

13.4.1 Map a Single Point

Creating a simple Leaflet map was relatively straightforward. Now Sadhana will show you how to add a single marker to a map and change the radius of the marker. However, she would like you to create a branch for adding points to a map for the GitHub repository so that new interns and employees can use this as a tutorial.

Now that we can create a simple Leaflet map, we can plot data on the map. First, let's create a new branch. Sadhana suggests that we name this branch "Mapping_Single_Points" since we'll map single points.

REWIND

Follow these steps to create a branch off of the main branch:

1. Navigate to your repository on your computer.
2. Make sure you're on the main branch by typing: `git branch`
3. If you're not on the main branch, type: `git checkout main`
4. Pull the changes from the main branch by typing: `git pull`
5. Create a new branch by typing: `git checkout -b [name_of_your_new_branch]`

In your new branch, we'll add a new folder inside the Mapping_Earthquakes folder. Since we're going to work with the same file names in the same folder structure, we'll use the same folder structure as we did for the Simple_Leaflet_Map branch.

Set up the folder structure as follows:

- Mapping_Single_Points
 - `index.html`
 - static
 - CSS
 - `style.css`
 - js
 - `config.js`
 - `logic.js`

The two files that we'll change most often are the `index.html` and the `logic.js` files. Also, we might add an external file in the js folder. After checking out the new Mapping_Single_Points branch, copy all files from your Simple_Leaflet_Map folder and add them to a new Mapping_Single_Points folder..

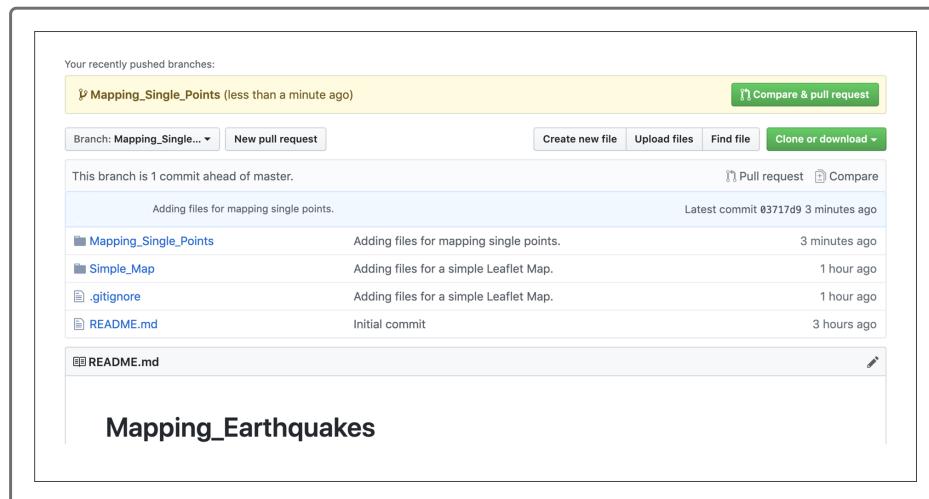
Next, push the latest changes to the Mapping_Single_Points branch to GitHub.

REWIND

Follow these steps to push changes to a new branch:

1. Type: `git status`
2. Add the folders and files by typing: `git add .`
3. Confirm the correct files will be added by typing: `git status`
4. Commit the changes by typing: `git commit -m`
5. Push the changes to the branch by typing: `git push --set-upstream origin Mapping_Single_Points`

After adding the folders and files to your Mapping_Single_Points branch, your repository should look like the following:



Next, we'll edit the `logic.js` file to add single points or markers to the basic map.

Add a Marker to the Map

Adding a marker to our simple map requires only one line of code, found in the [Leaflet Quick Start Guide](https://leafletjs.com/examples/quick-start/) (<https://leafletjs.com/examples/quick-start/>), under the "Markers, circles and polygons" subheading. Below the map is a line of code that reads as follows:

