Create Interactive Maps with Leaflet (JavaScript library)

Morgan Ostrander
November 24, 2020

Agenda

- Create GitHub account
- Create Leaflet choropleth web map
- Host map on GitHub
- Questions

What you'll need:

- GitHub account
- Email access (to verify your new account)
- Code editor of your choice:
  - Brackets
  - Atom
  - Notepad ++

Workshop materials

Please visit the Dropbox link below to access the code, sample data file, and slides (link in chat):

https://www.dropbox.com/sh/h4euplkb76hg7lk/AACSmzFq9bzLMPWhggoJlZ-fa?dl=0

Using these resources

I'm sharing these resources under a Creative Commons license (CC-BY-NC-SA), in the spirit of making this information free and accessible to others who may be interested.

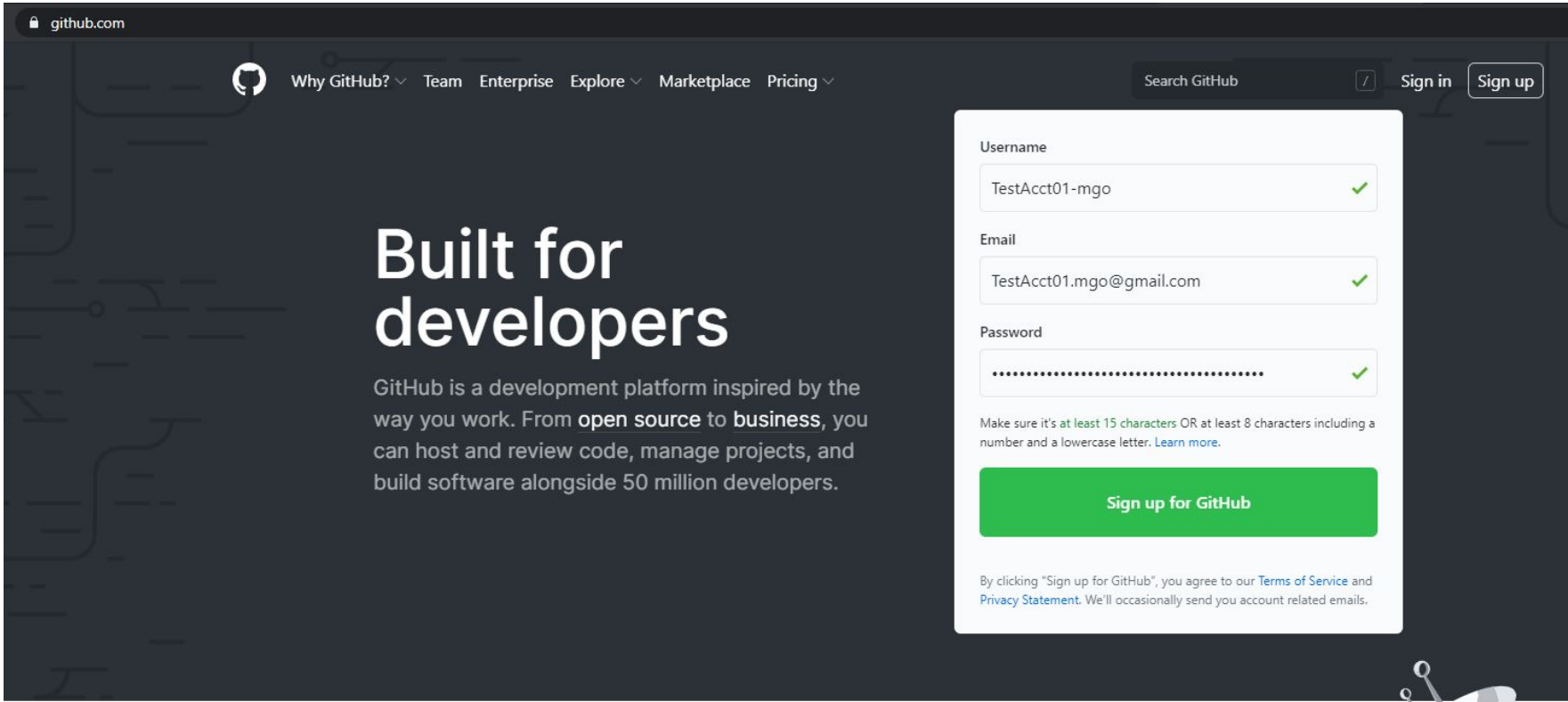
The key points:

- Feel free to **share and build on this material**, but please **credit me** if you do.
- Use and distribute this material for **non-commercial purposes**.
- If you build on this material, **share alike:** distribute with the same terms as this license.

POLL:

What is your experience with programming/development?

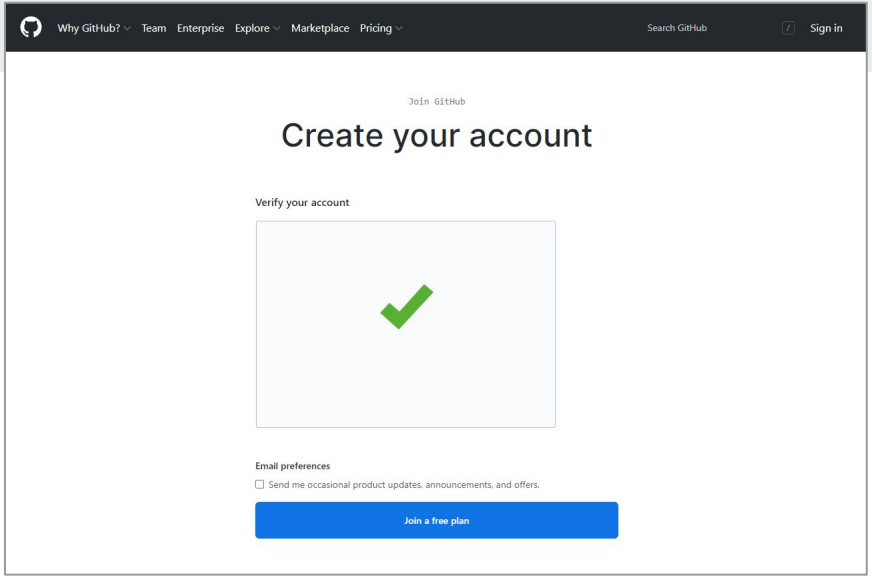
Sign up for GitHub

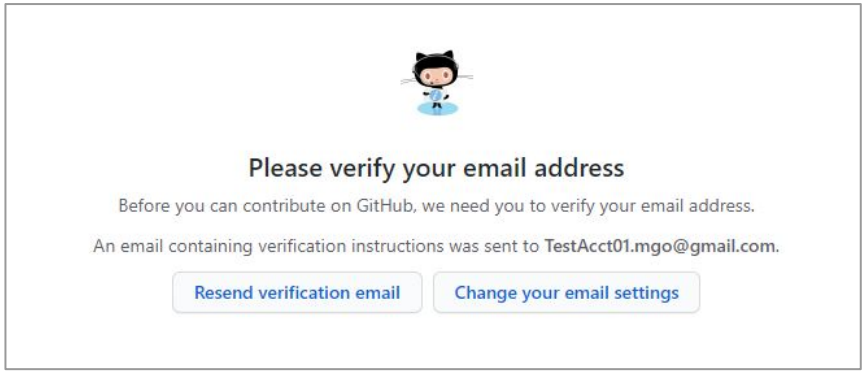


Verify your account

As you create your account, follow the prompts and click “Join a free plan”.

