INDEX

Symbols & Numbers

<?php and ?> (twi-character blocks), 209
@ (at sign), for SQL variables, 231
{ } (curly brackets)
    after if statement in PHP, 211
    for declaring JavaScript object, 65
$ (dollar sign), for PHP variables, 209
$ (dollar sign), jQuery object, 176–177
° (degrees), 3
$.get function, 177
′ (minutes), 3
% (modulus operator), 148
″ (seconds), 3
; (semicolon), for PHP, 209
[ ] (square brackets), for arrays, 66, 293

A

<a> tag, 77
access tokens, 165
addControls function, 309
addData function, 329, 334
addDir function, 281
addFilter function, 309
addHandler function, 102
addImageOverlay function, 310
adding
    content with jQuery, 302–303
    markers, 106
additive filtering, in Mapstraction, 269
addLargeControls function, 16, 310
addListener function (Google), 256
addMapTypeControls function, 311
addMarker function, 24, 311
add_marker function, for Weather project, 243
addMarkerWithData function, 311
addOverlay function, 188, 193, 249, 312
add_point function, 48
addPolyline function, 312
addPolylineWithData function, 312–313
address, getting from geographic point, 54–58
addSmallControls function, 15, 313
addTileLayer function, 313
add_weather function, 242
Administrative Area Name, in Google, 51
Airy, George Biddell, 4
Ajax (Asynchronous JavaScript And XML), 305
    call to PHP file, 228
    user location from Loader, 168–169
Ajax Loader, user location from, 168–169
alert function (JavaScript), 290
alerts, 109, 114
ALTER TABLE statement (SQL), 231–232
altitude, 51
Amazon account, 92
Amazon EC2 machine image, 90
Amazon Machine Images (AMIs), 91
anonymous functions, 102, 177, 300–301, 303–304
antipodal meridian, 4
Apache web server, and PHP, 206
API (Application Programming Interface), 2
API key from Yahoo!, 9, 53, 135
    for Upcoming, 263
applyOptions function, 314
area, rectangle for declaring, 71–72
arithmetic, in JavaScript, 292–293
array variable, push function for, 73
arrays
    in JavaScript, 293–294
    of LatLonPoints, 66
    for markers, 34
    in PHP, 212–213
associative array, 212–213
    looping through, 214
Asynchronous JavaScript And XML (Ajax). *See* Ajax (Asynchronous JavaScript and XML)
at sign (@), for SQL variables, 231
Atom format, GeoRSS inside, 185–186
attributes, in XML, 174
Auth for Web-based Services, 163

authenticating users, 163–164
authorize.php file, 164
auto-zooming, 35
`autoCenterAndZoom` function, 35, 314
AWS Management Console, 92

B

Basic Geo Vocabulary, 187
bearing, 130–131
blogs, geo-tagging, 184
bounding box
  checking if point is within, 137–140
  finding for polygon, 143–145
  getting random point in, 140–142
`BoundingBox` class, 71
`BoundingBox` object, 141
  global variable for, 112
`BoundingBox_to_Polyline` function, 139
bounds, moving map outside preset, 112–115
`boundsInBounds` function, 114–115
box, GeoRSS to declare, 185

C

callback function, for Ajax call, 177
callback page, 164
callback URL, for Fire Eagle, 163
callback.php file, 164–165
`call_twitter_geo` function, 273–274
Cascadenik, 97
Cascading Style Sheets (CSS)
  for earthquake project legend, 255
  to position `div` element, 244–245
cell tower triangulation, 158
center of map, 8
  preserving previous, 110
  resetting, 20
  retrieving current, 20
  returning to, after closing message box, 109–110
`centerAndZoomOnPoints` function, 314
`check_bounds` function, 138
`check_hover` function, 257–258
`check_intersection` function, 146–147
`check_polygon` function, 147–148, 149
child elements in XML, 174
circles for search radius, 67–70
  overlaying image, 69–70
city-level coordinates, precision, 6
`ClientLocation` JavaScript object (Google), 168–169, 271
`closeBubble` function, 29
`closeInfoBubble` event, 109
closest marker, determining, 125–128
closing message box, 107–108
  returning to center after, 109–110
closing tag in XML, 174
CloudMade, 93
cluster icon, changing, 41
ClusterMarker utility, 40
clusters of markers, 39–41
color
  of polylines, 65
  for states/countries, 74–76
`color_state` function, 76
combining strings, in PHP, 209
comma-separated values (CSV), 51
  importing to MySQL, 223–224
command interpreter for MySQL, 170, 218
  adding columns to table, 231–232
  importing data using, 225
comments, 20
  slashes for, 25
comparison operators, in PHP, 212
concatenating strings
  in JavaScript, 293
  in PHP, 209
concerts. *See* Music Events project
conditional statements
  in JavaScript, 294–296
  in PHP, 211–212
constants, in Mapstraction, 19
`contains` function, 324–325
continual updates, to user location, 160–161
controls
  adding, 16–17
  custom, 76–78
converting
  between decimal representation of coordinates and degree format, 5–6
  earthquake data to JSON, 252–253
  file format for Mapnik tile generator, 97
  textual values to floating point numbers, 134
  weather results to JSON, 239–241
  XML, 168
  XML to JSON, 198–200