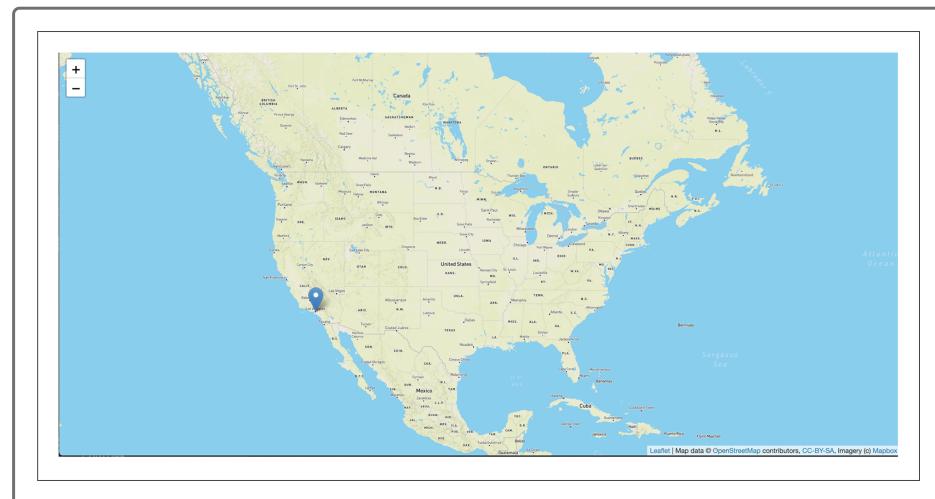
```
var marker = L.marker([51.5, -0.09]).addTo(map);
```

We're going to edit this line of code with the latitude and longitude for Los Angeles, California, and add it to our `logic.js` file that we used to create a simple map.

Open up the `logic.js` file using VS Code and add the following line of code before our `tileLayer()` code, and save the `logic.js` file:

```
// Add a marker to the map for Los Angeles, California.
let marker = L.marker([34.0522, -118.2437]).addTo(map);
```

Next, open the `index.html` file in your browser. Your map should look like the following:



Next, we'll change the marker to a circle.

Add a Circle to the Map

To change the marker on our map to a point or dot, we'll use the `circle()` function. The `circle()` function will place a circle on the map at the given coordinates. The syntax for using the `circle()` function is as follows:

```
L.circle([34.0522, -118.2437], {  
    radius: 100  
}).addTo(map);
```

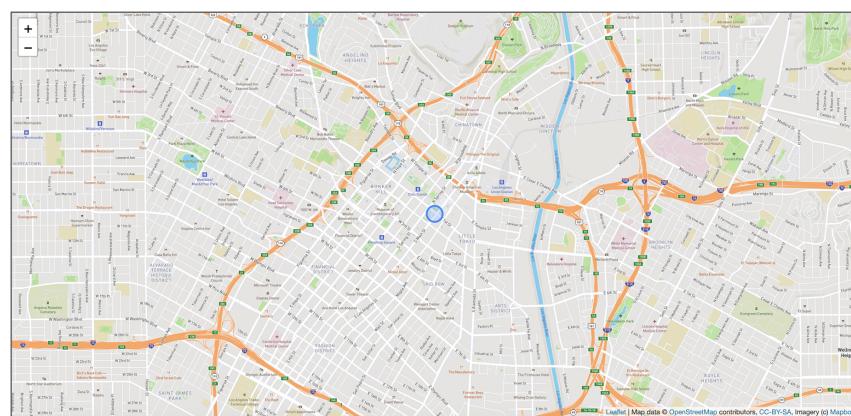
When using the `circle()` function, the default is just a small dot on the map, but we want to adjust the radius so that it's bigger and easier to see. The radius for the `circle()` function is measured in meters.

For the code above, add a circle with a 100-meter radius over Central Los Angeles when we assign a value to the `radius` key in the `circle()` function.

Copy the code for the `circle` function and replace it with the `marker()` function we used previously. We're also going to zoom in to a level of 14 on the `setView()` method. After editing your `logic.js` file, it should look like the following:

```
4 // Create the map object with a center and zoom level.  
5 let map = L.map('mapid').setView([34.0522, -118.2437], 14);  
6  
7 // Add a marker to the map for Los Angeles, California.  
8 L.circle([34.0522, -118.2437], {  
9   radius: 100  
0 }).addTo(map);
```

When we open our [index.html](#) file in our browser, it will show a circle over Central Los Angeles.

