

TUTORIAL 7 | JUMP INTO MAPBOX

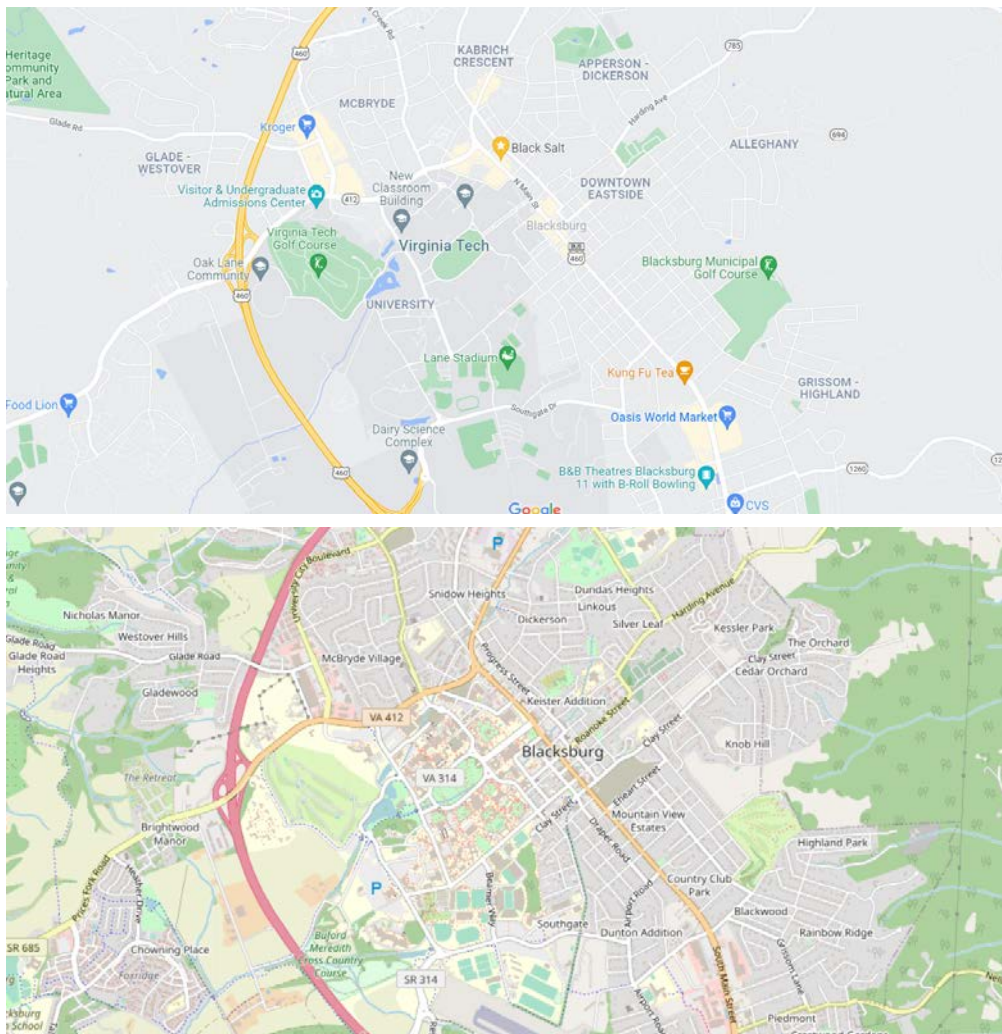
Goals

- Gain familiarity with Mapbox platform.
- Understand how to style layers and base map.
- Upload Census shapefile as data layer from QGIS.
- View map in browser.

Intro

Today, you're going to create a Mapbox base map style and add some basic data.

Think of your basemap as a styled map background. For example, take a look at the difference between Google Maps and Open Street Maps basemaps. Notice the difference in color, labeling, lineweight, information density, and so on. Your basemap is the canvas for information that you want to show. Google Maps emphasizes roads and points of interest – a car-based perspective. OSM, on the other hand, shows a less hierarchical urban fabric of land use as well as roads and places.



Mapbox terms:

