

### 13.4.3

## Map Lines

**Basil** and Sadhana are ecstatic that you can add multiple locations to a map. This will be highly beneficial when you need to add the earthquake data to a map. Now, Sadhana will walk you through how to add lines to a map.

On our Leaflet map, we can plot coordinates to create lines between locations, like transportation routes.

Before we plot lines on a map, let's create a new branch called "Mapping\_Lines" that has the following folder structure:

- Mapping\_Lines
  - `index.html`
  - static
    - CSS
      - `style.css`
    - js
      - `config.js`
      - `logic.js`

Copy the necessary folders and files from one of your Mapping\_Mulitple\_Points branches and add them to the Mapping\_Lines folder.

## Map a Single Line

Adding lines to a map requires that the coordinates for the starting and ending points be a one-dimensional array with two elements: latitude and longitude. To illustrate how lines are mapped, let's map the airline route from Los Angeles to San Francisco. Mapping airline routes will help us understand how tectonic plate data is added to a map.

The starting point for our line will be the Los Angeles International Airport (LAX), with the coordinates `[33.9416, -118.4085]`. The ending point for our line will be the San Francisco International Airport (SFO), with the coordinates `[37.6213, -122.3790]`.

When we create a line in Leaflet, the starting and ending points and all coordinates along the route need to be in an array. We can assign the array to the `line` variable like this:

```
// Coordinates for each point to be used in the line.
let line = [
  [33.9416, -118.4085],
  [37.6213, -122.3790]
];
```

Let's edit our `logic.js` file to create a line from LAX to SFO.

- First, change the coordinates for the center of the map to somewhere between LAX and SFO by adding `[36.1733, -120.1794]` in the `setView()` method.
- Change the zoom level in the `setView()` method to 7.
- Add the code for our `line` variable (as seen in the previous code snippet) below the `map` variable for the center of the map.
- Lastly, create a line on a map using the Leaflet `polyline()` function. Add the following line of code after the `line` variable:

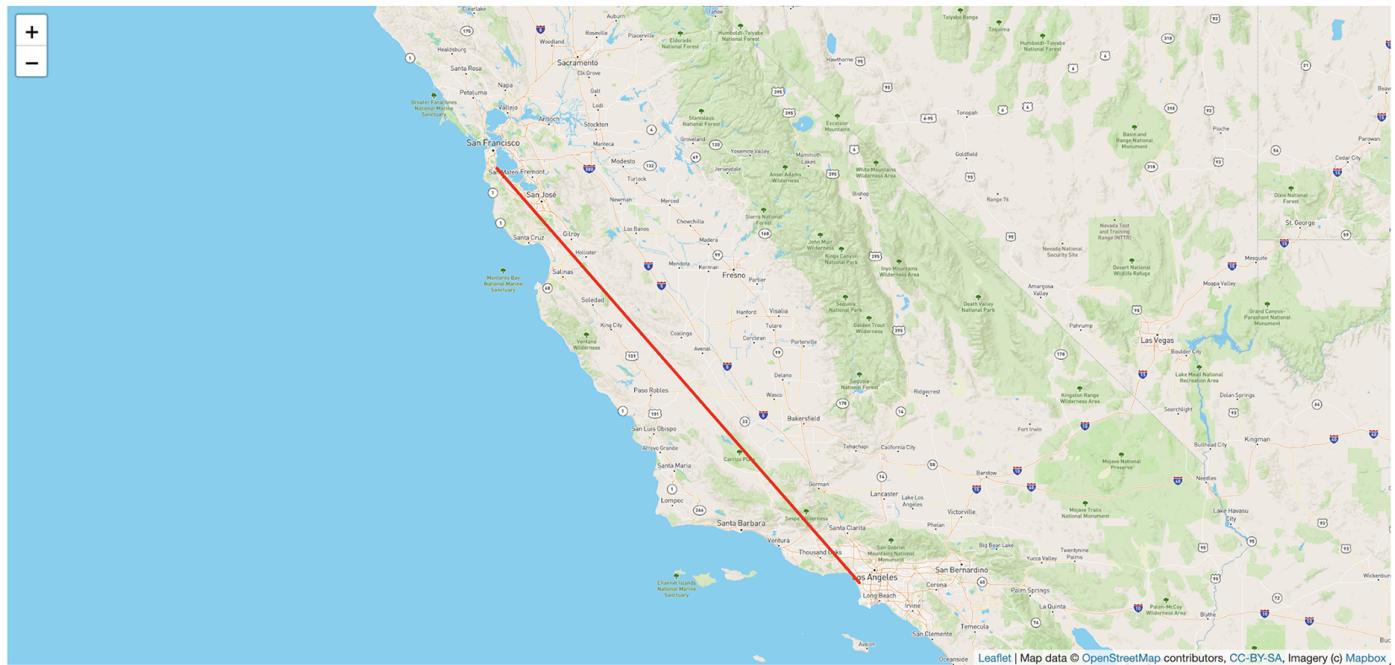
```
// Create a polyline using the line coordinates and make the line red.
L.polyline(line, {
  color: "red"
}).addTo(map);
```

In the `polyline()` function, we pass the line coordinates and the key-value pair `color: "red"` to make the line red.