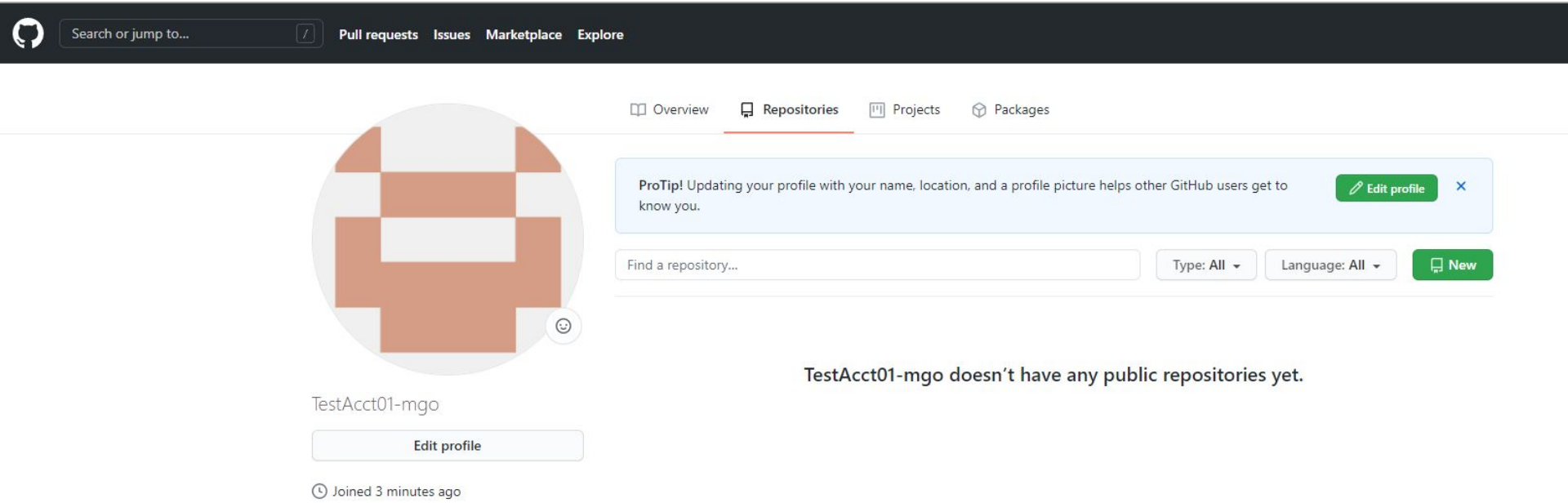
You’ll also be asked to verify your email address. Sign in to your email and find the verification message. Click the link to confirm your account.





Create a repository

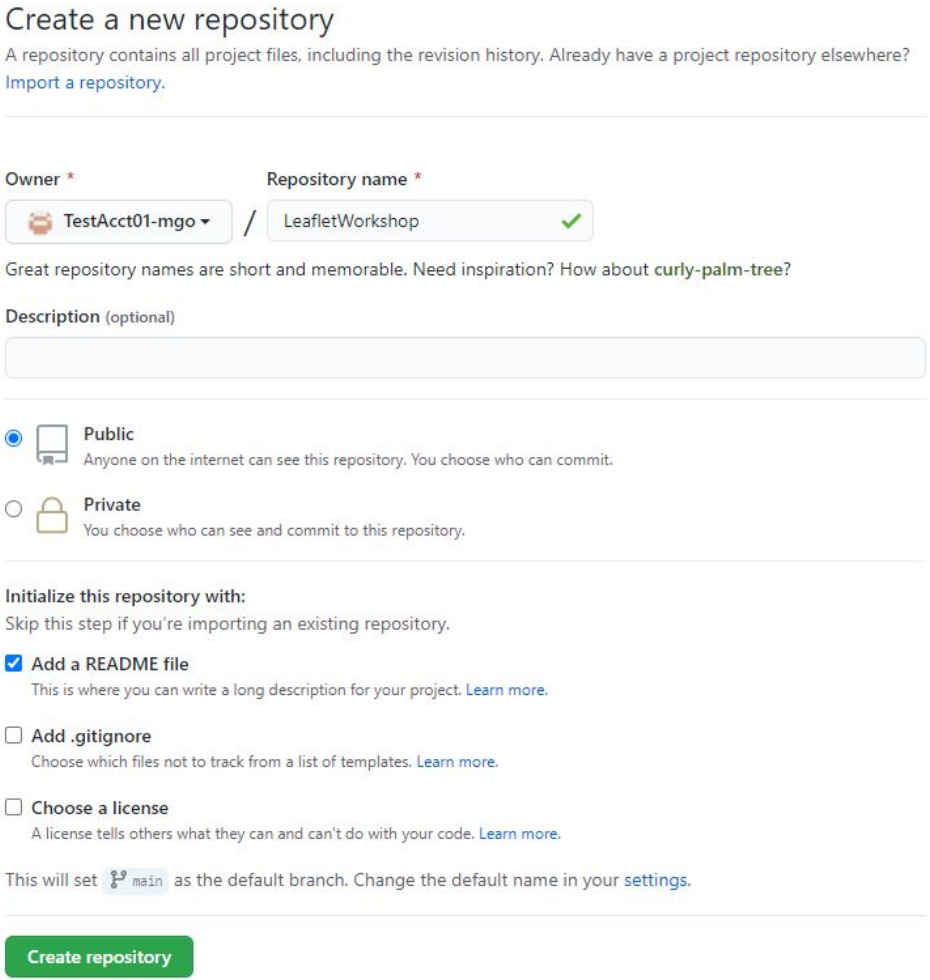
You're in! Now we'll create a repository to store our code and data files.



Create a repository

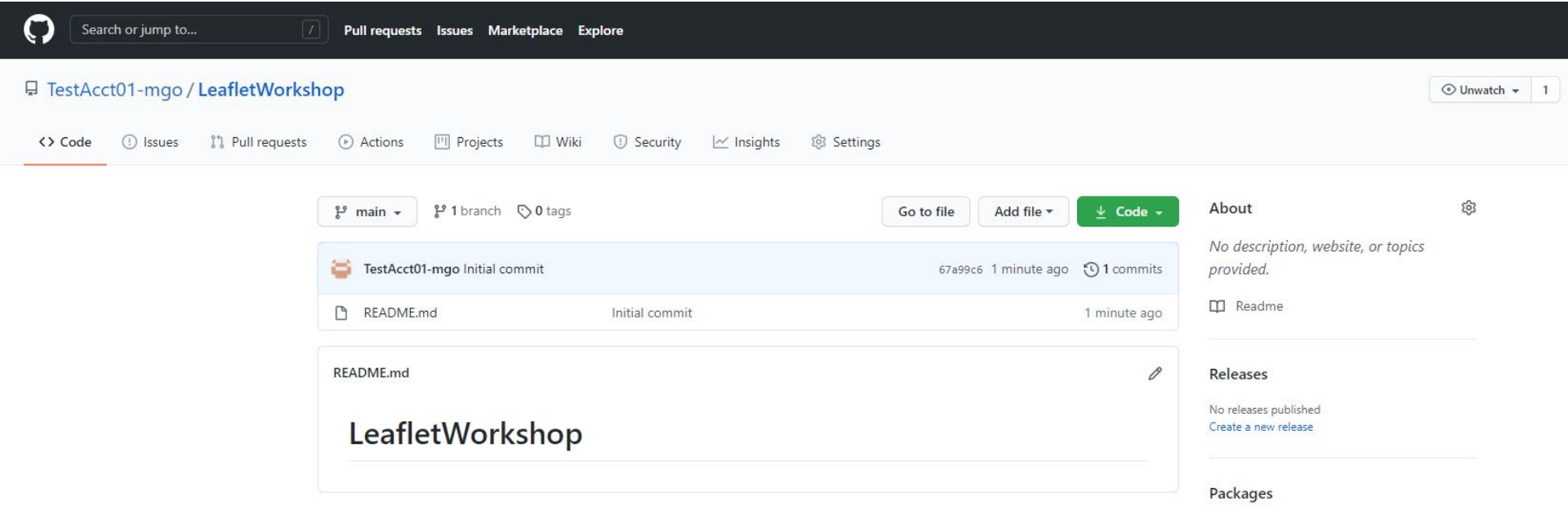
Choose your repository name and add a description if you like.

You can add a README file to include more detailed information about your project.



Create a repository

You’re done! Your repository is all set up to host your code and data files.



HTML, CSS, & JavaScript

Leaflet is a JavaScript library and requires HTML and CSS to render a web map.

HTML: Provides the structure of a web page.

- *In this project:* Controls the map container & references JavaScript/CSS to render the map.

CSS: Controls the style of a web page (fonts, colours, layout, responsive design).

- *In this project:* Controls the styling of the legend (font, background colours, spacing, etc.).