coordinate systems
decimal representation, 3
for geographic points, 3–7
postal code, 58–60
copyright, 14
countries, color for, 74–76
create_map function, 8
adding markers, 24
calling Google geocoder, 55
with circle, 68
click event handler, 103
for clustering markers, 40
for drawing lines, 62
for driving directions, 124
for initializing map, 47
for reverse geocode, 56
setting map type, 19
Creative Commons, 81
CSS (Cascading Style Sheets)
for earthquake project legend, 255
to position div element, 244–245
CSV (comma-separated values), 51
importing to MySQL, 223–224
cUrl, 216
curly brackets ({ })
after if statement in PHP, 211
for declaring JavaScript object, 65
custom controls, 76–78
custom icons, adding to map, 30
custom tiles, 90–99

D

data formats, 173. *See also* JSON (JavaScript Object Notation); XML (Extensible Markup Language)
GeoRSS, 184–188
GPX, 194–198
KML (Keyhole Markup Language), 50, 188–193
data, loading with jQuery, 305–306
database, 218. *See also* MySQL
creating, 220
getting nearest locations from, 150–151
hosting IP database, 169–171
of postal code coordinates, 59–60
storing locations to, 219–223
date function (PHP), 265
decimal representation of coordinates, 3
converting between degree format and, 5–6
determining precision, 6–7
declutterMarkers function, 314
degree representation of coordinates, converting between decimal format and, 5–6
degrees (°), 3
degrees_to_radians function, 284
digital cameras, timestamps for photos, 195
DirectionsService object, 121, 280
distance
calculating between two points, 117–119
finding with routing, 120–122
distance function, 118, 327
div tag, 8
CSS to position, 244–245
images as child elements, 14
<Document> tag (KML), 189
documentation, for Mapstraction, 308
document.getElementById function, 267
doFilter function, 314–315
dollar sign ($), for PHP variables, 209
dollar sign ($) object (jQuery), 176–177
downloading
circle graphics, 69
jQuery JavaScript library, 176, 301
Mapstraction files, 11
postal code databases, 59–60
dragging map, 13
user interactivity, 103–105
driving directions, creating, 122–125
driving distance, determining, 120

E

Earthquakes project, 236, 247–260
converting earthquake data to JSON, 252–253
custom earthquake map, 250–260
GeoRSS for displaying, 248–250
legend, 255–256
plotting earthquakes, 253–255
zoom to hotspot regions, 256–260
elastic compute cloud, 91
else statement, 295
empty values, 296
enableHighAccuracy option for geolocation, 162
endPan event, 113

equals function, 327
equator, 3
event handler, 102, 107
event model of Mapstraction, 102
event_args object, 107
    marker property of, 106
events, 101
Excel, 223
extend function, 325
external files, for JavaScript, 290

F

fields in database, 218
file format, converting for Mapnik tile generator, 97
file function (PHP), 216
files, loading with jQuery, 305–306
fill color, of polylines, 67
filtering
    additive, 269
    markers, 36–38
    search results, 269–270
    with Yahoo! Pipes, 201–202
filter_select function, 269
find_closest_marker function, 127
findMidpoint function, 282–283
find_region function, 257
Fire Eagle
    essentials, 163
    to get user location, 162–166
    token from, 163
Firebug developer add-on, 14
Firebug panel, 15
    Inspect, 14
fireeagle.php file, 163
Firefox browser, 14
Flickr, 81
floating point number, converting textual values to, 134
for statement (JavaScript), 34, 296–297
for statement (PHP), 213
foreach statement (PHP), 213
forms, for geocode user input, 48
found_address function, 55, 56
foundLoc function, 157
free editable map of world, 90
functions
    in JavaScript, 297–301
    in PHP, 214–215
    recursive, 276