- **Style:** base map

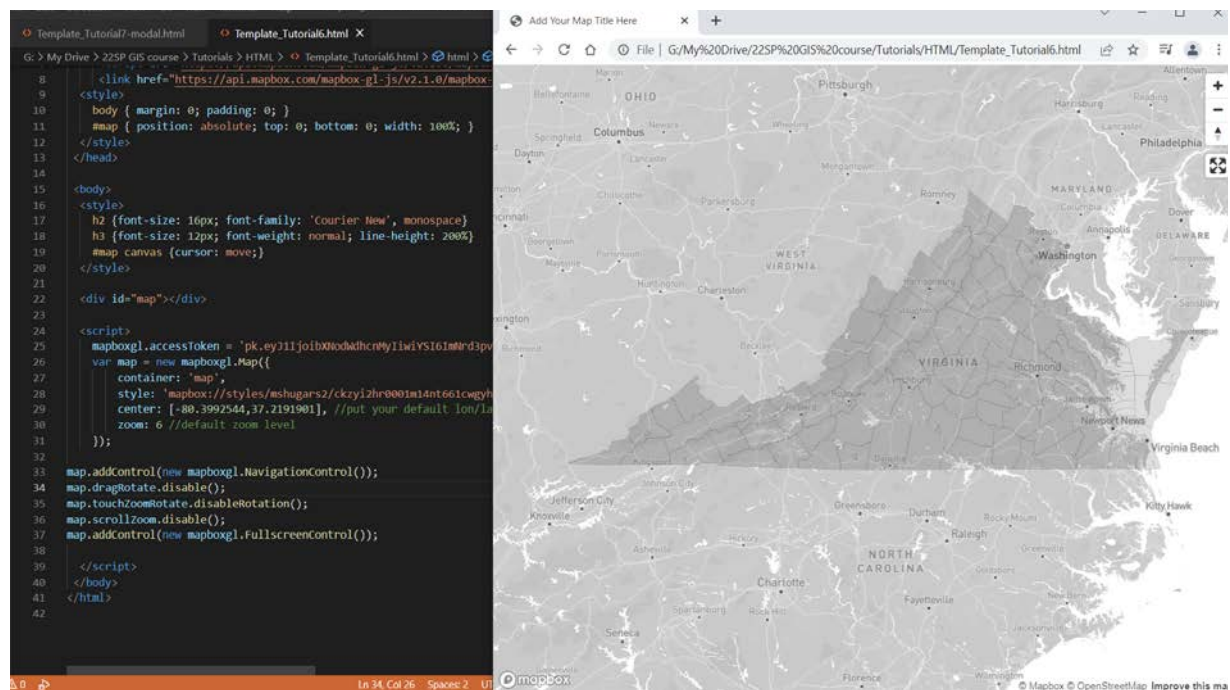
- **Studio:** where you create and edit basemaps
- **Tileset:** map image data that styles by zoom level; generally, basemap information
- **Dataset:** as opposed to tilesets, datasets contain a specific set of data (eg location of community centers, or industrial land use polygons)
- **Access token:** unique id number that you create for your account and use to activate your map in HTML code
- **Component:** groups of shapefile or information layers – generally tilesets – for instance, “land water & sky” or “road network”. These are tilesets made by Mapbox (you can’t upload or make your own component).
- **Layer:** specific subset of data, for instance “water” or “sky” or “bridges”; each component contains multiple layers. When you upload your own shapefile or database information, you can add them as layers to your map.

Visual you can style with Mapbox Studio:

- color and outline of **buildings, waterways, roads, natural features, and ground**
- content, size, font, and color of **labels**
- what content displays at which **zoom levels**
- map **cant** (view angle), and **fog**
- note: Mapbox does not show topography lines by default, but you can add the Tileset as a layer to your basemap

You can also **upload** custom **Datasets** or **Tilesets** to your Mapbox base map. This information can be point data, lines, or polygons – that is, traditional Shapefiles. It can also be csv spreadsheets with geographic information in it (latitude and longitude), or even Geo-located raster images (GeoTIFFs). In this Tutorial you’ll upload some useful census shapefile data from QGIS.

On **Mapbox** you create the map **object** with all of its relevant visual and informational layers. With **code**, you **interact** (and allow other people to interact) with that object. Once you embed the map object in a website, you will add any interactive features, such as on-hover information popups or clicks, with code.

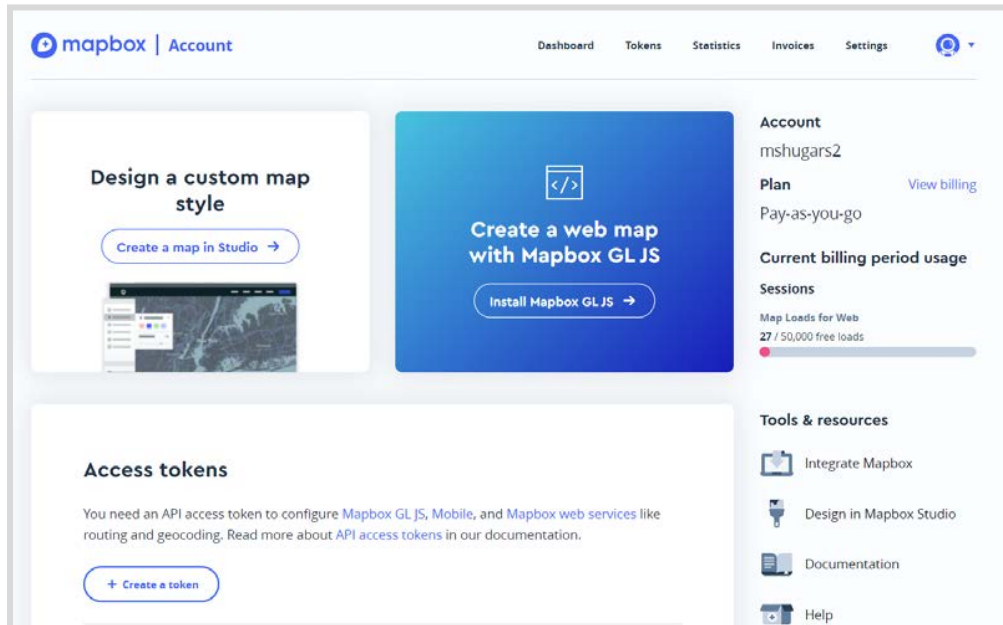


Step 1: Create a Mapbox account.