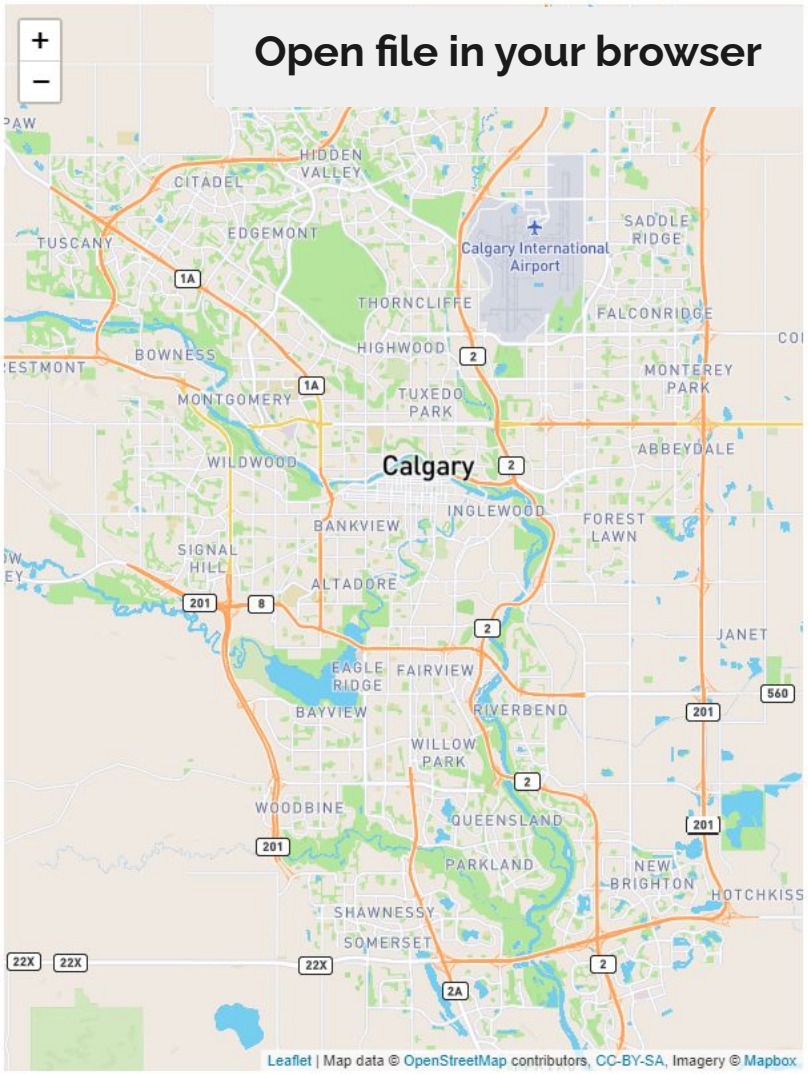
JavaScript: Facilitates interactive/complex website elements and additional functionality.

- *In this project:* The Leaflet JS library provides specialized mapping functionality. Styling (e.g., colours, transparency) is also supported.

Exploring the starter code

01_index_starter.html

```html
<!DOCTYPE html>
<html>
<head>

    <title>Leaflet Starter Code - GoGeomatics Workshop (Nov 24, 2020)</title>

    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0">


    <link rel="stylesheet" href="https://unpkg.com/leaflet@1.7.1/dist/leaflet.css" integrity="sha512-xodZBNTC5n17Xt2atTPuE1HxjVMSvLVW9ocqUKLsCC5CXdbqCmblAshOMAS6/keqq/sMZMZ19scR4PsZChSR7A==" crossorigin=""/>



    <!-- Make sure you put this AFTER Leaflet's CSS -->
    <script src="https://unpkg.com/leaflet@1.7.1/dist/leaflet.js" integrity="sha512-XQoYMqMTK8LvdxXYG3nZ448hOEQiglfqkJs1NOQV44cWnUrBc8PkAOcXy20w0vlaXaVUearIOBhiXZ5V3ynxwA==" crossorigin=""></script>


</head>

<body>


<div id="mapid" style="width: 600px; height: 800px;"></div>
<script>

    // Initialize map and set its view to Calgary, Alberta

    var mymap = L.map('mapid').setView([51.0300, -114.0750], 11);



    // Add the Mapbox Streets tile layer to the map

    L.tileLayer('https://api.mapbox.com/styles/v1/{id}/tiles/{z}/{x}/{y}?access_token=pk.eyJ1IjoibWFwYm94IiwiYSI6ImNpejY4NXVycTA2emYycXBndHRqcmZ3N3gifQ.rJcFIG214AriISLbB6B5aw', {
        maxZoom: 18,
        attribution: 'Map data &copy; <a href="https://www.openstreetmap.org/">OpenStreetMap</a> contributors, ' +
            '<a href="https://creativecommons.org/licenses/by-sa/2.0/">CC-BY-SA</a>, ' +
            'Imagery © <a href="https://www.mapbox.com/">Mapbox</a>',
        id: 'mapbox/streets-v11',
        tileSize: 512,
        zoomOffset: -1
    }).addTo(mymap);
```




```
<!DOCTYPE html>
<html>
<head>

    <title>Leaflet Starter Code - GoGeomatics Workshop (Nov 24, 2020)</title>

    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0">


    <link rel="stylesheet" href="https://unpkg.com/leaflet@1.7.1/dist/leaflet.css" integrity="sha512-xodZBNTC5n17Xt2atTPuE1HxjVMSvLVW9ocqUKLsCC5CXdbqCmblAshOMAS6/keqq/sMZMZ19scR4PsZChSR7A==" crossorigin=""/>



    <!-- Make sure you put this AFTER Leaflet's CSS -->
    <script src="https://unpkg.com/leaflet@1.7.1/dist/leaflet.js" integrity="sha512-XQoYMqMTK8LvdxXYG3nZ448hOEQiglfqkJs1NOQV44cWnUrBc8PkAOcXy20w0vlaXaVUearIOBhiXZ5V3ynxwA==" crossorigin=""></script>


</head>

<body>


<div id="mapid" style="width: 600px; height: 800px;"></div>
<script>

    // Initialize map and set its view to Calgary, Alberta

    var mymap = L.map('mapid').setView([51.0300, -114.0750], 11);



    // Add the Mapbox Streets tile layer to the map

    L.tileLayer('https://api.mapbox.com/styles/v1/{id}/tiles/{z}/{x}/{y}?access_token=pk.eyJ1IjoibWFwYm94IiwiYSI6ImNpejY4NXVycTA2emYycXBndHRqcmZ3N3gifQ.rJcFIG214AriISLbB6B5aw', {
        maxZoom: 18,
        attribution: 'Map data &copy; <a href="https://www.openstreetmap.org/">OpenStreetMap</a> contributors, ' +
            '<a href="https://creativecommons.org/licenses/by-sa/2.0/">CC-BY-SA</a>, ' +
            'Imagery © <a href="https://www.mapbox.com/">Mapbox</a>',
        id: 'mapbox/streets-v11',
        tileSize: 512,
        zoomOffset: -1
    }).addTo(mymap);
```

<head>

This section contains metadata.

It sets document properties like the **character set** and references the **Leaflet CSS and JavaScript files** that are hosted online.

```
<!DOCTYPE html>
<html>
<head>

    <title>Leaflet Starter Code - GoGeomatics Workshop (Nov 24, 2020)</title>

    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0">


    <link rel="stylesheet" href="https://unpkg.com/leaflet@1.7.1/dist/leaflet.css" integrity="sha512-xodZBNTC5n17Xt2atTPuE1HxjVMSvLVW9ocqUKLsCC5CXdbqCmblAshOMAS6/keqq/sMZMZ19scR4PsZChSR7A==" crossorigin=""/>



    <!-- Make sure you put this AFTER Leaflet's CSS -->
    <script src="https://unpkg.com/leaflet@1.7.1/dist/leaflet.js" integrity="sha512-XQoYMqMTK8LvdxXYG3nZ448hOEQiglfqkJs1NOQV44cWnUrBc8PkAOcXy20w0vlaXaVUearIOBhiXZ5V3ynxwA==" crossorigin=""></script>


</head>

```

- **<link rel="stylesheet" href="https://unpkg.com/leaflet@1.7.1/dist/leaflet.css"...**
  Leaflet CSS file (controls Leaflet's styling/formatting specifications)

- **<script src="https://unpkg.com/leaflet@1.7.1/dist/leaflet.js"...**
  Leaflet JavaScript file (controls Leaflet functionality)

<body>

This section controls the web map itself - the size, layers, and functionality.

