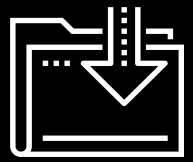
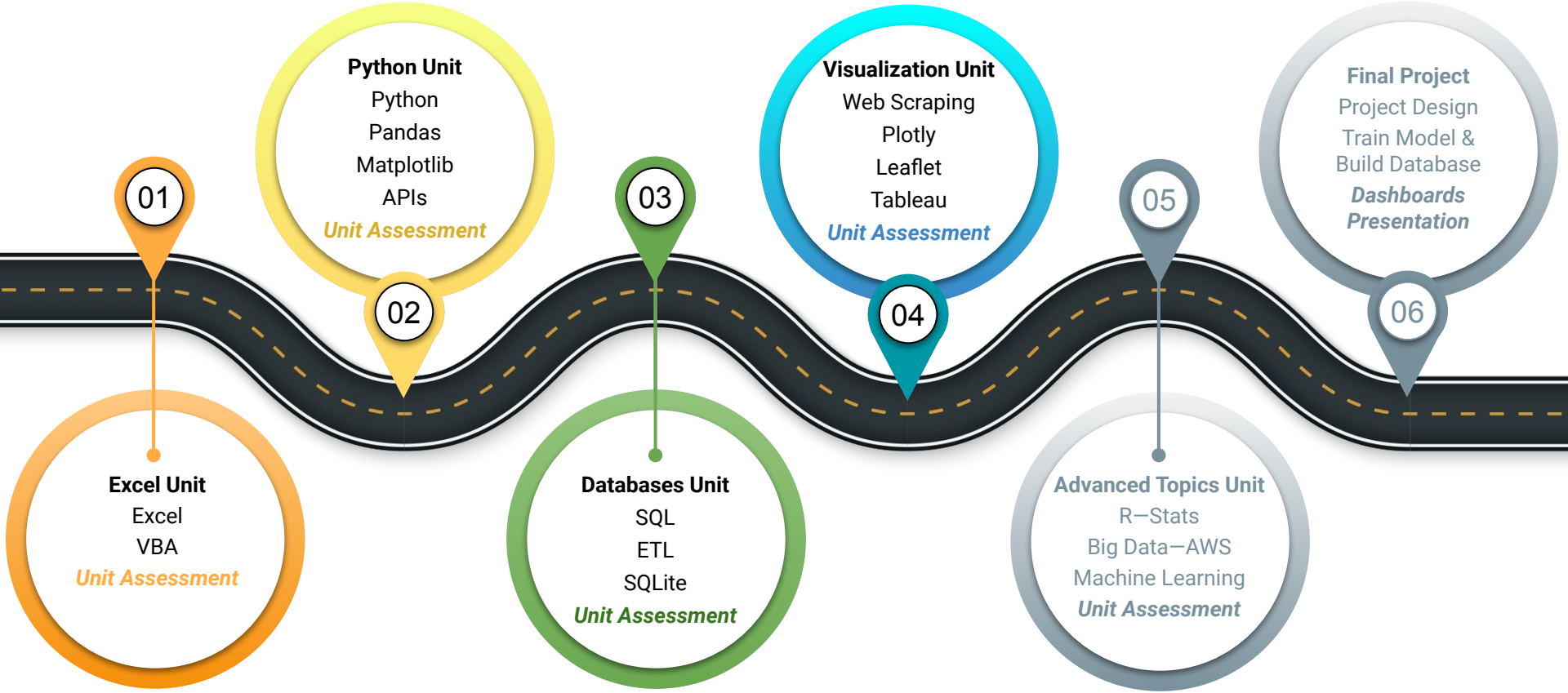
Leaflet.js

Data Boot Camp

Lesson 13.1



The Big Picture



Pro Tip:

Don’t forget how useful documentation is, especially when working with a new library! Go explore!

