94, 102
`maxZoom` option, 204
`memoize()` function, 274–275
Mercator projection, 35
`metrics` property, 338
`metricSummary` property, 320
`min()` function, 292
`minSpotColor` option, 94, 102
`MM.Map` object, 199
`mode` option, 63
`.Models` property, 304
Modest Maps library, 197–200
Moment.js library, 301, 314
mouse, tracking position of, 108

Data Visualization with JavaScript
© Stephen A. Thomas

mousedown event, 208
mouseout event, 72, 73, 104
mouseovers, 31
-moz- prefix, 166
multiple data sets, graphing on line charts, 17

N

navigate() method, 348
navigation plug-in, 60
network graphs, 130–138. *See also* force-directed network graphs
- adding interactivity, 137–138
- adding nodes to, 132–133
- automating layout of, 134–136
- connecting nodes with edges, 133–134
- libraries required, 130–131
- preparing data, 131
- when to use, 130

new Date() function, 247
Node.js platform, 298
node_modules/ folder, 299
nodeName property, 144
nodes, in network graphs
- adding, 132–133
- connecting with edges, 133–134

nonbreaking space (&nbsp;), 104
normalRangeMin option, 94

O

obj2Html() method, 320, 321
object() function, 284
offset field, 216
offset parameter, 331
<ol> element, 155
omit() function, 286
.on() function, 62, 241
onAdd() method, 207–208
opacity property, 220
OpenStreetMap, 35, 200
options attribute, 214
options object, 207, 343
options parameter, 329
options variable, 63
ordered lists, 155
overflow property, 163

P

package.json file, 299
padding, 40
padding-left property, 162
pairs() function, 284
parents() function, 346
parse() function, 330, 332
<path> element, 188, 189, 192, 195, 247, 248, 259, 262
pause() function, 213
pick() function, 285–286, 329
pickColor function, 123
pie charts, 21–25
- defining data to display, 23
- drawing, 23–24
- labeling, 24–25
- vs. line charts, 21–22
- titles for, 24–25
- when to use, 22, 46

pixel (px) units, 161
plot extension, 51
plot() function, 55, 58–59, 61–62, 63, 68–69, 71
plothover events, 72
plotObj.draw() function, 58–59
plotObj.setupGrid() function, 58–59
plotselected event, 62
pluck() function, 291
pointOffset() function, 72
polyline() function, 204, 205, 206
position property, 72, 128
position: relative style, 142
prefix parameter, 78
ProgrammableWeb, 87
Properties view, 318–321
pure libraries, 297
px (pixel) units, 161

Q

qTip2 library, 148
querySelectorAll() function, 195

R

r attribute, 238
radar charts, 41–45
- creating, 44–45
- defining data to display, 42–44
- when to use, 41–42, 46

Data Visualization with JavaScript
© Stephen A. Thomas

range() function, 282–283
ranges parameter, 62
Raphaël library, 120, 148
readability, of line charts, 17–18
real-time data, 115–117
<rect> element, 231
reduce() method, 290, 320
reject() function, 291
relatedTarget property, 73
relationships. *See* network graphs
remove() method, 351
remove option, 333
render() method, 309, 310, 314, 318, 321, 322, 323, 338
renderRun() method, 310, 335, 346
reset() method, 209
resolve() method, 82
rest() function, 277
retrieving data using AJAX, 75–87
  creating chart, 85–87
  first level of data, 77–80
  processing response, 80–81, 84–85
  second level of data, 81–82
  source data format and structure, 76–77
routes
  adding to maps, 205–206
  animating, 211–213
routes property, 340–341
run.js file, 303
run.spec.js file, 303

S

same-origin policy, 78
--save option, 301
scalable map, creating
  adding interactivity, 250–252
  drawing map, 248
  map data, retrieving, 247–288
  map projection, 246
  overview, 245
  plotting data, 248–249
  preparing data, 245–246
  retrieving data, 248
  setting up page, 246
  SVG container, initializing, 247
Scalable Vector Graphics (SVGs)
  adding interactivity, 194–197
  adding legend, 193–194
  collecting data, 190–191
  colors, 191–192
  creating SVG map, 188
  embedding map in page, 189
  overview, 186–188
scale objects, 254
scalingRatio parameter, 135
scatter charts, 25–33
  adjusting chart's axes, 27–28
  answering users' questions, 30–33
  clarifying x-axis, 29–30
  defining data to display, 26
  formatting data, 26
  labeling, 28–29
  plotting data, 26–27
  subtitles for, 29
  titles for, 29
  when to use, 25–33, 46
search rank, 155
selected class, 244
selected() method, 348
selecting chart content
  adding controls, 57–58
  data structure for interaction, 54–55
  determining data based on interaction state, 55–56
  <div> element to hold chart, 50
  drawing chart, 51–52
  JavaScript required, 49–50
  preparing data, 50–51
  responding to interaction controls, 58–59
selection option, 63
selection plug-in, 60
selection.enter() function, 237
semantic HTML, creating timeline in, 155–157
series option, 68
setInterval() function, 116
setTimeout() function, 116
settings object, 329
s.graph.nodes() function, 137, 138
shading, in tree maps, 122–125
shallow parameter, 281
shift() method, 219
show property, 56, 58, 70
showLabels property, 23
shuffle() function, 293
Sigma library, 130. *See also* network graphs