Try repositioning the map to a new location!

```
<body>


<div id="mapid" style="width: 600px; height: 800px;"></div>
<script>

    // Initialize map and set its view to Calgary, Alberta

    var mymap = L.map('mapid').setView([51.0300, -114.0750], 11);



    // Add the Mapbox Streets tile layer to the map

    L.tileLayer('https://api.mapbox.com/styles/v1/{id}/tiles/{z}/{x}/{y}?
    access_token=pk.eyJ1IjoibWFwYm94IiwiYSI6ImNpejY4NXVycTA2emYycXBndHRqcmZ3N3gifQ.rJcFIG214AriISLbB6
    B5aw', {
        maxZoom: 18,
        attribution: 'Map data &copy; <a href="https://www.openstreetmap.org/">OpenStreetMap</a>
        contributors, ' +
            '<a href="https://creativecommons.org/licenses/by-sa/2.0/">CC-BY-SA</a>, ' +
            'Imagery © <a href="https://www.mapbox.com/">Mapbox</a>',
        id: 'mapbox/streets-v11',
        tileSize: 512,
        zoomOffset: -1
    }).addTo(mymap);
```

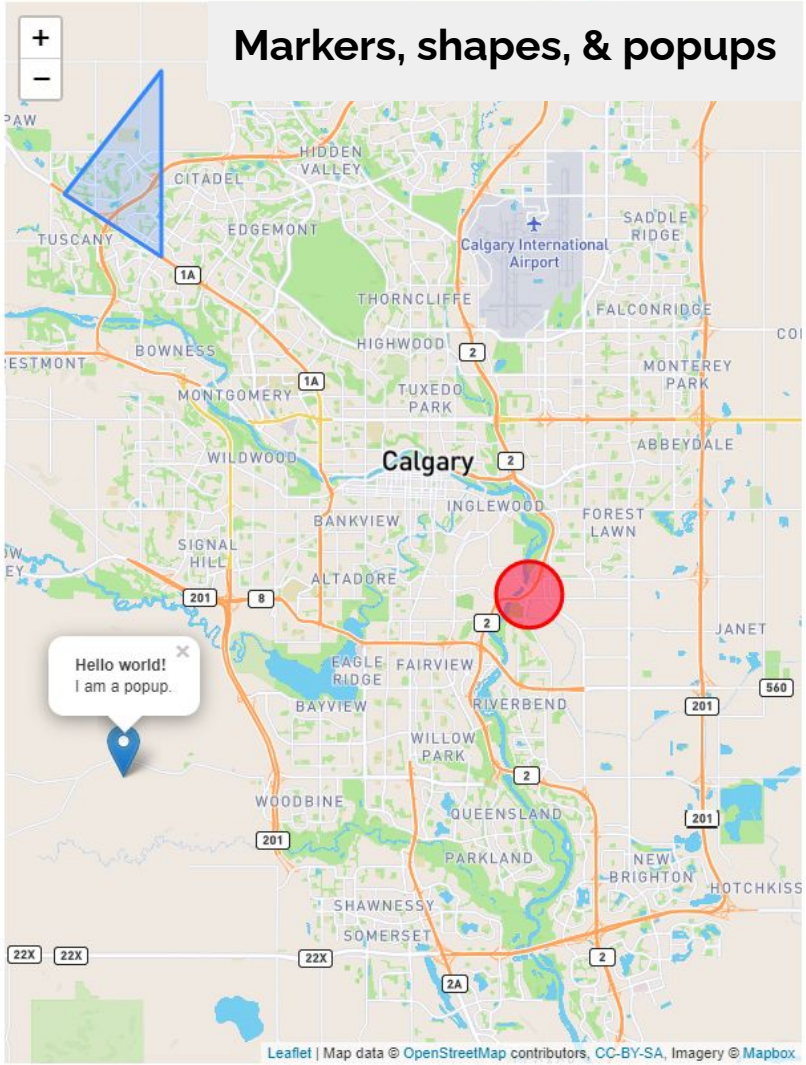
- **var mymap = L.map('mapid').setView([51.0300, -114.0750], 11);**
  Sets the map coordinates and zoom level.

- **L.tileLayer...**
  Add a tile layer of your choice and configure the parameters; in this case, the Mapbox Streets layer was used.

Syntax notes for reference

HTML		**CSS**		**JavaScript (JS)**	
<script> </script>	Denotes beginning/end of JS code	.info	“.” denotes class selector - way of identifying an HTML element	function polyStyle()	Denotes a function (package of code that is called from another part of the code).
<!-- *Comment* -->	Comment (not executable code)	/* *Comment* */	Comment (not executable code)	// *Comment*	Comment (not executable code)

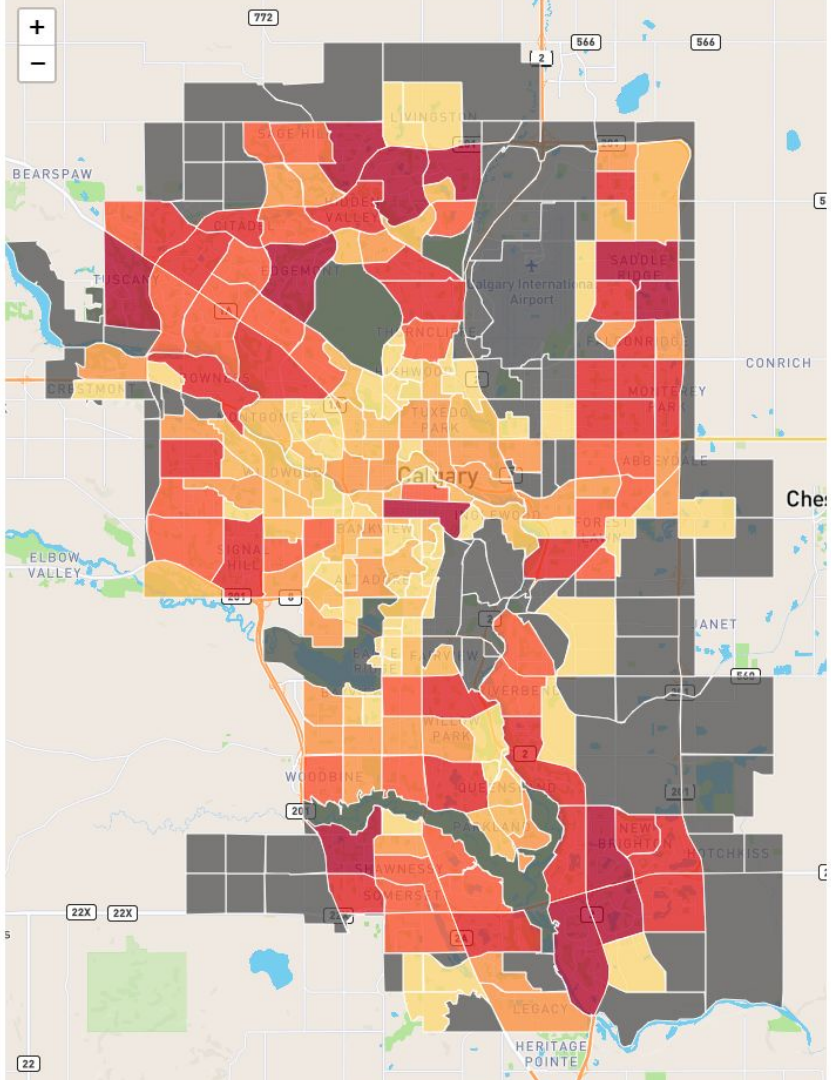
Markers, shapes, & popups




```
        maxZoom: 18,
        attribution: 'Map data &copy; <a href="https://www.openstreetmap.org/">OpenStreetMap</a>
        contributors, ' +
            '<a href="https://creativecommons.org/licenses/by-sa/2.0/">CC-BY-SA</a>, ' +
            'Imagery © <a href="https://www.mapbox.com/">Mapbox</a>',
        id: 'mapbox/streets-v11',
        tileSize: 512,
        zoomOffset: -1
    }).addTo(mymap);



    // ---------- BEGIN SECTION (Markers, circles, polygons, & popups) ------------
    // Enable the code below to show additional elements that can be added to your map.



    L.marker([50.95, -114.22]).addTo(mymap)
        .bindPopup("<b>Hello world!</b><br />I am a popup.").openPopup();



    L.circle([51.0100, -114.0100], 1200, {
        color: 'red',
        fillColor: '#f03',
        fillOpacity: 0.5
    }).addTo(mymap).bindPopup("I am a circle.");

    L.polygon([
        [51.1800, -114.20],
        [51.1200, -114.20],
        [51.1400, -114.25]
    ]).addTo(mymap).bindPopup("I am a polygon.");


    var popup = L.popup();

    function onMapClick(e) {
        popup
            .setLatLng(e.latlng)
            .setContent("You clicked the map at " + e.latlng.toString())
            .openOn(mymap);
    }

    mymap.on('click', onMapClick);



    // ---------- END SECTION (Markers, circles, polygons, & popups) ----------------

</script>
</body>
</html>
```