G

geo-referencing map, 85–87
geo-tagged content, 184
    photos, 81, 195
geo-tweets. *See* Twitter project
geocode_form function, 273
geocoder.us, 54
geocoding, 43
    with Google web service, 49–53
    how it works, 44–45
    with HTTP web service, 49–54
    with JavaScript, 46–48
    JavaScript vs. HTTP, 45–46
    list of services, 54
    user input, 48
geodesic polyline, 133
geographic points
    checking for location within bounding box, 137–140
    checking for location within shape, 142–149
    connecting to outside point, 145–146
    coordinate systems for, 3–7
    determining new based on bearing and coordinates, 131–133
    distance calculation between, 117–119
    finding along line, 128–133
    finding for user click, 21
    getting address from, 54–58
    getting nearest locations from database, 229–232
    getting random in bounding box, 140–142
    plotting from MySQL database, 226–228
Geography Markup Language (GML), 186
geolocation methods, accuracy of, 158
GeoRSS, 184–188, 239
    alternate encodings, 186–187
    displaying on map, 187–188
    for earthquake display, 248–250
<georss:where> tag, 186
GET command (HTTP), 156
get function, 305
$_GET PHP variable, 210
getAttribute function, 329–330
getAttributeExtremes function, 315
get_bearing function, 130–131, 284
getBounds function, 315

getCenter function, 21, 315
getCurrentPosition function, 157, 160
get_destination function, 131–132, 284
get_icon function, 254
getJSON function, 306
getloc.php file, 165–166
getMap function, 316
getMapType function, 316
getNorthEast function, 325
get_quakes function, 253, 254
get_random_by_bounds function, 33, 36, 125, 141
getSouthWest function, 325
get_twitter_geo function, 275–276
get_url function, 52
getZoom function, 18
getZoom function, 316
getZoomLevelForBoundingBox function, 316
global objects, markers as, 26
global positioning satellite (GPS), 158
global variable, 125, 292, 298–299
  for BoundingBox, 112
GML (Geography Markup Language), 186
<gml:Point> tag, 186
goDir function, 280–281
Google
  ClientLocation JavaScript object, 168–169
  geocoding web service, 49–53
  reverse geocoding with web service, 57–58
Google Charts API, 31–32
Google Earth, 188
Google map
  creating first, 7–9
  default window icon, 25
  Mapstraction to create, 11
Google Maps API
  driving directions, 120, 122–125
  HTML with, 123
  for Music Events project, 262
googlev3.core.js file, 11
GPS (global positioning satellite), 158
GPX, 194–198
  displaying tracks on map, 195–198
graphics
  plotting thumbnail images on map, 81–83
  for weather conditions, 243
  for zoom buttons, 79
Greenwich, London, 4

H

handler, 102
Haversine function, 118, 229
hide function, 26
hideAllMarkers function, 38
hiding
  all markers, 38–39
  content with jQuery, 302–303
  forecast details pane, 245
  markers, 26–27
highlight_region function, 258
hit test, 142, 146
  performing, 147–148
hosting IP database, 169–171
HostIP web service, 167–168
HTML (Hypertext Markup Language)
  form for user input on location, 154–155
  with Google Maps API, 123
  JavaScript location in pages, 289–290
HTTP (Hypertext Transfer Protocol)
  geocoding with web service, 49–54
  vs. JavaScript geocoding, 45–46
  retrieving local results with, 134–137
HYBRID map type, 19

I

icons
  adding to map, 32–33
  for clustered markers, 41
<IconStyle> tag (KML), 191
if statement
  in JavaScript, 294–296
  in PHP, 211–212
image editor, for marker icons, 29–30
images
  as child elements of div tag, 14
  overlaying on map, 83–89
importing IP address data, 170
include file, for custom PHP functions, 217
increment operator (PHP), 213
index, for PHP array, 212
INET_ATON function (MySQL), 171
infinite loop, risk of, 276
InfoBubble, 27
input.php file, 156

installing
MySQL, 217–219
PHP, 206–208
postal code database, 59–60
interactivity of map, 103
Internet Explorer, for parsing XML, 176
intersection of lines, checking for, 146–147
IP address
finding location, 171
hosting your own database to store, 169–171
importing data, 170
and user location, 153, 166–169
IPInfoDB, 170
`isEmpty` function, 326
iterations, 297