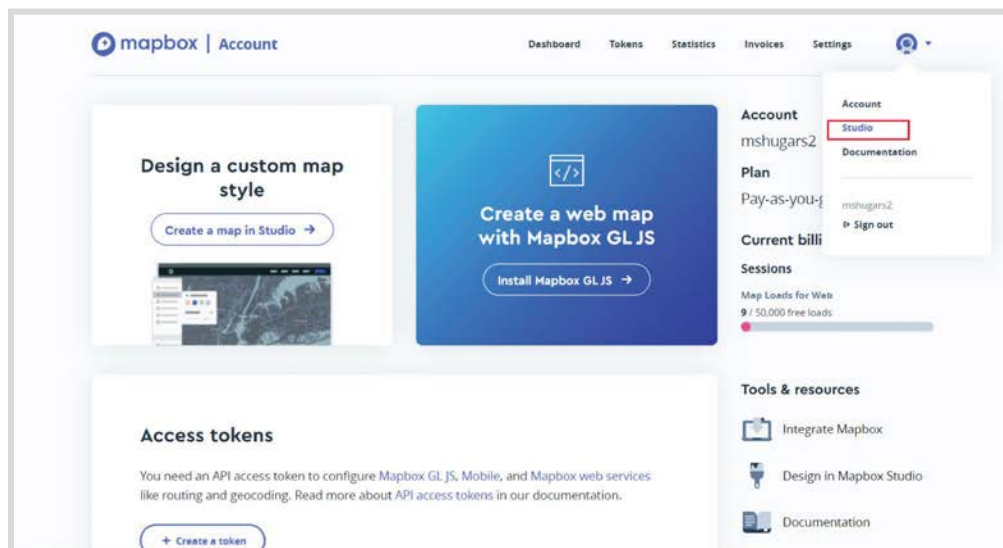
Mapbox is free for non-commercial use (ie if your map has fewer than 50,000 loads per month).

<https://account.mapbox.com/>

After confirming your account, you will see the Mapbox home page:

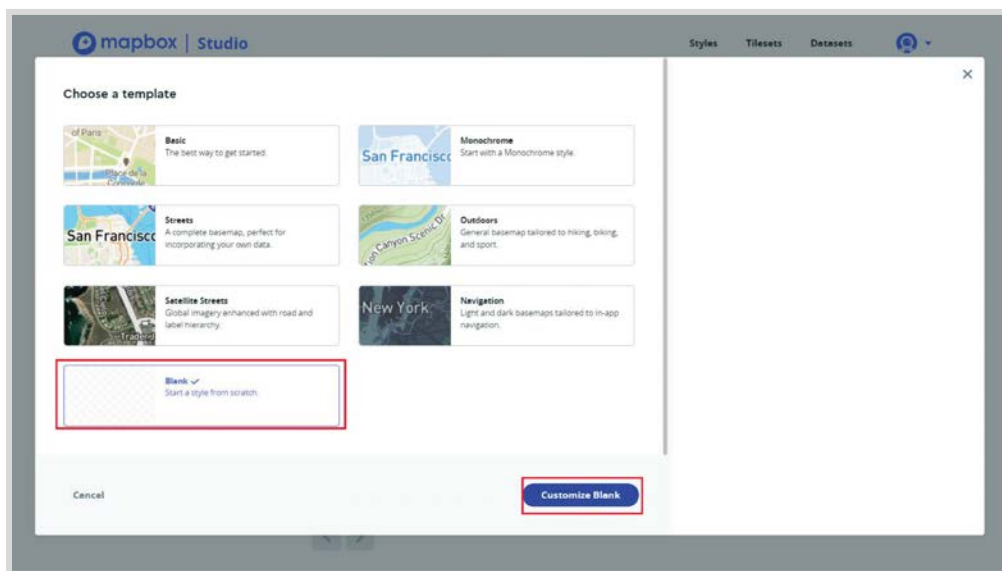
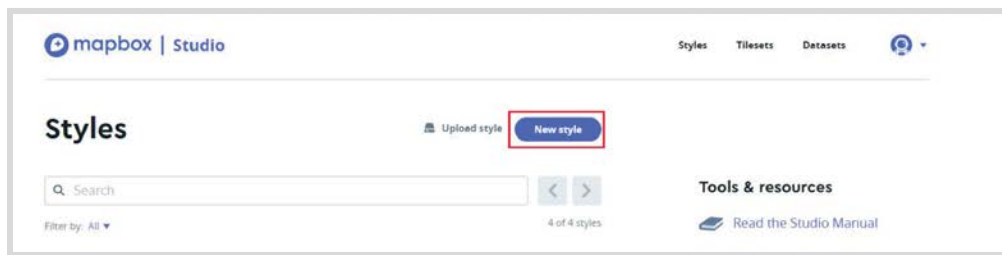


1b Navigate to the Mapbox Studio, where the visual magic happens, by clicking on your profile icon in the upper right and selecting “Studio”



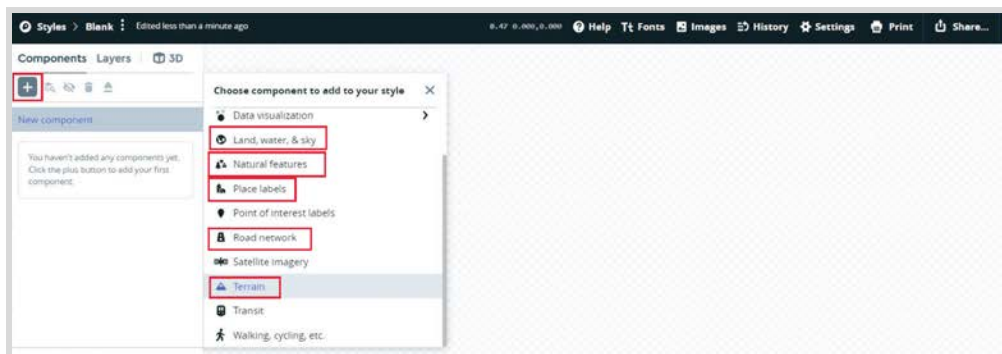
Step 2: In Mapbox Studio, create a new Map Style.

2a Click **"New Style"** and scroll down to **Blank** map style.