Save the `logic.js` file with the changes. It should look like the following:

```
4 // We create the map object with options.  
5 let map = L.map('mapid').setView([36.1733, -120.1794], 7);  
6  
7 // Coordinates for each point to be used in the polyline  
8 let line = [  
9   [33.9416, -118.4085],  
10  [37.6213, -122.3790]  
11];  
12  
13 // Create a polyline using the line coordinates and make the line red.  
14 L.polyline(line, {  
15   color: "red"  
16 }).addTo(map);
```

When you open the `index.html` file in your browser, your map should have a red line between LAX and SFO.



Now we'll add a few more stops on our airline route.

## Map Multiple Lines

Let's edit the `logic.js` file and add two more airport stops to our `line` variable: Salt Lake City International Airport (SLC) and Seattle-Tacoma International Airport (SEA). Follow these steps:

1. Edit the `line` variable in the `logic.js` file so that it includes the two new sets of coordinates.

```
// Coordinates for each point to be used in the polyline.  
let line = [  
  [33.9416, -118.4085],  
  [37.6213, -122.3790],  
  [40.7899, -111.9791],  
  [47.4502, -122.3088]  
];
```

2. Make the line yellow by editing the value for the "color" key in the `polyline()` function to `yellow`.

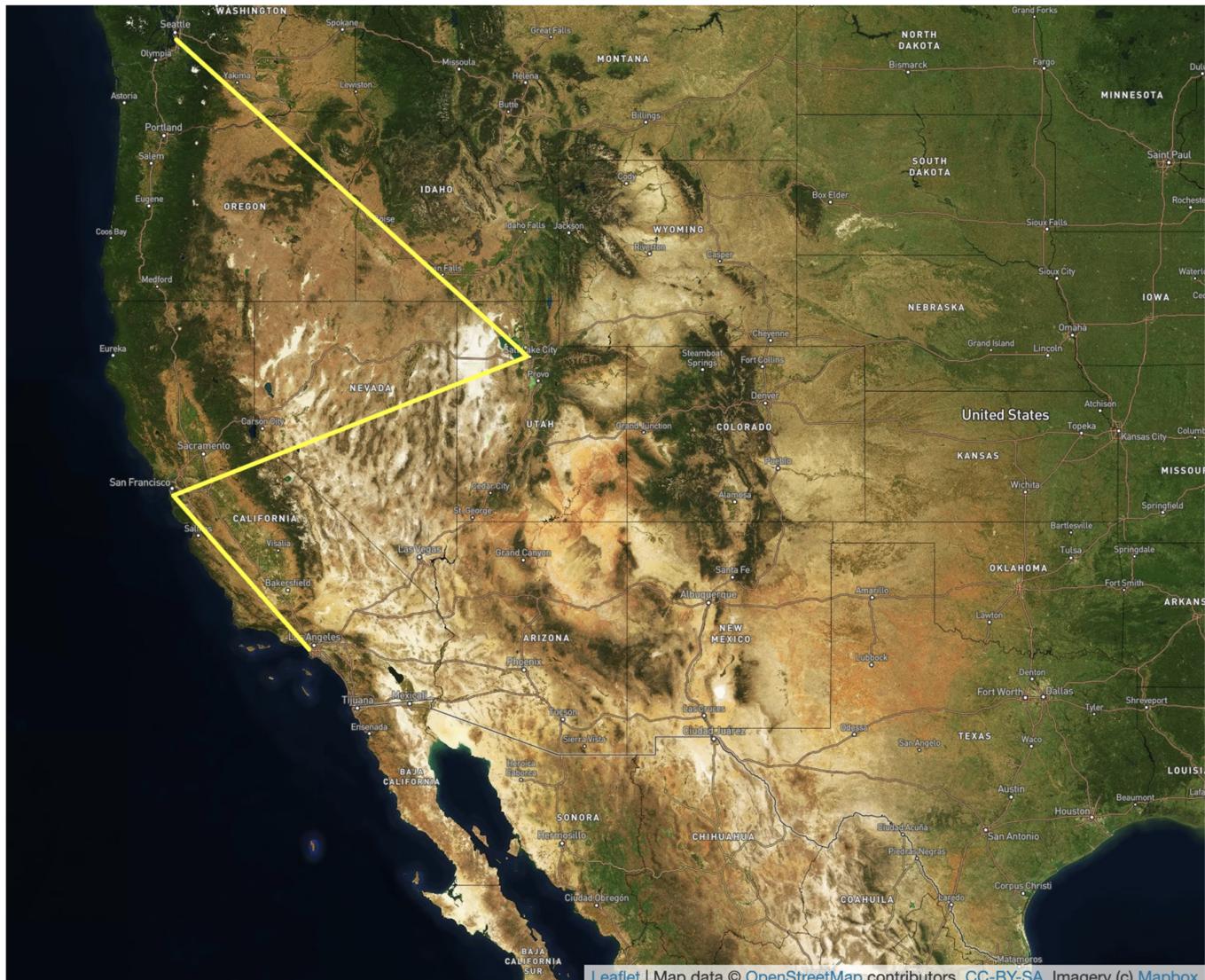
```
// Create a polyline using the line coordinates and make the line yellow.  
L.polyline(line, {  
  color: "yellow"  
}).addTo(map);
```