J

JavaScript. *See also* jQuery JavaScript library
anonymous functions, 102, 177, 300–301
arithmetic, 292–293
arrays, 293–294
conditionals, 294–296
curly brackets ({ }) for declaring objects, 65
external files for, 290
functions, 297–301
for getting location, 157–162
for getting user input, 154–155
vs. HTTP geocoding, 45–46
location in HTML pages, 289–290
loops, 296–297
objects, 294
other data from, 159
parsing JSON with, 181–182
parsing XML with, 174–176
reverse geocoding with, 55–56
variable scope, 298–299
variables, 291–294
JavaScript Object Notation. *See* JSON (JavaScript Object Notation)
jQuery JavaScript library, 196, 228, 301–306
to fetch JSON, 242
inserting and hiding content, 302–303
loading files and data, 305–306
for Music Events project, 262
parsing JSON with, 181–182
parsing XML with, 176–177
query document objects, 301–302
response to browser events, 303–305
for Weather project, 238
JSON (JavaScript Object Notation), 51, 180–184, 264
converting earthquake data to, 252–253
converting weather results to, 239–241
converting XML to, 198–200
for Music Events project, 266
output from MySQL as, 226–227
parsing, 181–182
parsing with PHP, 182–184
plotting places from, 228
`json_decode` function (PHP), 183
`json_encode` function (PHP), 183

K

`key=value` pairs in XML, 174
kilometer, for distance measurement, 118
KML (Keyhole Markup Language), 50, 188–193
displaying on map, 193
lines in, 189–190
polygons in, 190–191
styles in, 191–192
`KMToMiles` function, 118

L

LAMP (Linux, Apache, MySQL, and PHP), 207, 218–219
large controls, adding, 16
`latConv` function, 328
latitude coordinate, 3
layers, 61–99
lines, in KML, 189–190
lines on map
checking for intersections, 146–147
drawing, 62–65
drawing along clicks, 72–73
drawing multiple segments, 63–64
finding points along, 128–133
GeoRSS to declare, 185

<LineStyle> tag (KML), 191
loadDir function, 124, 125
loadYelp function, 285–286
local businesses, Yelp for searching, 285–287
local results
  parsing with PHP, 136–137
  retrieving with HTTP, 134–137
local variable, in PHP, 214
localhost, 207
location. *See* user location
location IDs
  for cities, 241–242
  for Yahoo! Weather API, 239
lonConv function, 328
longitude coordinate, 3, 4
looping
  in JavaScript, 296–297
  in PHP, 213–214
  risk of infinite, 276
  through markers, 34

M

MAMP, 170, 207, 218–219
Map Rectifier (MetaCarta), 85
map type, 8
  setting, 19–20
map-type controls, 17
Map Warper, 85
MapCruncher (Microsoft), 85
Mapnik, 90
mapping APIs, 2
MapQuest, zoom levels, 18
maps
  adding markers, 24–25
  changing size, 15
  creating first, 7–12
  drawing shapes on, 65–67
  recentering, 20
  tiles for, 13–14
Mapstraction, 2–3
  adding GeoRSS, 187–188
  additive filtering in, 269
  code to describe Utah, 142
  constants in, 19
  creating first map, 10–12
  documentation, 308
  event model of, 102
  geocoder, 46, 55
  for Music Events project, 262
  programmatically changing map size, 15
  Yahoo! maps with, 12
marker_clicked function, 246
markers
  adding or removing, 106
  adding to map, 24–25
  clusters of, 39–41
  creating custom icon, 29–30
  determining closest, 125–128
  determining zoom level based on, 34–36
  filtering, 36–38
  looping through, 34
  message box for clicked, 27–28
  numbered, 31–33
  removing or hiding, 26–27
  removing or hiding all, 38–39
  showing and hiding message box without clicking, 29
  user click on, 108–109
mashups, 235–237
  Earthquakes project, 236, 247–260
  Midpoint Search project, 236, 277–287
  Music Events project, 236, 260–270
  Twitter project, 236, 270–277
  Weather project, 236, 237–246
mathematical order of operations, 292
Math.random function, 141
maximumAge option for geolocation, 162
MaxMind, 167
Meet in Middle project. *See* Midpoint Search project
merging data, with Yahoo! Pipes, 202–203
message boxes, 82
  for clicked markers, 27–28
  returning to center after closing, 109–110
  showing and hiding without clicking marker, 29
  user opening or closing, 107–108
MetaCarta, Map Rectifier, 85
Microsoft, MapCruncher, 85
Midpoint Search project, 236, 277–287
  driving directions retrieval, 280–281
  finding midpoint of route, 282–284
  map and form preparation, 278–280
  searching Yelp, 285–287