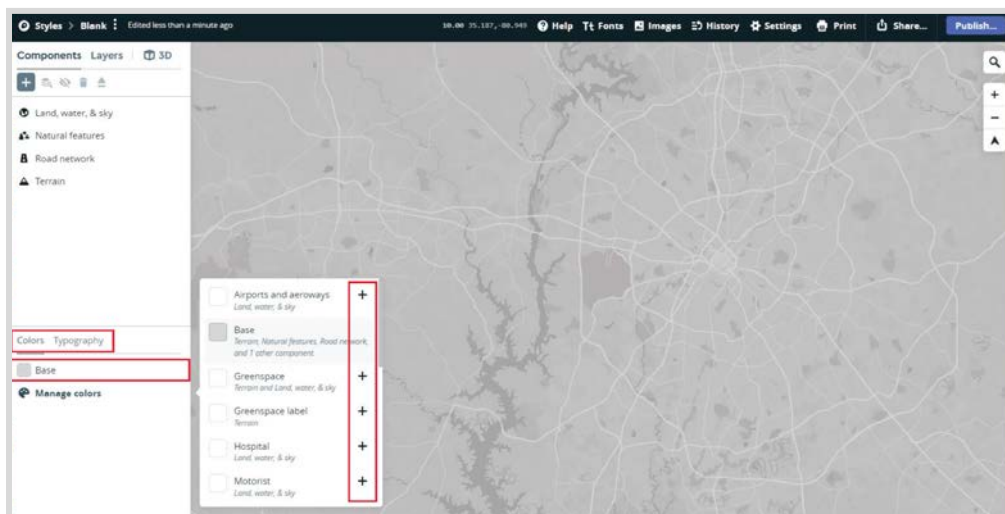
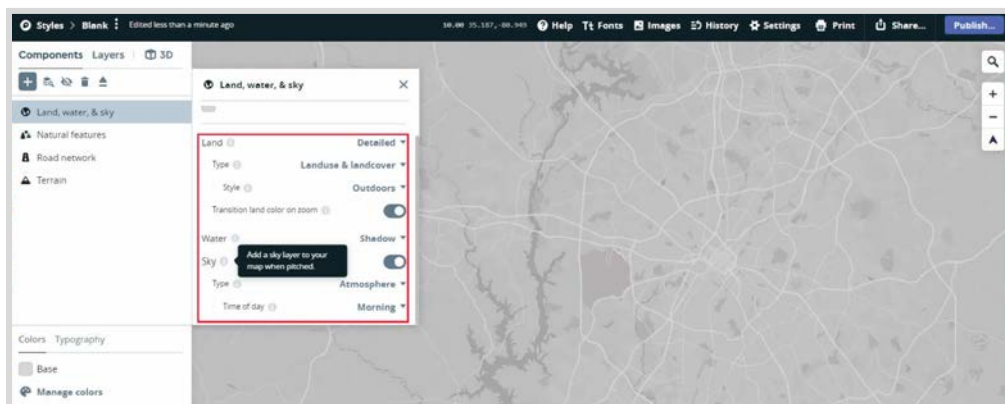
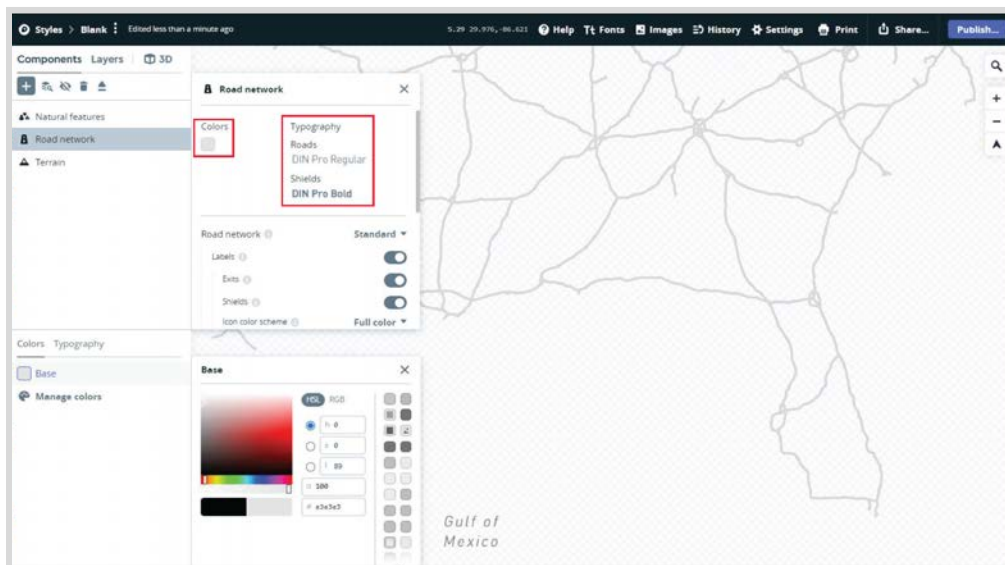


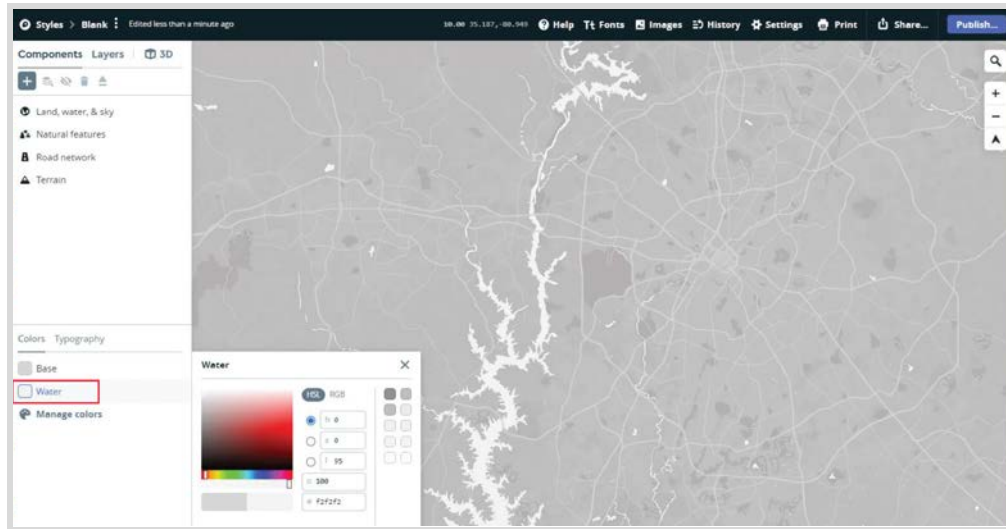
2b Now, add the following **Components**:

- Land, water, & sky
- Natural features
- Place labels
- Road network
- Terrain



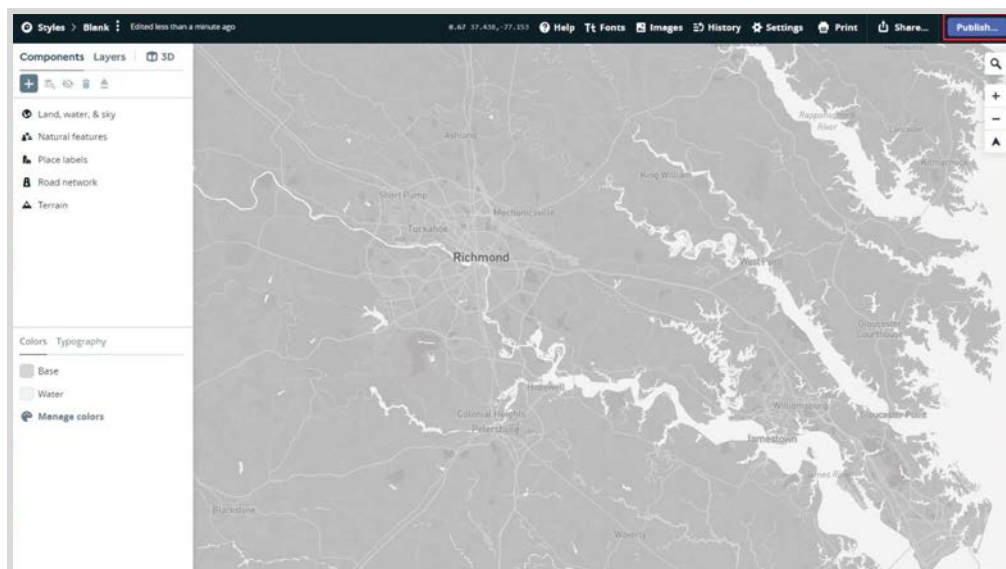
Play around with different styling options by clicking on these components and scrolling down to see the options. **Note: Label components are separated from Line and Polygon components.** This means that you will style the label (font, color, etc.) separately from the polygons that they refer to.





Mapbox's user interface has some redundancy baked in; that is, you can sometimes edit the same thing in two different places. As an example, you'll see **Typography** and **Color** tabs at the bottom of the Component editor area. Think of these as overall controls – you can control the base color of the entire map, or of certain general features (water, roads, etc.); likewise, you can change the fonts and colors of categories of labels. To change more granular information (eg. creeks or streams verses rivers in the “water” category), you will edit the specific component or its layers.

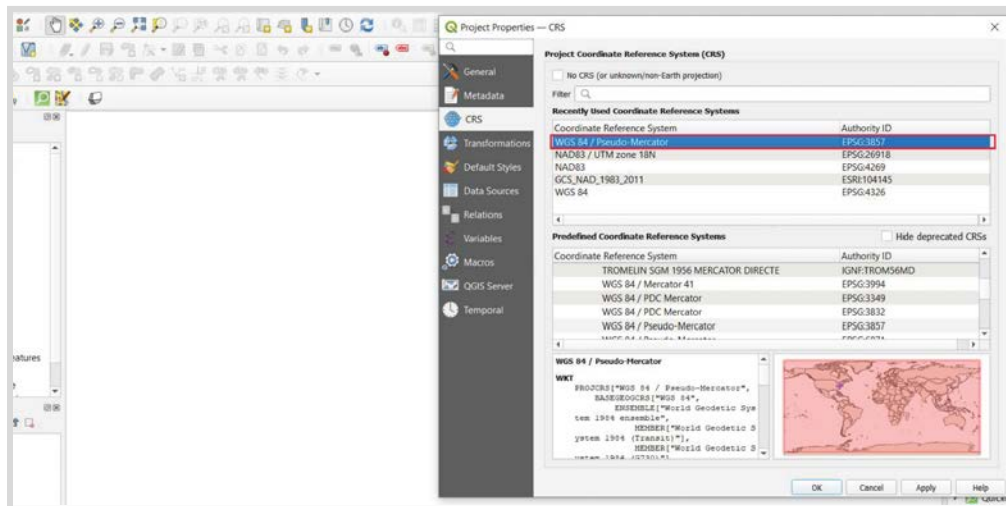
2c **Publish** your new Map style. This makes your changes live on the web, so you can see your updated map when you embed it. Note that when you make changes to your map, you will not be able to see these changes in your embedded map until you've Published them.



Step 3: Export TIGER county data from QGIS for your Appalachian city's state.