Adding the data

02_index_addData.html
Census_by_Community_2019.geojson



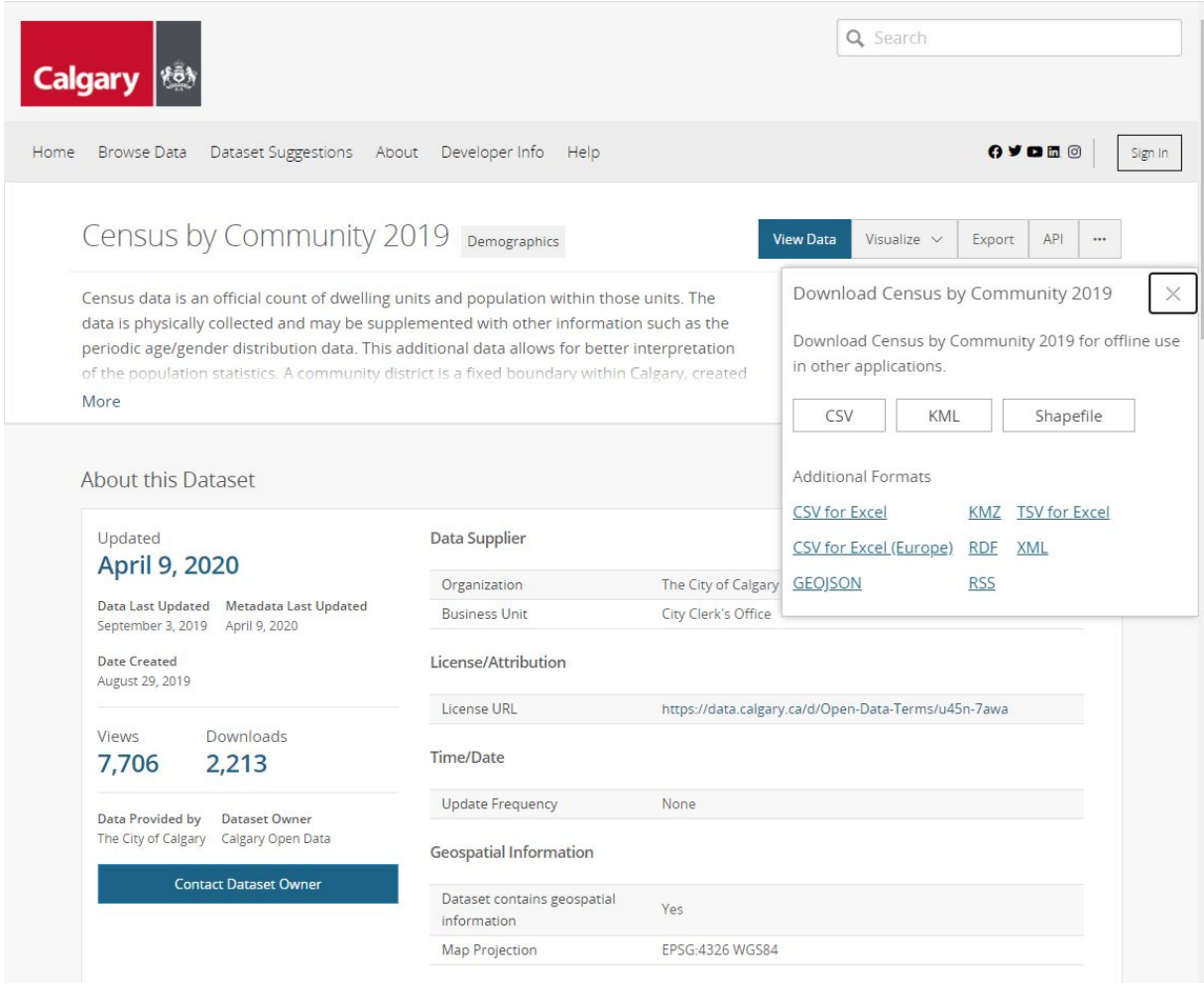
Find data

We'll need to include a GeoJSON file in our Leaflet map.

I downloaded a community census layer from the City of Calgary's website. This is the original link to the data layer:

https://data.calgary.ca/Demographics/Census-by-Community-2019/rkfr-buzb

After you've downloaded it, store it in the same folder along with your HTML Leaflet mapping file.



Add jQuery library

We'll add a reference to the jQuery library so that we can easily pull in our data layer.

NOTE: jQuery is a bit outdated, but it's a simple approach to accessing data. Feel free to use other methods to connect to your data if you wish.

```
<!DOCTYPE html>
<html>
<head>

    <title>Leaflet Starter Code - GoGeomatics Workshop (Nov 24, 2020)</title>

    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0">


    <link rel="stylesheet" href="https://unpkg.com/leaflet@1.7.1/dist/leaflet.css" integrity="sha512-xodZBNTC5n17Xt2atTPuE1HxjVMSvLVW9ocqUKLsCC5CXdbqCmblAshOMAS6/keqq/sMZMZ19scR4PsZChSR7A==" crossorigin=""/>


    <!-- Make sure you put this AFTER Leaflet's CSS -->
    <script src="https://unpkg.com/leaflet@1.7.1/dist/leaflet.js" integrity="sha512-XQoYMqMTK8LvdxXYG3nZ448hOEQiglfqkJs1NOQV44cWnUrBc8PkAOcXy20w0vlaXaVUearIOBhiXZ5V3ynxwA==" crossorigin=""></script>


    <!-- Add jQuery  -->
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>


</head>

```

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
```

This jQuery script reference points to a version of jQuery that is hosted by Google.

Get the data layer

We'll use the $.getJSON method to pull in the data layer. The $ is equivalent to "jQuery" (so the method is jQuery.getJSON()).

L.geoJson is a Leaflet class that parses data and displays it on the map.

```
36
37      // Add the Mapbox Streets tile layer to the map
38
39      L.tileLayer('https://api.mapbox.com/styles/v1/{id}/tiles/{z}/{x}/{y}?
        access_token=pk.eyJ1IjoibWFwYm94IiwiYSI6ImNpejY4NXVycTA2emYycXBndHRqcmZ3N3gifQ.rJcFIG214AriISLbB6B5aw
        ', {
40          maxZoom: 18,
41          attribution: 'Map data &copy; <a href="https://www.openstreetmap.org/">OpenStreetMap</a>
            contributors, ' +
42              '<a href="https://creativecommons.org/licenses/by-sa/2.0/">CC-BY-SA</a>, ' +
43              'Imagery © <a href="https://www.mapbox.com/">Mapbox</a>',
44          id: 'mapbox/streets-v11',
45          tileSize: 512,
46          zoomOffset: -1
47      }).addTo(mymap);
48
49
50
51      // loading GeoJSON file - https://www.igismap.com/add-load-geojson-file-point-polyline-polygon-map-
        leaflet-js/
52      $.getJSON("Census_by_Community_2019.geojson",function(data){
53          // L.geoJson function is used to parse geojson file and load on to map
54          L.geoJson(data, {
55              style: polyStyle
56          }).addTo(mymap);
57      });
58
59
60
```

```
$.getJSON("Census_by_Community_2019.geojson",function(data){
    L.geoJson(data, {
        style: polyStyle
    }).addTo(mymap);
});
```

polyStyle *will control the colours of the polygons - we'll define this in the next step.*

Style the data

We'll add two functions:

colourBreaks:

- Defines the data breaks and assigns a colour to each range.
- **Substitute different HTML colour codes to define colours of your choice!**

polyStyle:

- Sets the colour breaks (previous function) as the fill colour for the polygons and links these styles to the RES_CNT data field.
- Sets the weight, line colours, line opacity, and fill opacity.