milesToKM function, 118
minutes (′), 3
modulus operator (%), 148
moving map outside preset bounds, 112–116
Music Events project, 236, 260–270
  filtering results by ticket price, 269–270
  HTML for search interface, 261–263
  plotting event search results, 267–269
  retrieving event data server-side, 264–267
  Yahoo! Upcoming API search, 263
mxn.BoundingBox class, 307, 324–326
mxn.BoundingBox function, 324
mxn.core.js file, 11
mxn.google.geocoder.js file, downloading, 46
mxn.js file, 11
mxn.LatLonPoint class, 307, 326-328
mxn.LatLonPoint function, 326
mxn.Mapstraction class, variables, 307, 308-323
mxn.Mapstraction function, 308
mxn.Marker class, 307, 328–333
mxn.Marker function, 328–329
mxn.Polyline class, 307, 334–336
mxn.Polyline function, 334
mxn.util class, 307
mxn.util namespace, 336–340
mxn.util.getAvailableProviders function, 337
mxn.util.getStyle function, 338
mxn.util.KMToMiles function, 338
mxn.util.LoadScript function, 338–339
mxn.util.loadStyle function, 339
mxn.util.lonToMetres function, 339
mxn.util.$m function, 337
mxn.util.MetresToLon function, 339–340
mxn.util.milesToKM function, 340
mxn.util.stringFormat function, 340
myboxclosed function, 109
mymarkerclicked function, 109
MySQL, 170
  checking web host for, 218
  database creation, 220
  getting nearest locations from, 229–232
  importing data from spreadsheet, 223–225
  installing, 217–219
  nearest locations to postal code, 232–234
  plotting locations from database, 226–228
  precalculating values in new columns, 231–232
  storing locations to, 219–223
  using from PHP, 225–226
mysql command, 170

N

named functions, 102
namespace, in XML, 186
nearest locations, getting from database, 150–151
network latency, 170
  when dragging map, 13
noLoc function, 157
<noscript> tag, 290
numbered markers, 31–33

O

OAuth, 163
OAuth.php file, 163
objects, in JavaScript, 294
onload attribute, vs. jQuery ready event, 303–304
opacity, 75, 96
  of polylines, 67
openBubble function, 29
openInfoBubble event, 110
opening message box, 107–108
OpenOffice Calc, 223
OpenStreetMap, 90, 195
  tags, 98
overlaying
  adding tile overlays to map, 95–96
  circle image, 69–70
  image on map, 83–89

P

panning, triggering event with, 104
parameters, for Google geocode, 49
parseFloat function, 134, 198
parse_gpx function, 196–197

parsing, 174
JSON (JavaScript Object Notation), 181–184
local results with PHP, 136–137
XML with JavaScript, 174–176
XML with jQuery JavaScript library, 176–177
XML with PHP, 177–179
performance
IP database and, 170
`onload` attribute vs. jQuery `ready` event, 303–304
of query, 230–231
photos, latitude and longitude coordinates for, 195
PHP, 134
arrays, 212–213
basics, 208–210
comparison operators in, 212
conditional statements, 211–212
for converting XML to JSON, 198–199
for Fire Eagle, 163
functions, 214–215
for getting user input, 155–157
for Google geocoder web service, 52
installing, 206–208
IP availability, 168
looping, 213–214
MySQL used from, 225–226
parsing JSON with, 182–184
parsing local results with, 136–137
parsing XML with, 177–179
server-side script, 264–267
user input, 210–211
web page retrieval, 215–217
website, 208
.php file extension, 208
phpMyAdmin, 151, 218
adding data to table, 222–223
for database creation, 220
for database table creation, 220–221
Pipes. *See* Yahoo! Pipes
Placemarks, 50, 188–189
`plotResults` function, 134
`plot_thumbnail` function, 82
plotting
earthquakes on map, 253–255
from JSON, 228
locations from database, 226–228
Music Events search results, 267–269
route, 129
thumbnail images on map, 81–83
weather conditions on map, 241–244
`plot_upcoming` function, 268
.png file
for marker icons, 29–30
transparent, 69
`<Point>` tag (KML), 189
points. *See* geographic points
`points_to_bounds` function, 143–144
`polygon_circle` function, 276
polygons, 65
approximating circle with, 67–69
finding bounding box for, 143–145
GeoRSS to declare, 185
in KML, 190–191
`polylineCenterAndZoom` function, 317
polylines, 62
adding or removing, 106–107
color and thickness, 65
drawing multiple segments, 63–64
fill color and opacity, 67
geodesic, 133
GPX to store, 194
`poly.setColor` function, 65
`poly.setWidth` function, 65
`<PolyStyle>` tag (KML), 191
`POST` command (HTTP), 156
`$_POST` PHP variable, 211
postal code coordinates, 58–60
database install, 59–60
getting nearest locations to, 232–234
precision, of decimal coordinates, 6–7
preloading tiles, 13
`prepare` function, 155
Prime Meridian, 4
`print` function (PHP), 209
privacy, 153
`push` function, 73
Pythagorean Theorem, 118