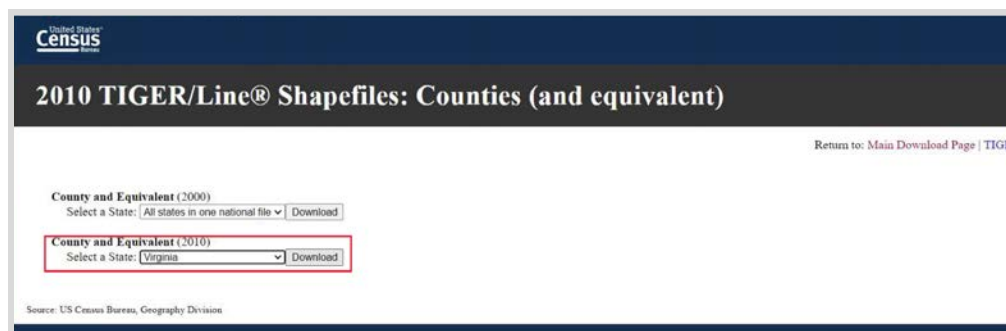
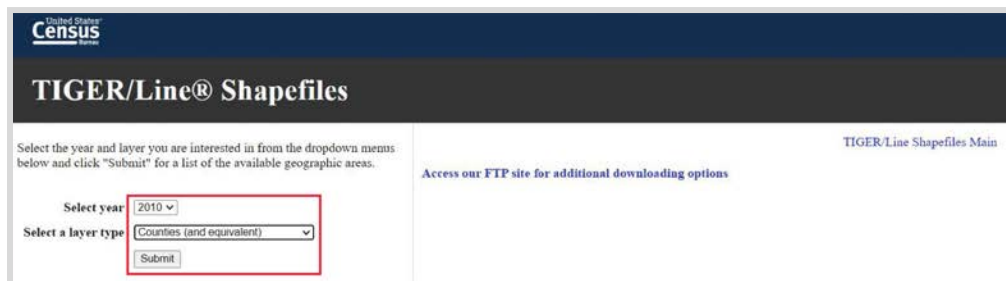
Note: this is a larger scale than we've been dealing with (we've been using census block groups). Because we're uploading data for the entire state, it will be easier to see patterns at the county level.

3a Create a new QGIS file. Change the CRS to **EPSG 3857: Pseudo Mercator**. This is Mapbox's (and Google Maps's) projection system.



3b Download the **2010 TIGER shapefile for counties in your city's state:**

<https://www.census.gov/geographies/mapping-files/time-series/geo/tiger-line-file.htm>



3c Download BOTH the **2013 and 2019** county ACS tables **B02001** (racial): <https://data.census.gov/cedsci/>

The screenshot shows the 'Advanced Search' interface on the Census Data Search website. The search term 'b02001' is entered in the search bar. The '1 Filter' section shows 'All Counties within Virginia' selected. The 'Find a Filter' section shows 'Geography' selected. The 'Select Table Vintages' section shows the selection of 'ACS 5-Year Estimates Detailed Tables' for the years 2013 and 2019. A red box highlights the '2019' and '2013' columns, and another red box highlights the 'ACS 5-Year Estimates Detailed Tables' row. A red arrow points to the 'File Type' section, which has 'CSV' selected. A red box with the text 'BE SURE TO SELECT 5-YEAR ESTIMATES' points to the 'ACS 5-Year Estimates Detailed Tables' row. The 'What You're Getting' section lists the files: 2 .csv files (metadata), 2 .csv files (data), and 2 .txt files (table title). The 'Compressed Size Estimate' is 10.2 MB.

NOTE: **UNZIP** your downloaded folders before doing anything else with them. Do not try to open or import the shapefile or census tables without unzipping them first. Be sure to save your unzipped files in your class folder.

3d Clean up your ACS data and find the correct GEOIDs to match the TIGER file. To simplify this, check out the GEOID attribute table of your shapefile verses your ACS GEOIDs, and you'll notice that the leading string is always the same: **0500000US**. You can simply **Find and Replace** this with nothing in Google Sheets to remove it from all your GEOIDs (which will then match the TIGER file).

NOTE: be sure to include the year in your ACS column names (eg **WhiteOnly2019** or **WhiteOnly2013**).

The screenshot shows a Google Sheet with ACS data. The columns are labeled A through L. The rows show data for B02001_001E and B02001_001M. A red box highlights the 'B02001_001M' column, and another red box highlights the 'B02001_001E' column. A red arrow points to the 'B02001_001M' column, and another red arrow points to the 'B02001_001E' column. A red box with the text 'DELETE' points to the 'B02001_001M' column.

The screenshot shows a Google Sheet with the cleanup of GEOIDs. The columns are labeled A through G. The rows show the 'Total2013' and 'WhiteOnly2013' columns. A red box highlights the 'GEO_ID' column, and another red box highlights the 'WhiteOnly2013' column. A red arrow points to the 'GEO_ID' column, and another red arrow points to the 'WhiteOnly2013' column. A red box with the text 'DELETE' points to the 'GEO_ID' column.

The screenshot shows two windows. The left window is QGIS, displaying a table with columns: STATEFP10, COUNTYFP10, COUNTYNS10, GEOID10, and NAME10. The right window is a Google Docs spreadsheet, displaying a table with columns: GEO_ID, WhiteOnly2013, and CBSAFP10.