Data Visualization with JavaScript
© Stephen A. Thomas

Silver Comet's timetable, 201–202
simplicity, importance of, 5–6
single-page applications (SPAs), 340
Sketch application, 188
`slice()` method, 216, 218
`slideDown()` function, 113
`slideUp()` function, 113
`slowDown` parameter, 135
`.some()` method, 134, 145
`sortby()` function, 292
`sortedIndex()` function, 282
`source` property, 138, 243
Southerner's timetable, 201–202
SPAs. *See* single-page applications (SPAs)
space, nonbreaking (`&nbsp;`), 104
`<span>` element, 54, 57
  semantic markup vs., 155
  using with map fonts, 181, 182
`sparkline()` function, 94–95, 97, 99, 100
`sparklineClick` event, 111
`sparkLineColor` attribute, 99
`sparklineRegionChange` event, 103–104, 108, 109
sparklines
  adjusting to match Tufte's definition, 93
  annotating, 101–105
    adding chart, 102–103
    adding primary annotation, 103–104
    overview, 101
    preparing data, 102
    preparing HTML markup, 102
    providing additional information, 105
  charting many variables using, 94–101
    creating unique style for specific chart, 99–101
    default style for chart, 96–97
    drawing chart, 95–96
    modifying default style for special classes, 97–99
    overview, 94
    preparing HTML markup, 94–95
  clicks on, responding to, 110–115
  composite charts using, 105–109
  drawing, 92
  drawing composite charts
    adding annotations, 107–108
    adding closing price chart, 106–107
    drawing trading volume chart, 106
    overview, 105–106
    showing details as chart, 108–109
  HTML markup, 91–92
  as information dashboards, 115–117
  JavaScript required, 91
  origin of, 90
  responding to click events
    adding chart, 110
    animating, 114–115
    handling click events, 111–112
    improving transitions, 112–114
    overview, 110
  tool tips and, 101–105
  updating charts in real time
    overview, 115
    retrieving data, 116
    updating visualization, 116–117
`sparklines` property, 103
`spotColor` property, 94
`s.refresh()` function, 138
Stack Overflow, 140
Stamen Design, 35, 197, 199, 200
Stately font, 180
storyjs_embed.js wrapper, 172
styles, using with timelines, 161–163
subtitles, for scatter charts, 29
`summary()` method, 341, 347, 349, 350
`summaryRow.ejs` template, 312
SVGs. *See* Scalable Vector Graphics (SVGs)
`<svg>` element, 226, 227, 235
`sync()` method, 328, 331, 344

T

`<table>` element, 193, 312
tables, use with sparklines, 94–101
tag clouds. *See* word clouds
`tagName` property, 308, 309, 310, 319
`tagOptionsPrefix` parameter, 99
tags, 140
`tags` array, 145

Data Visualization with JavaScript
© Stephen A. Thomas

`target` property, 138, 164, 243
`<tbody>` element, 312, 346
`template` property, 308, 309
test/ folder, 299
text, in bar charts, 15
`this.collection` parameter, 309
`tick` event, 240
`tickDecimals` property, 10, 29
`tickFormatter()` function, 20, 30, 70
`ticks` property, 11
`<time>` element, 156, 161
`timeline.css` stylesheet, 174–175
TimelineJS component
- adjusting timeline styles, 174–177
- creating default timeline, 172–174
- overview, 167
- preparing data, 170–172
- previewing standard component, 168–170
- required components, 170

timelines
- building with JavaScript
  - adding interactivity, 163–167
  - adding styles, 161–163
  - creating timeline in semantic HTML, 155–157
  - fixing timeline problems with CSS, 159–160
  - including supporting content, 157–159
  - overview, 153–154
  - preparing HTML, 154
  - starting JavaScript execution, 154
  - using jQuery features, 159
- building with library
  - drawing timeline, 150
  - libraries required, 148–149
  - preparing data, 149–150
  - setting Chronoline.js options for data, 150–153
- overview, 147–148
- using web component
  - adjusting timeline styles, 174–177
  - creating default timeline, 172–174
  - overview, 167
  - preparing data, 170–172
  - previewing standard component, 168–170
  - required components, 170