Q

queries. *See also* MySQL
performance, 230–231
quotes, for setting variables holding text, 292

R

radians, 118
radians_to_degrees function, 284
Radius object, 68
records in database, 218
rectangle, for declaring area, 71–72
recursive function, 276
regular expressions, 266
remainder, 148
removeAllFilters function, 37, 317
removeAllMarkers function, 38, 317
removeAllPolylines function, 317
removeFilter function, 317–318
removeMarker function, 26, 318
removePolyline function, 318
removing
    all markers, 38–39
    markers, 26–27, 106
resizeTo function, 14, 318–319
response.Placemark array, 56
return statement (JavaScript), 298
return_center function, 246
reverse geocoding, 54–58
    in click, 56–57
    with Google web service, 57–58
    with JavaScript, 55–56
Richter value, 254
ROAD map type, 19
road maps, 90
root element (XML), 174
routes
    finding distance with, 120–122
    plotting, 129
    polyline segments for displaying, 63–64
Royal Observatory (Greenwich, London), 4
RSS feed, GeoRSS inside, 184
rubbersheeting, 85

S

SATELLITE map type, 19
satellite view, 90
    control for, 17
scope of variables, 26, 298–299
<script> tag (HTML), 289–290
search, for map location, 133
search radius, circles for, 67–70
search_upcoming function, 267
seconds (″), 3
security
    and access to outside APIs, 264
    and data retrieval, 305
    parseJSON function and, 181
semicolon (;), for PHP, 209
server-side processing, 205
SET command (SQL), 231
setAttribute function, 37, 330
setBounds function, 319
setCenter function, 20, 319
setCenterAndZoom function, 18, 319–320
setClosed function, 335
setColor function, 335
setDDist function, 122
setDebug function, 320
setDir function, 125
setDraggable function, 330
setFillColor function, 335
setHover function, 330
setHoverIcon function, 331
setIcon function, 331
setIconAnchor function, 331–332
setIconSize function, 332
setInfoBubble function, 27, 29, 332
setInfoDiv function, 245, 332–333
setLabel function, 333
setMapType function, 19, 320
setOption function, 320–321
setOptions function, 321
set_region function, 259
setShadowIcon function, 333
setWidth function, 336
setZoom function, 18, 321
shadows
    for markers, 30
    for thumbnails, 82
shapes
    checking for point within, 142–149
    drawing on map, 65–67
    GeoRSS to declare, 185
sharing location, 162–163
show function, 26
SimpleXML class (PHP), 178
SimpleXML function, 168
SimpleXML object, 266
SimpleXMLElement object, 178
simplexml_load_string function (PHP), 178
simplify function, 336
size of maps, changing, 15
size of tiles, 13
slot two in string, 254

small controls, adding, 16
sorting data with Yahoo! Pipes, 203
SQL (Structured Query Language), 150. *See also* MySQL
  for finding ZIP Code coordinates, 60
SQL injection attack, avoiding, 156
square brackets ([ ]), for arrays, 66, 293
`src` attribute
  of image, URL for tile, 14
  of `<script>` tag, 290
states (U.S.)
  color for, 74–76
  declaring points for boundary, 75
strings, 209
  concatenating, 293
  slot two in, 254
`strtotime` function (PHP), 265
styles in KML, 191–192
`swap` function, 322
`switch…case` statement, for actions based on map type, 20