This Week: Leaflet.js

By the end of this week, you'll know how to:

- Create and merge a new branch from the main branch on GitHub
- Retrieve data from a GeoJSON file
- Make API requests to a server to host geographical maps
- Populate maps with GeoJSON data using JavaScript and the D3 library
- Add multiple layers to maps using Leaflet control plugins to add user interface controls
- Use JavaScript ES6 functions to add GeoJSON data, features, and interactivity to maps
- Render maps on a local server

Module 13

This Week: Leaflet.js

This Week's Challenge

Using the skills learned throughout the week, add tectonic plate and earthquake data to the map you've created, and create a new map of your choice.

Career Connection

How will you use this module's content in your career?

Module 13

How to Succeed This Week

Pro Tip:

Get creative! Leaflet.js will allow you to make appealing visualizations. Have fun with it!

Module 13

Today’s Agenda

Today's Agenda

By completing today's activities, you'll learn the following skills:

01 Leaflet.js

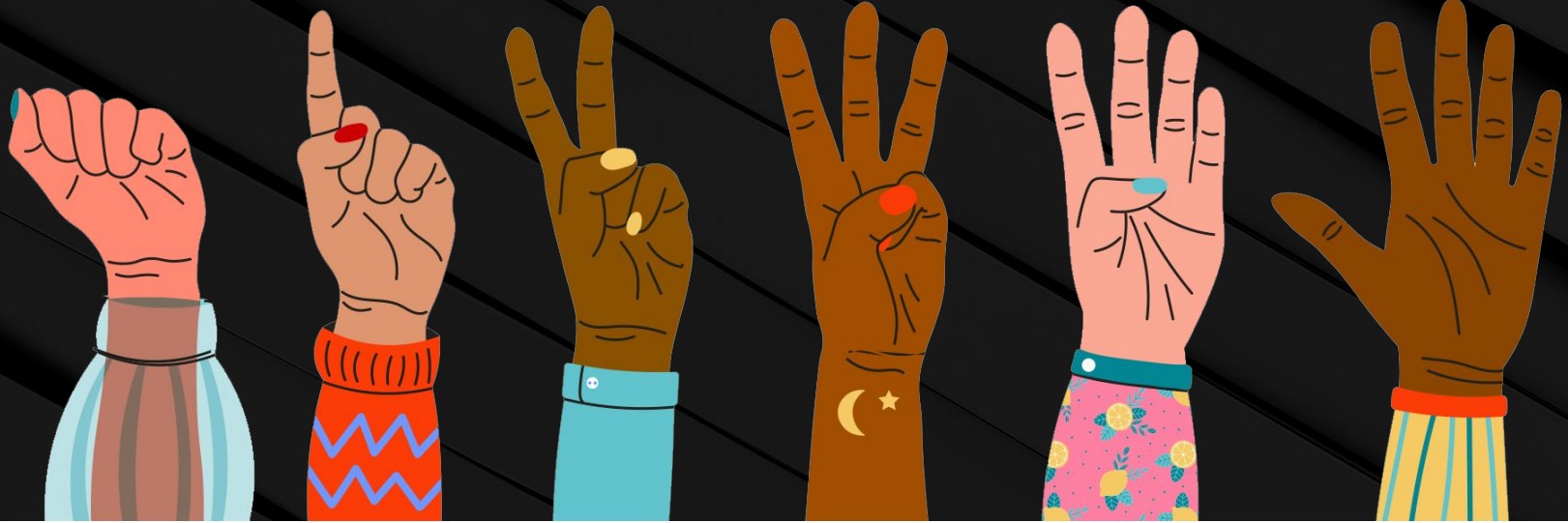
02 Mapbox API

03 Map interactivity

Make sure you've downloaded any relevant class files!

FIST TO FIVE:

How comfortable do you feel with this topic?



Instructor Demonstration

Basic Leaflet Map

Questions?