```
// loading GeoJSON file - https://www.igismap.com/add-load-geojson-file-point-polyline-polygon-map-leaflet-js/
$.getJSON("Census_by_Community_2019.geojson",function(data){
    // L.geoJson function is used to parse geojson file and load on to map
    L.geoJson(data, {
        style: polyStyle
    }).addTo(mymap);
});


// Define colours based on data ranges (create choropleth map)
function colourBreaks(d) {
    return d > 15000 ? '#b10026' :
           d > 10000 ? '#e31a1c' :
           d > 7000 ? '#fc4e2a' :
           d > 5000 ? '#fd8d3c' :
           d > 2500 ? '#feb24c' :
           d > 1 ? '#fed976' :
                   '#525252';
}


// Style polygons and control opacity
function polyStyle(feature) {
    return {
        weight: 1,
        fillColor: colourBreaks(feature.properties.res_cnt),
        color: '#FFFFFF',
        opacity: 1,
        fillOpacity: 0.75
    };
}
```

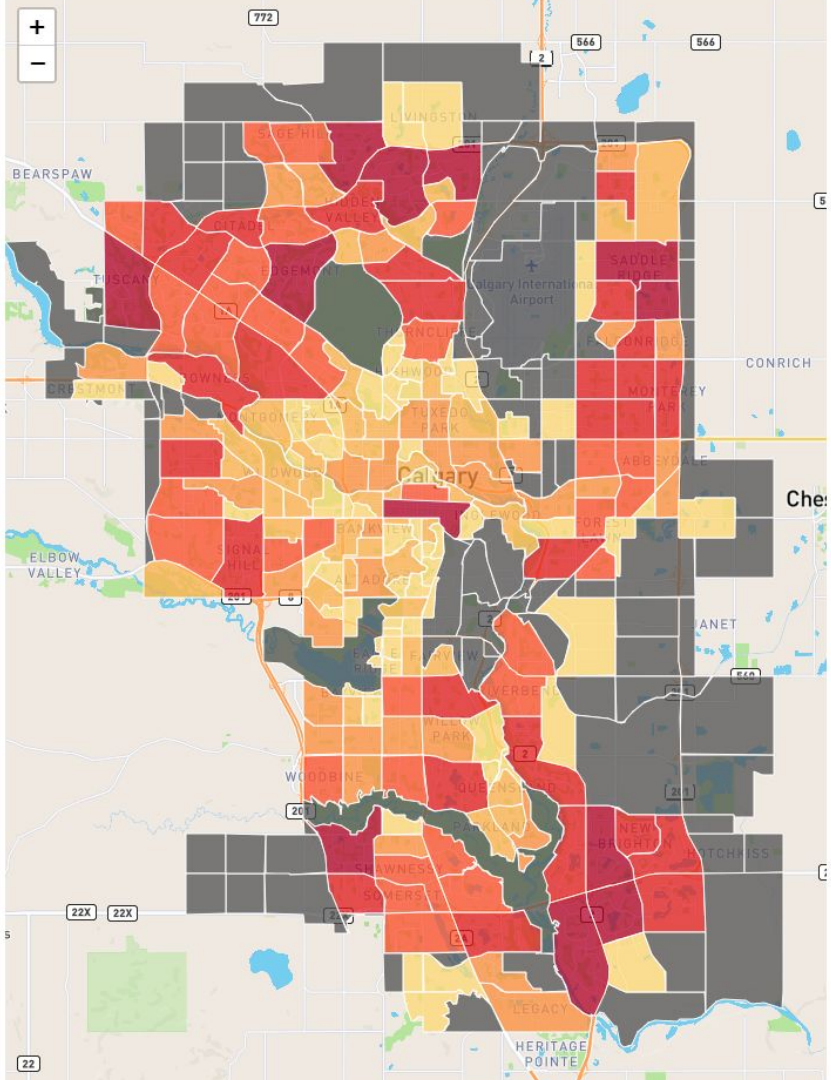
Keep res_cnt lowercase

RES_CNT	Number of residents

Test it out!

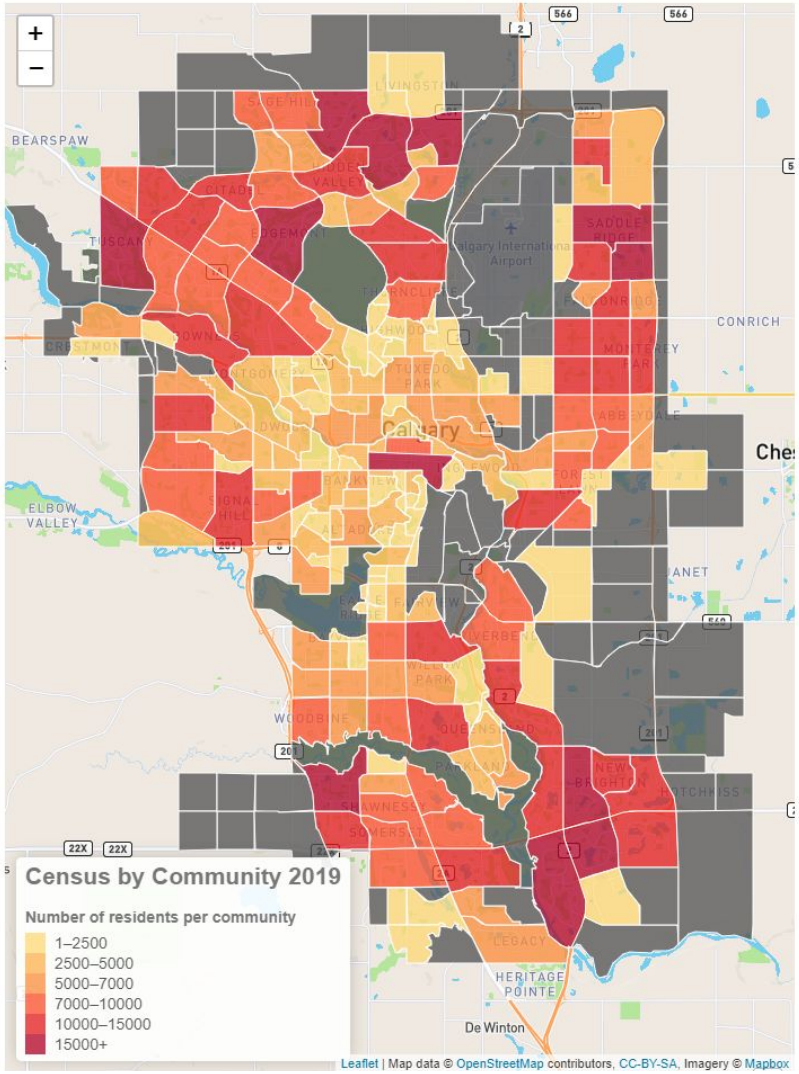
Open your index HTML file in your browser and make sure your data layer appears.

Check-in: How are you doing?



Adding a legend

03_index_addLegend.html
styles.css

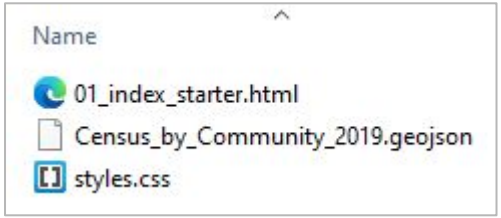


Add styles.css

We'll add a reference to an external CSS style sheet (styles.css) to control the look of our legend.

Ensure the styles.css file is saved in the same folder as your index.html file and data file.

```
<!DOCTYPE html>
<html>
<head>

    <title>Leaflet Starter Code - GoGeomatics Workshop (Nov 24, 2020)</title>

    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0">


    <link rel="stylesheet" href="https://unpkg.com/leaflet@1.7.1/dist/leaflet.css" integrity="sha512-xodZBNTC5n17Xt2atTPuE1HxjVMSvLVW9ocqUKLsCC5CXdbqCmblAshOMAS6/keqq/sMZMZ19scR4PsZChSR7A==" crossorigin=""/>

    <link rel="stylesheet" href="styles.css"/>


    <!-- Make sure you put this AFTER Leaflet's CSS -->
    <script src="https://unpkg.com/leaflet@1.7.1/dist/leaflet.js" integrity="sha512-XQoYMqMTK8LvdxXYG3nZ448hOEQiglfqkJs1NOQV44cWnUrBc8PkAOcXy20w0vlaXaVUearIOBhiXZ5V3ynxwA==" crossorigin="">
    </script>