STATEFP10	COUNTYFP10	COUNTYNS10	GEOID10	NAME10
1	51	001	01480091	51001
2	51	003	01675170	51003
3	51	005	01673675	51005
4	51	007	01497770	51007
5	51	009	01480095	51009
6	51	011	01497238	51011
7	51	013	01480097	51013
8	51	015	01480098	51015
9	51	017	01673638	51017
10	51	019	01674818	51019
11	51	021	01494613	51021
12	51	023	01674418	51023

GEO_ID	WhiteOnly2013	CBSAFP10
0500000US51001	22712	NULL
0500000US51003	82359	16820
0500000US51005	15174	NULL
0500000US51007	9171	40060
0500000US51009	24747	31340
0500000US51011	11578	31340
0500000US51013	153109	47900
0500000US51015	68628	44420
0500000US51017	4362	NULL
0500000US51019	63281	31340

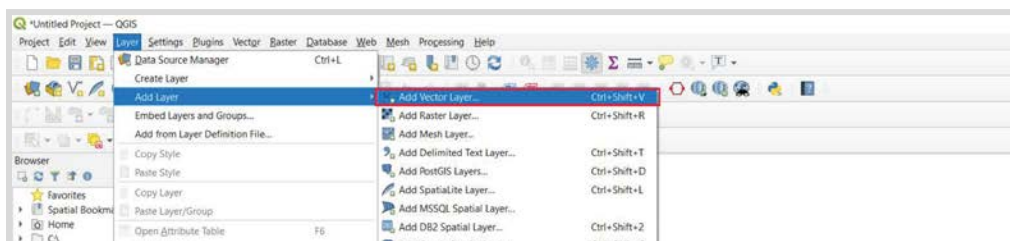
The screenshot shows a Google Docs spreadsheet with a 'Find and replace' dialog box open. The 'Find' field contains '0500000US' and the 'Replace with' field is empty. The 'Search' dropdown is set to 'All sheets'.

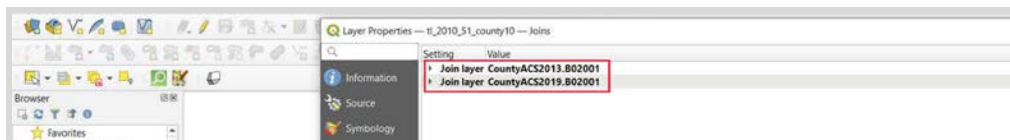
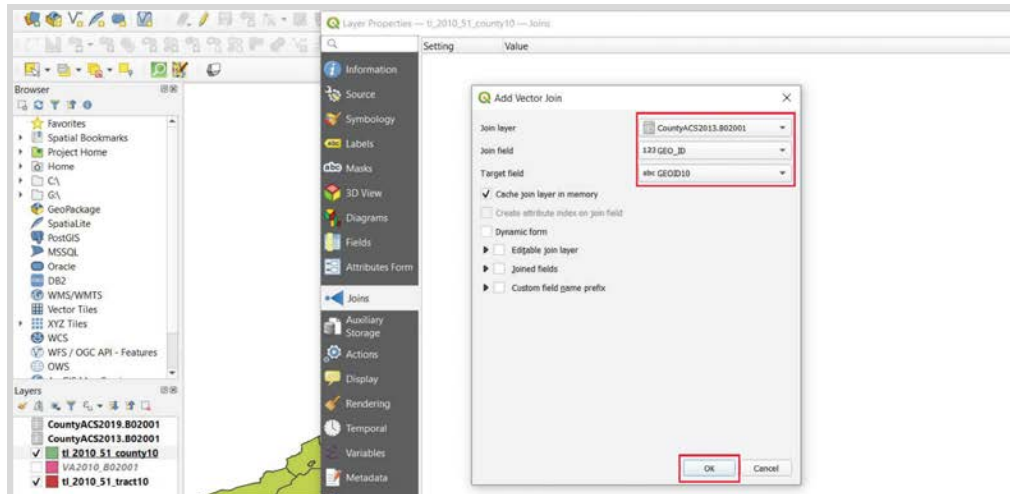
	A	B	C
1	Total2013	WhiteOnly2013	GEO_ID
2	33289	22712	0500000US51001
3	100636	82359	0500000US51003
4	16240	15174	0500000US51005
5	12712	9171	0500000US51007
6	32244	24747	0500000US51009
7	15054	11578	0500000US51011
8	214861	153109	0500000US51013
9	73726	68628	0500000US51015
10	4686	4362	0500000US51017
11	69175	63281	0500000US51019
12	6795	6442	0500000US51021
13	33076	31297	0500000US51023
14	17220	7149	0500000US51025
15	23920	23032	0500000US51027
16	17126	10686	0500000US51029
17	54967	45262	0500000US51031
18	28757	19109	0500000US51033
19	29979	29286	0500000US51035
20	7205	3027	0500000US51036
21	12478	8415	0500000US51037
22	320430	222794	0500000US51041
23	14191	12860	0500000US51043
24	5199	5041	0500000US51045

The screenshot shows a Google Docs spreadsheet with the 'GEO_ID' column highlighted. The data in the 'GEO_ID' column is as follows:

	A	B	C
1	Total2013	WhiteOnly2013	GEO_ID
2	33289	22712	51001
3	100636	82359	51003
4	16240	15174	51005
5	12712	9171	51007
6	32244	24747	51009
7	15054	11578	51011
8	214861	153109	51013
9	73726	68628	51015
10	4686	4362	51017
11	69175	63281	51019
12	6795	6442	51021
13	33076	31297	51023
14	17220	7149	51025
15	23920	23032	51027

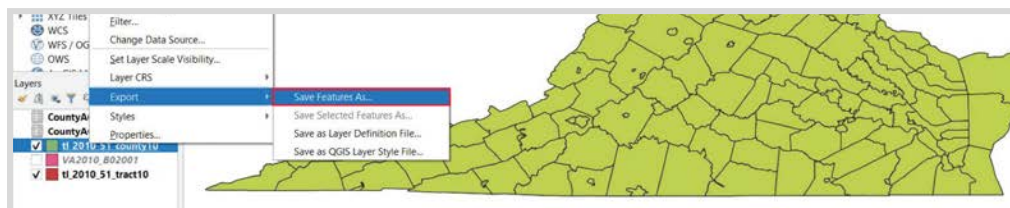
3e **Import** the **TIGER** file and **ACS .csvs** to QGIS, and join **BOTH** the 2013 and 2019 tables to your TIGER shapefile, and check that the join was successful in the Attribute Table.