Instructor Demonstration

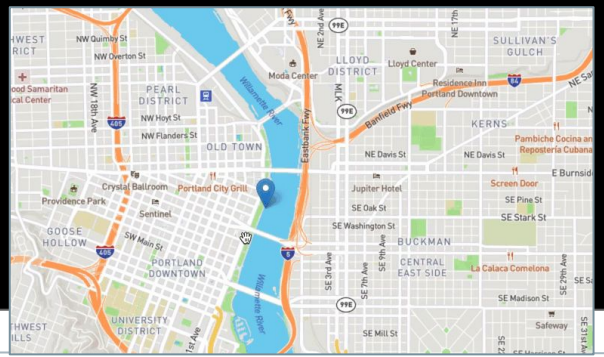
Add Markers to the Map

Add Makers to the Map

Code added to `logic.js` to add markers to the map.

```
var myMap = L.map("map", {
 center: [45.52, -122.67],
 zoom: 13`
});

L.tileLayer("https://api.mapbox.com/styles/v1/{id}/tiles/{z}/{x}/{y}?access_token={accessToken}", {
 attribution: "© <a href='https://www.mapbox.com/about/maps/'>Mapbox</a> © <a
href='http://www.openstreetmap.org/copyright'>OpenStreetMap</a> <strong><a
href='https://www.mapbox.com/map-feedback/' target='_blank'>Improve this map</a></strong>",
 tileSize: 512,
 maxZoom: 18,
 zoomOffset: -1,
 id: "mapbox/streets-v11",
 accessToken: API_KEY
}).addTo(myMap);
```



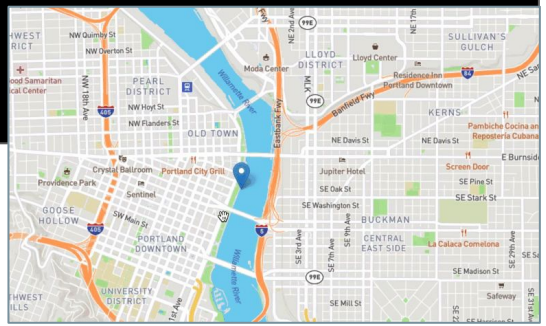
Add Makers to the Map

```
// Create a new marker
// Pass in some initial options, and then add it to the map
using the addTo method
var marker = L.marker([45.52, -122.67], {
 draggable: true,
 title: "My First Marker"
}).addTo(myMap);

// Binding a pop-up to our marker
marker.bindPopup("Hello There!");
```

Method used to add each map layer.

Method used to add text to the marker when clicked.



Activity: City Marker Map

In this activity, you will refactor the code for our US cities map to use layer groups and a layer control to be able to represent the population for the entire state as well as the city.

Suggested Time:

20 minutes

Activity: City Marker Map

Instructions	Hints
Open the logic.js file inside of the Unsolved folder. Add logic to the file to accomplish the following: • Create one layer group for city markers and a separate layer group for state markers. All of the markers have been created for you already and are stored in the `cityMarkers` and `stateMarkers` arrays. Store these layer groups in variables named “cities” and “states”. • Create a `baseMaps` object to contain the `streetmap` and `darkmap` tiles, which have been already defined. • Create an `overlayMaps` object to contain "State Population" and "City Population" layers. • Add a layers key to the options object inside of the `L.map` method, and set its value to an array containing our `streetmap`, states, and cities layers. These will determine which layers are displayed when the map first loads. • Finally, create a layer control and pass in the `baseMaps` and `overlayMaps` objects. Add the layer control to the map.	If you get stuck, refer to the Leaflet Layers Control Docs. If successful, you should be able to toggle between Street Map and Dark Map base layers, as well as turn State Population and City Population overlay layers on and off. 

Time’s Up! Let’s Review.