3. Change the map style to "satellite-streets-v11."

4. Finally, change the center of the map to SFO and change the zoom to 5 so that we can see the line.

```
// Create the map object with center at the San Francisco airport.  
let map = L.map('mapid').setView([37.6213, -122.3790], 5);
```

After you save the `logic.js` file and open the `index.html` file in your browser, your map should look like the following, showing the route from LAX, SFO, SLC, and SEA:



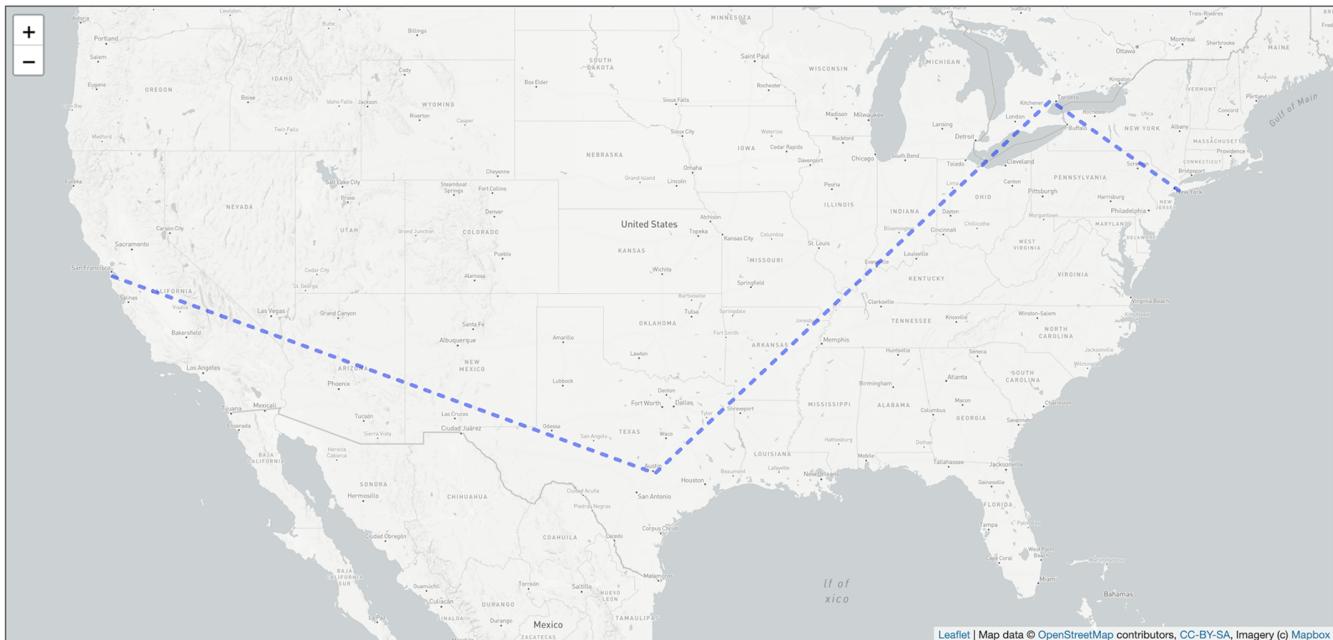
## SKILL DRILL

Edit your [logic.js](#) to create an airline route from SFO to John F. Kennedy International Airport (JFK) with two stops, Austin-Bergstrom International Airport (AUS) and Toronto Pearson International Airport (YYZ). Make the route a blue dashed line, with a weight of 4 and opacity of 0.5 on the light map.

**Hint:** You'll need to find the coordinates for some of these airports.

**Bonus:** Add your city or another city as a stopping point.

Your map should look similar to the following:



Great job on mapping routes on your map!

## ADD/COMMIT/PUSH

Add, commit, and push your changes to your Mapping\_Lines branch. Don't delete the branch so that others can use it to learn how to map lines.

After you push your changes to the branch, Sadhana will show you how to plot data from a GeoJSON ([.json](#)) file.

## NOTE

For more information, see the [Leaflet documentation on the polyline\(\) function](#) (<https://leafletjs.com/SlavaUkraini/reference-1.6.0.html#polyline>).