T

tables in database
  adding data, 222–223
  creating, 220–221
  importing data from spreadsheet, 223–225
tags in XML, 174
textual values, converting to floating point numbers, 134
thickness, of polylines, 65
`this` variable, 177
thumbnails, plotting images on map, 81–83
Tile Drawer, 90, 92, 94–95
  "scratch" style sheet, 96
  starting EC2 instance, 91–92
tile styles, creating, 97–99
tile URLs, 14
tiles for maps, 13–14
  adding overlays to map, 95–96
  custom, 90–99
timeline, of geo-referencing session, 88
`timeout` option for geolocation, 162
timestamps for photos, from digital cameras, 195
`toggleFilter` function, 322
`toggleTileLayer` function, 323
token, from Fire Eagle, 163
`toString` function, 328
transparent *.png* file, 69
triggering event, with panning, 104
`trim` function (PHP), 216
tweets, geo-tagging. *See* Twitter project
twi-character blocks (`<?php` and `?>`), 209
Twitter project, 236, 270–277
  geocode user input, 273–274
  preparing map with user location, 271–273
  retrieving geo-tweets, 274–277
  search API, 274

U

United States Geological Survey, 184
  earthquake data, 247
Upcoming API key, 263
`urlencode` function, 136–137
user click
  drawing lines along, 72–73
  finding point for, 21
  map interactivity, 103
  on marker, 108–109
  reverse geocoding for, 56–57
user data, declaring for instance, 92–94
user input
  geocoding, 48, 273–274
  JavaScript for getting, 154–155
  PHP for getting, 155–157
user location, 153
  from Ajax Loader, 168–169
  asking user for, 154–157
  Fire Eagle to get, 162–166
  getting by IP address, 166–169
  JavaScript for getting, 157–162
  map preparation for Twitter project, 271–273
  receiving continual updates, 160–161
  sharing, 162–163
  using, 159–160
  Wi-Fi for, 158
users
  authenticating, 163–164
  moving map outside preset bounds, 112–115
  opening or closing message box, 107–108
Utah, Mapstraction code to describe, 142

V

`var` keyword, 291–292, 299
variables
  in JavaScript, 291–294
  scope, 298–299
  XML content in, 178
Veness, Chris, 130
`view_world` function, 259
`visibleCenterAndZoom` function, 35, 323

W

WAMP, 207, 218–219
warped map, applying, 87–89
`watchPosition` function, 160–161
Waters, Tim, 85
Weather project, 236, 237–246
  add forecast details pane, 244–246
  basic US map preparation, 237–238
  converting weather results to JSON, 239–241
  plotting conditions on map, 241–244
web host, checking for PHP availability, 206
web pages, retrieving in PHP, 215–217
web server, and PHP, 206
`while` loop
  in JavaScript, 296
  in PHP, 226
Wi-Fi, for location, 158
world
  free editable map of, 90
  GPX to map, 195
World Wide Web Consortium (W3C), 157, 187

X

XML (Extensible Markup Language), 174–179
  alternate data formats, 51–53
  converting, 168
  converting to JSON, 198–200
  for Google HTTP geocoder, 50
  namespace in, 186
  parsing with JavaScript, 174–176
  parsing with jQuery JavaScript library, 176–177
  parsing with PHP, 177–179
  with XPath, 179
  from Yahoo!, 53
`xpath` function, 168
XPath, XML with, 179

Y

Yahoo! account, 9
Yahoo! API key, 9, 53, 135
Yahoo!, for Fire Eagle, 163
Yahoo! Local Search API, 134
  parameters accepted, 137
Yahoo! maps
  creating first, 9–10
  geocoding web service, 53–54
  with Mapstraction, 12
  zoom levels, 18
Yahoo! Pipes
  for converting XML to JSON, 199–200, 240
  filtering, merging and sorting data with, 200–203
Yahoo! Query Language (YQL), 252
Yahoo! Upcoming API, 260
Yahoo! Weather API, 237, 239
yahoo.core.js file, 12
Yelp, 285–287

Z

ZIP Codes, 58–60
  database install, 59–60
  getting nearest locations to, 232–234
zoom control
  adding, 16–17
  graphics for, 79
zoom interface, creating, 79–81
zoom level, 8
  changes, 105–106
  changing, and tile downloads, 13
  determining correct based on markers, 34–36
  and number of tiles required to display earth, 90
  retrieving current, 18
  setting, 18–19
  setting style, 98