Instructor Demonstration

City Population Visualized

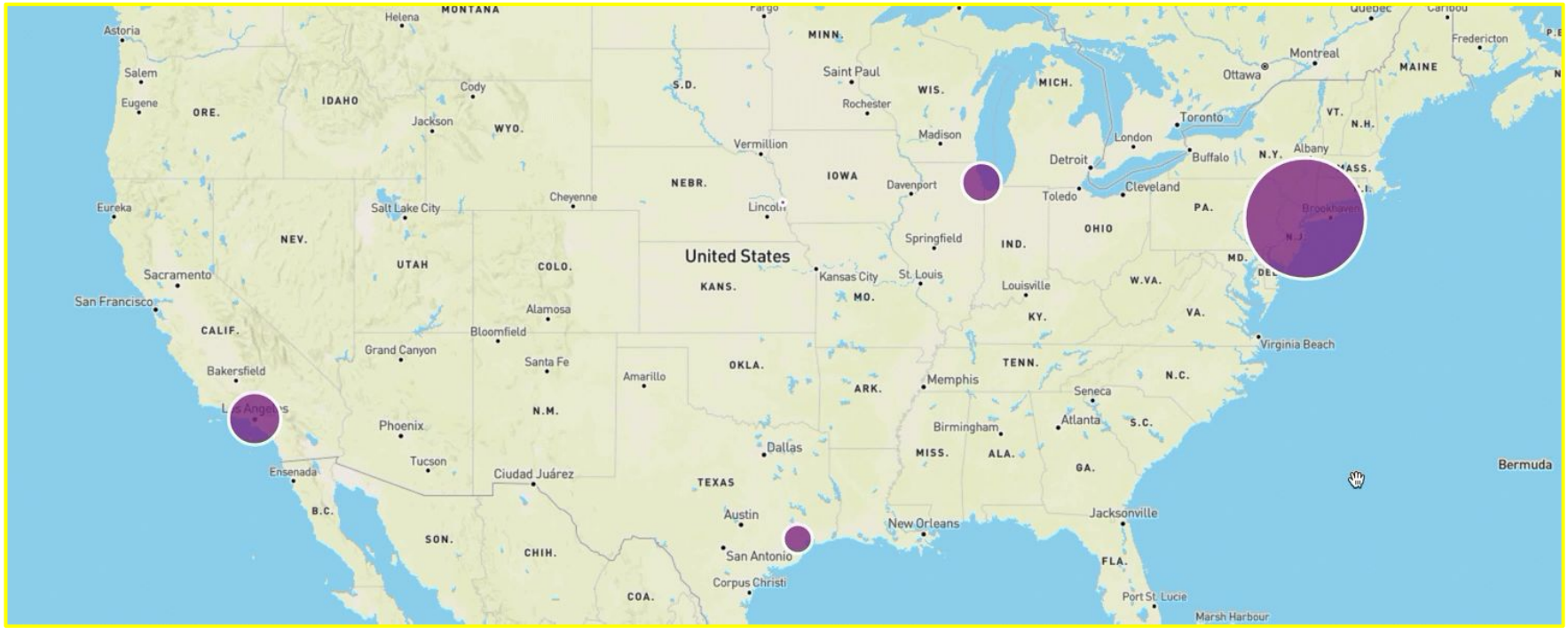
City Population Visualized

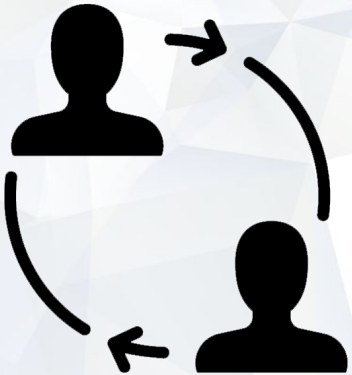
We can control the size of a circle vector layer by adjusting the population size of each city.

Run the `markerSize` function we defined above to calculate each city's circle radius based on it's population.

```
// Loop through the cities array and create one marker for each city object
for (var i = 0; i < cities.length; i++) {
 L.circle(cities[i].location, {
   fillOpacity: 0.75,
   color: "white",
   fillColor: "purple",
   // Setting our circle's radius equal to the output of our markerSize function
   // This will make our maker's size proportionate to its population
   radius: markerSize(cities[i].population)
 }).bindPopup("<h1>" + cities[i].name + "</h1> <hr> <h3>Population: " +
cities[i].population + "</h3>").addTo(myMap);
}
```

City Population Visualized





Partners Activity: World Cup Visualized

In this activity, you will work in pairs to create graduated circle maps to represent the total all-time 3-point wins for the top-ten winningest countries in the FIFA World Cup up through 2018 tournament.

Suggested Time:

15 minutes

Time’s Up! Let’s Review.