    <!-- Add jQuery  -->
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>


</head>
```




```
<link rel="stylesheet" href="styles.css"/>
```

Add legend

This section adds the legend to the bottom left corner of the map.

It leverages the colourBreaks() function to generate the coloured squares for each range.

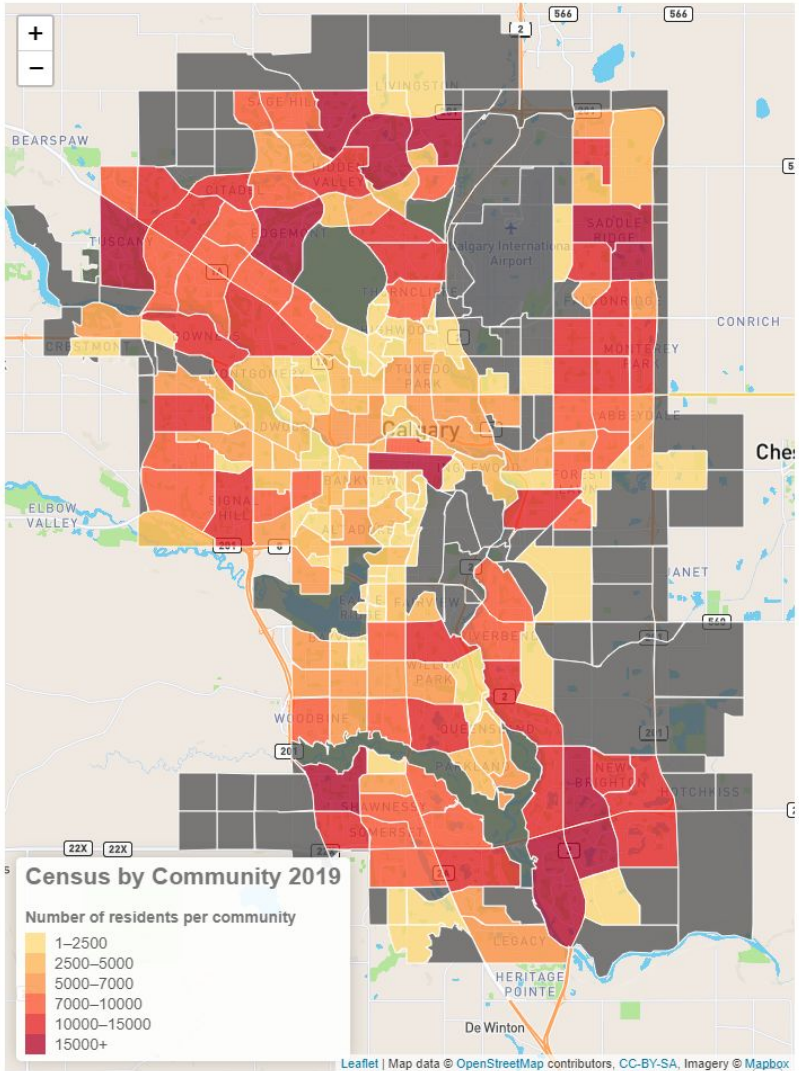
The styles.css file controls the font, background colours, and positioning.

```
function polyStyle(feature) {
    return {
        weight: 1,
        fillColor: colourBreaks(feature.properties.res_cnt),
        color: '#FFFFFF',
        opacity: 1,
        fillOpacity: 0.75
    };
}


// Add legend - modified from https://leafletjs.com/examples/choropleth/

var legend = L.control({position: 'bottomleft'});

legend.onAdd = function (mymap) {

var div = L.DomUtil.create('div', 'info legend'),
    grades = [1, 2500, 5000, 7000, 10000, 15000],
    labels = [];

div.innerHTML ='<h2>Census by Community 2019</h2>' + '<br>' + '<h4>Number of residents per community</h4>';

// loop through our density intervals and generate a label with a colored square for each interval
for (var i = 0; i < grades.length; i++) {
    div.innerHTML +=
        '<i style="background:' + colourBreaks(grades[i] + 1) + '"></i> ' +
        grades[i] + (grades[i + 1] ? '&ndash;' + grades[i + 1] + '<br>' : '+');
}

return div;
};