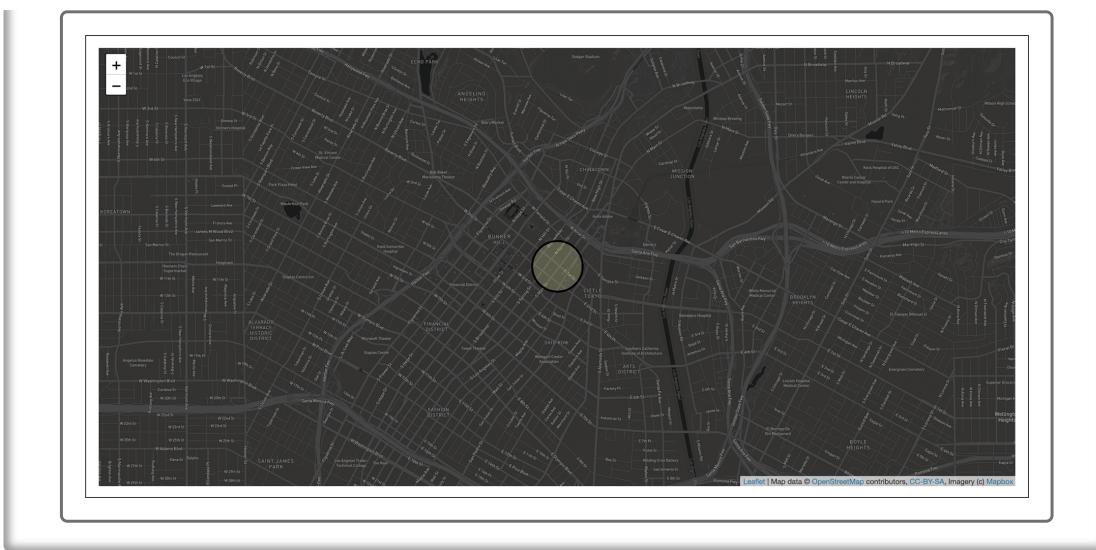


Now test your skills in the following Skill Drill:

SKILL DRILL

Using the Leaflet documentation, create a light-yellow circle with black lines indicating a 300-meter radius of Central Los Angeles on a dark map.

Your map should look like the following:



Alternatively, we can create a circle using the `circleMarker()` function. The `circleMarker()` function measures the radius of the circle in pixels, with the default radius set at 10 pixels. The syntax for using the `circleMarker()` function follows:

```
L.circleMarker([34.0522, -118.2437]).addTo(map);
```

Let's create a light-yellow circle with black lines indicating a 300-pixel radius on a dark map. Edit your `logic.js` file from the previous Skill Drill by changing your `circle()` function to a `circleMarker()` function. Your `logic.js` file should now look like the following:

```
// Add a circle to the map
L.circleMarker([34.0522, -118.2437], {
  radius: 300,
  color: "black",
  fillColor: '#ffffa1'
}).addTo(map);
```

If you didn't get the correct map style in the Skill Drill, replace the "streets-v11" in our `tileLayer()` code with "dark-v10" to look like the following:

```
// We create the tile layer that will be the background of our map.  
let streets = L.tileLayer('https://api.mapbox.com/styles/v1/mapbox/dark-v10/')
```

Save your `logic.js` file and open your `index.html` file in our browser. The circle will show a 300-pixel radius of Central Los Angeles.



Wow! What a big difference between the `circle()` and `circleMarker()` functions.

Remember, it's a best practice to commit early and often! Before you commit your code for the Mapping_Single_Points branch to GitHub, check to see if all the files will be tracked in the branch.

In the Mapping_Single_Points branch on the command line, type `git status` and you'll see that the `logic.js` file will be tracked:



Great job! Now, commit and push these files to the Mapping_Single_Points branch. Don't delete the branch, so that others can use it to learn how to map single points.

Next, Sadhana is going to show you how to add multiple locations to a map and change the radius of each marker.

NOTE

Use the links below to learn more about these Leaflet functions:

- **marker() function** [\(https://leafletjs.com/reference-1.6.0.html#marker\)](https://leafletjs.com/reference-1.6.0.html#marker)
- **circle() function** [\(https://leafletjs.com/reference-1.6.0.html#circle\)](https://leafletjs.com/reference-1.6.0.html#circle)
- **circleMarker() function** [\(https://leafletjs.com/reference-1.6.0.html#circlemarker\)](https://leafletjs.com/reference-1.6.0.html#circlemarker)

© 2020 - 2022 Trilogy Education Services, a 2U, Inc. brand. All Rights Reserved.