`title` attribute, 113
`titleize()` function, 320
titles
- for line charts, 20
- for pie charts, 24–25
- for radar charts, 45
- for scatter charts, 29

`to` property, 62
TodoMVC, 297
`toJSON()` method, 314, 321
tool tips, 31, 101–105
`top` position, 162
`<tr>` element, 311
tracking data values, 65–75
- `<div>` element to hold chart, 66–67
- drawing chart, 68–71
- implementing interaction, 71–75
- preparing data, 67–68

tracking mouse position, 108
traditional chart types, adapting using D3.js
- adding data to chart, 231
- answering users' questions, 231–232
- controlling chart's dimensions, 226–228
- creating stage for visualization, 226
- drawing chart framework, 228–230
- overview, 224
- setting up web page, 225–226

`transition` property, 166
transitions
- animating, 113–114
- CSS, 165–166

transparency, 106
tree maps, 120–125
- drawing, 121–122
- libraries required, 120–121
- overview, 120
- preparing data, 121
- shading in, 122–125

`TreeMap.draw()` function, 124
trends, 21
`({ trigger: true })` parameter, 348
Tufte, Edward, 90, 93
`type` property, 171

Data Visualization with JavaScript
© Stephen A. Thomas

U

underscore character (_), 274
Underscore.js library, 270
  arrays
    combining, 278–280
    extracting elements by position, 275–277
    finding elements in, 281–282
    generating, 282–283
    overview, 275
    removing invalid data values, 280–281
  collections
    finding elements in collection, 290–291
    iteration utilities, 289–290
    overview, 288–289
    rearranging, 292–293
    testing, 292
  enhancing objects
    cleaning up object subsets, 285–286
    keys and values, 283–285
    overview, 283
    updating attributes, 286–288
  memoizing JavaScript functions using, 274–275
Underscore.string library, 301, 319, 320
`union()` function, 278
`uniq()` function, 279, 282
`update()` function, 238
`url()` method, 339

V

`validate()` method, 335, 336
`values()` function, 284
`.values` property, 258
van Wijk, Jarke J., 120
variables
  charting many, 94–101
  local, 57
Venturini, Tommaso, 135
vertical axis
  extending range of, in line charts, 18
  fixing in bar charts, 9–10
`verticalLines` property, 18, 23
Visual Display of Quantitative Information, The (Tufte), 90
visualization. *See* D3.js library; map-based visualization
`VMM.Timeline` constructor, 173
`volume` property, 106

W

web applications, data-driven
  adding unique dependencies, 301–302
  authorizing users, 328–330
  collections
    defining, 306–307
    filtering, 335–336
    paging, 330–334
    supporting run models outside of, 342–344
  creating Backbone.js router, 340–342
  defining new project, 298–300
  fine-tuning the app, 349–351
  installing development tools, 298
  model
    defining application's, 303–304
    implementing, 304–306
  overview, 295–296
  responses
    accepting, 330
    parsing, 336
  retrieving details, 336–340
  selecting application library, 297
  views
    additional, defining, 317
    allowing users to change, 345–349
    Charts view, 322–324
    Details view, 317–318
    dynamically updating, 334–335
    main view, 307–314
    Map view, 322–324
    Properties view, 318–321
`-webkit-` prefix, 166
`.when()` function, 84
`where()` function, 291
`width` variable, 226–227
Wied, Patrick, 125
Wikimedia Commons, 188
`wind` property, 38
`window.onload` event, 8
`wins2` array, 13
`without` method, 281

Data Visualization with JavaScript
© Stephen A. Thomas

WolframAlpha, 111
word clouds, 138–145
    adding interactivity, 143–146
    creating, 142
    libraries required, 139–140
    markup required, 141–142
    overview, 138–139
    preparing data, 140–141
wordcloud2 library, 139. *See also* word clouds
WordFreq JavaScript library, 140
WorldGrayCanvas set, 204

X

`.x` function, 257
x-axis. *See* horizontal axis
`xaxis` object, 62, 68

Y

`.y` function, 257
y-axis. *See* vertical axis
`yaxis` object, 62, 69
Yeoman application, 298–301
    defining application's collections, 306–307
    defining application's main view, 307–311
    defining main view templates, 311–314
    defining model for app, 303–304
    implementing model, 304–306
    refining main view, 314–316

Z

`z-index`, 72, 130, 173
`zip()` function, 279
zooming charts, 59–65
    drawing chart, 60–61
    enabling interaction, 63–64
    preparing data to support interaction, 61–62
    preparing page, 60
    preparing to accept interaction events, 62–63
zooming maps, 203–204

Data Visualization with JavaScript
© Stephen A. Thomas