legend.addTo(mymap);
```

Copy the JavaScript code from *03_index_addLegend.html* and add it below the polyStyle() function.

Test it out!

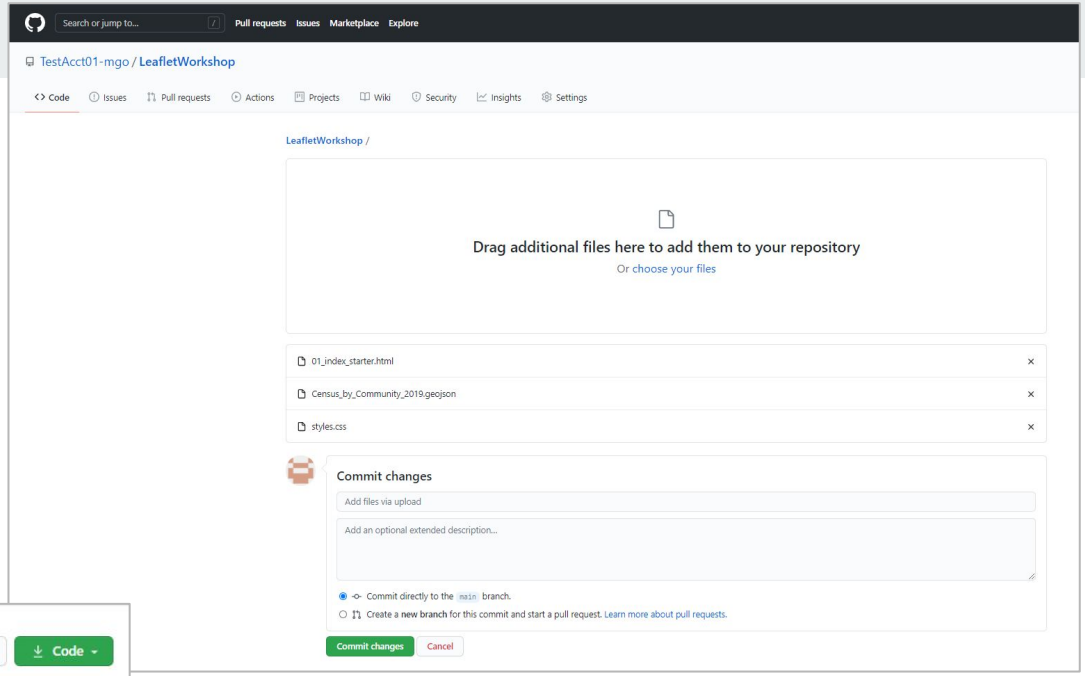
Open your index HTML file in your browser and check if your legend appears.

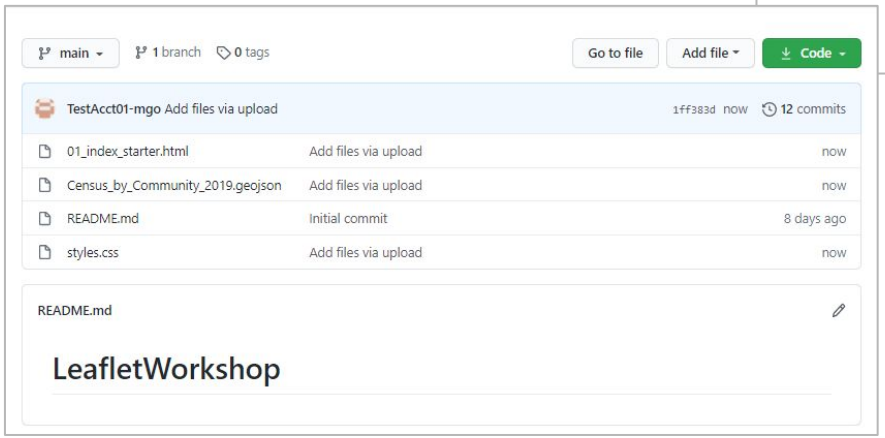


Add code to repo

Upload your code files to your repo:

- 01_index_starter.html
- Census_by_Community_2019.geojson
- styles.css

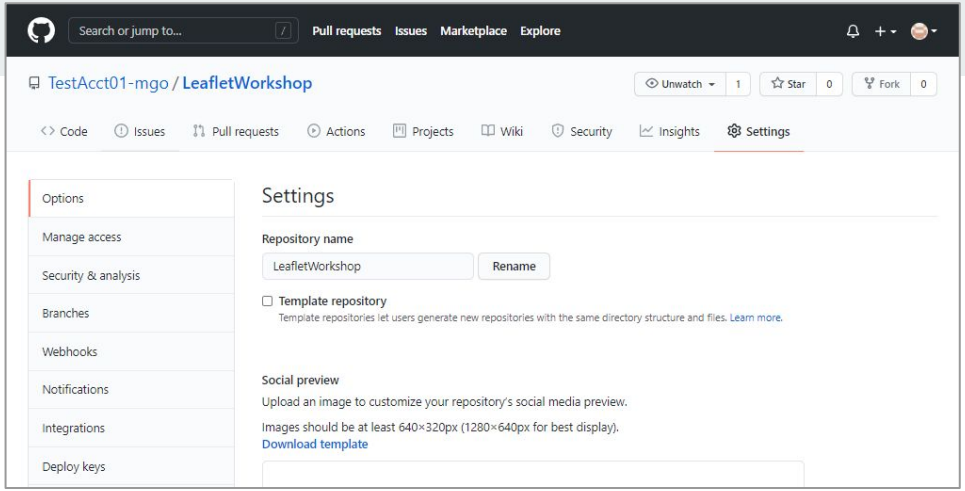


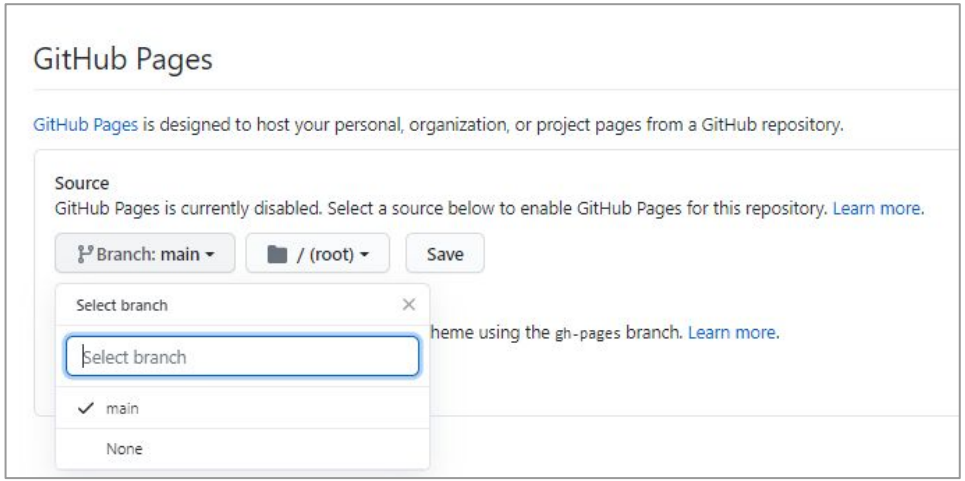


Set up GitHub Pages

To actually view the result of our HTML file (not just the source code), we will create a GitHub Pages site to render the file.

- In your repo, click Settings.
- Scroll down the page to “GitHub Pages”.
- Set the branch to “main” and the folder to “/root”. Save.

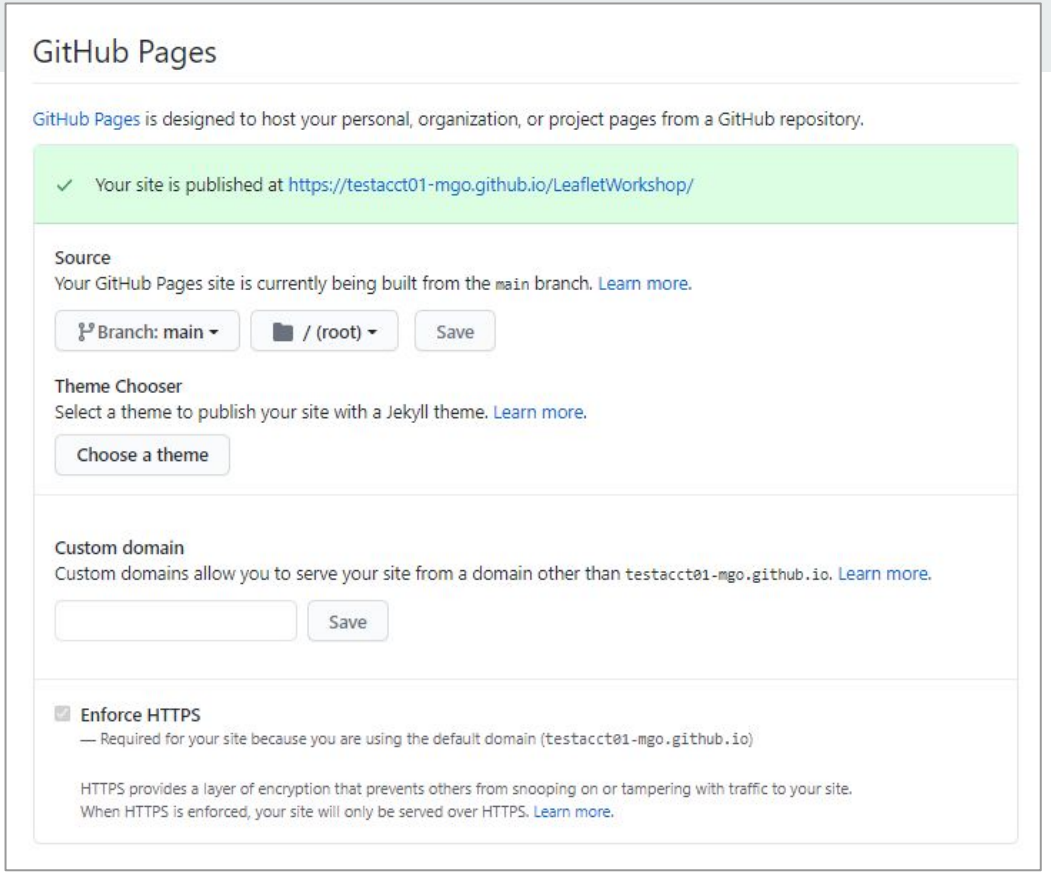




Confirm URL

Once you've saved the GitHub Pages settings, go back into settings and back down to the GitHub Pages section to confirm your site's URL.

Your site's URL should have the following format:

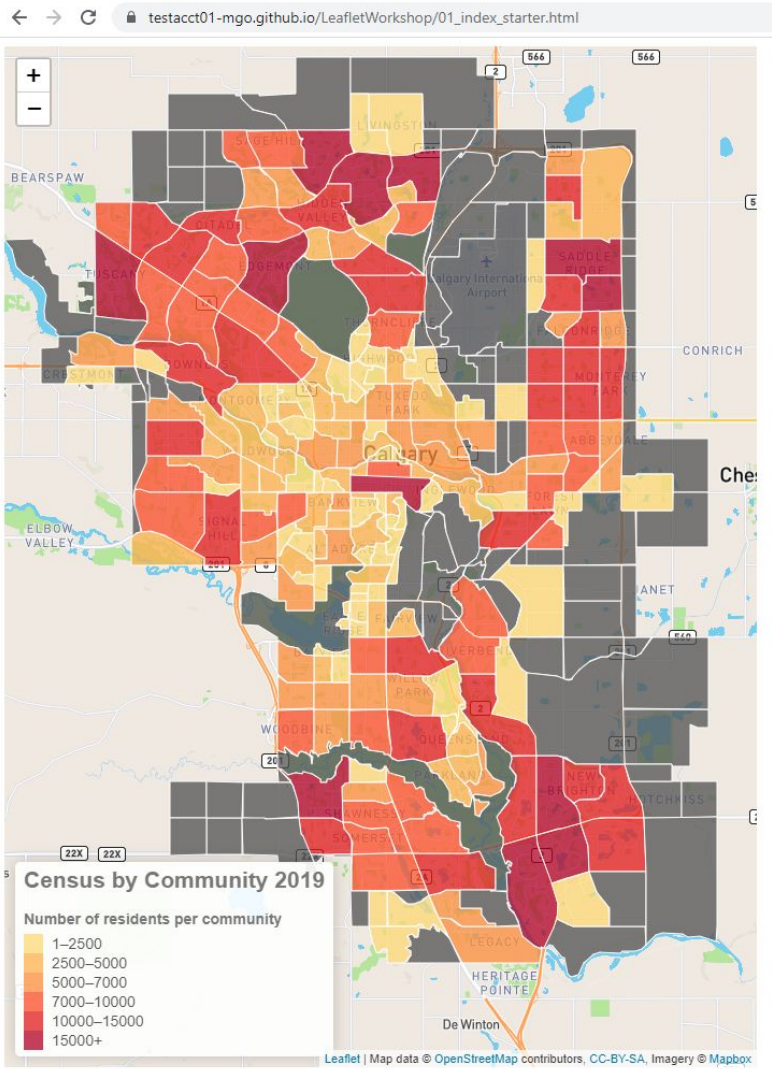


https://<user>.github.io/<repository>/<*Optional: page name*>

(e.g., https://testacct01-mgo.github.io/LeafletWorkshop/01_index_starter.html)

View your hosted map

Example:
https://testacct01-mgo.github.io/LeafletWorkshop/01_index_starter.html



Questions?