Instructor Demonstration

Layer Groups & Layer Controls

Layer Groups & Layer Controls

There are two types of layers:

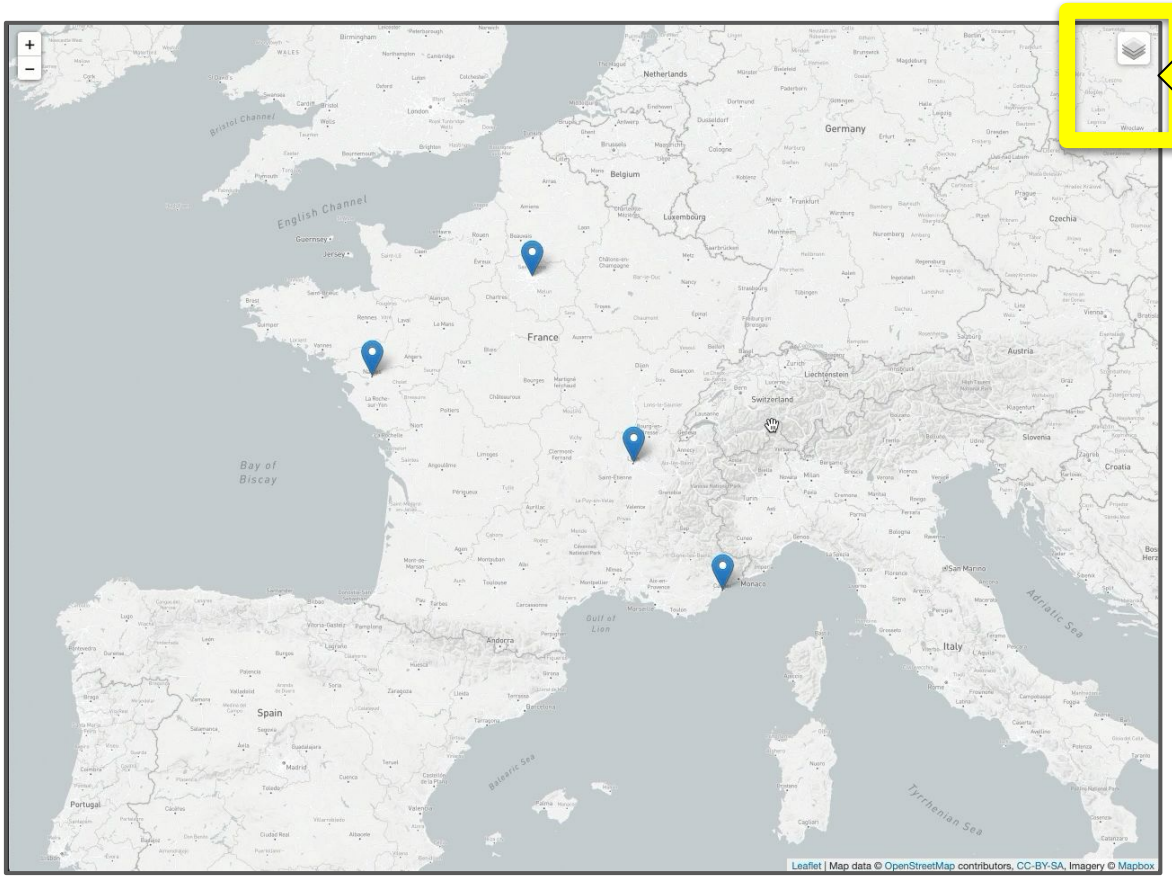
Base Layers

These are mutually exclusive to one another, so only one can be visible at a time.

Overlays

These layer over the base layers and can be turned on or off.

Layer Groups & Layer Controls



Layer Control

Layer Groups

Use LayerGroup class when you have a bunch of layers you want to handle as one in your code.

```
// An array which will be used to store created cityMarkers
var cityMarkers = [];

for (var i = 0; i < cities.length; i++) {
  // loop through the cities array, create a new marker, push it to the cityMarkers array
   cityMarkers.push(
     L.marker(cities[i].location).bindPopup("<h1>" + cities[i].name + "</h1>")
   );
}

// Add all the cityMarkers to a new layer group.
// Now we can handle them as one group instead of referencing each individually
var cityLayer = L.layerGroup(cityMarkers);
```

Time to Code

Adding More Layers

Suggested Time:

20 minutes

Questions?