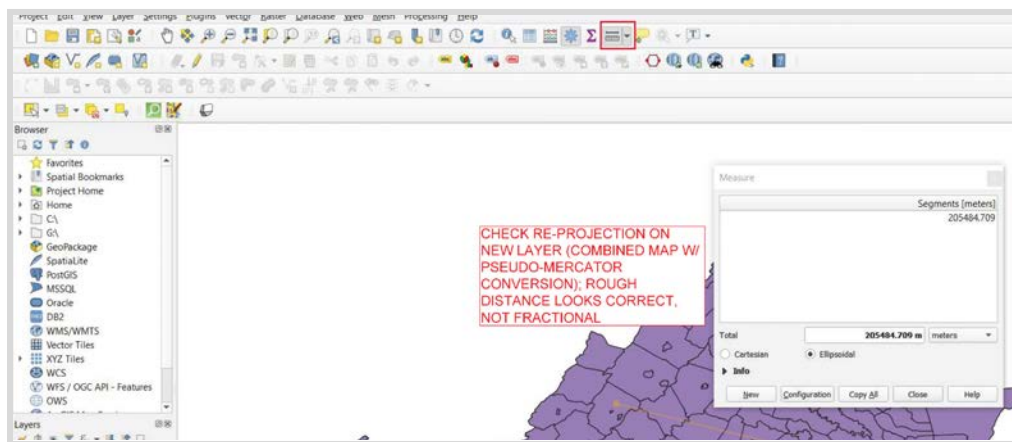
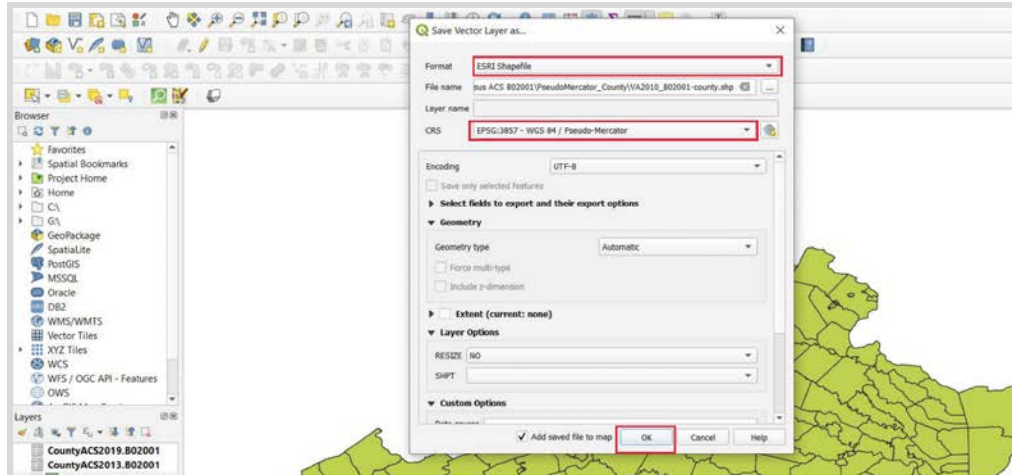




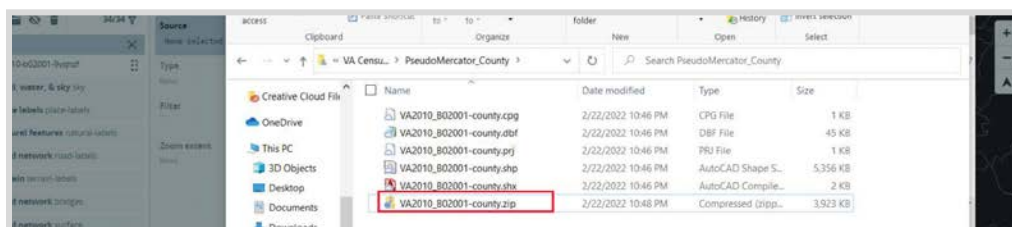
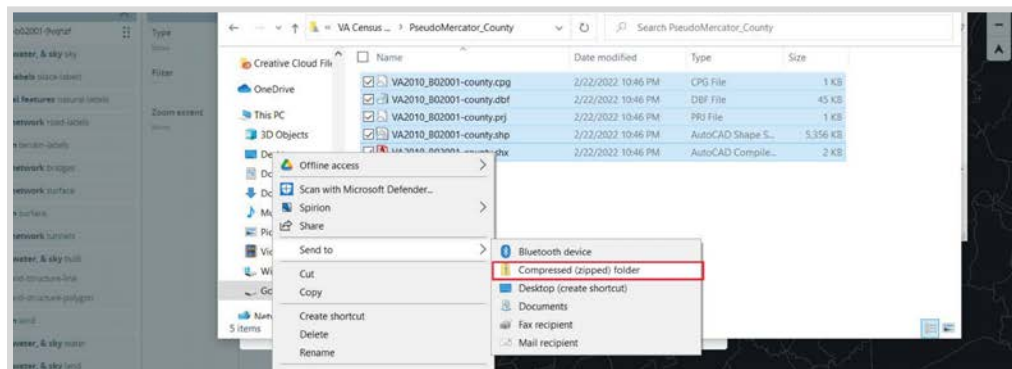
	MITCC10	CSAFP10	CBSAFP10	METDNP10	FUNCTSTAT10	ALAND10	AWATER10	INTPTLAT10	INTPTLON10	CS2013.B02001.T; 2013.B02001.Whtr.CS2019.B02001.T; 2019.B02001.Whtr
1	020	NULL	NULL	NULL	A	1164189289	2228794013	+37.7659435	-075.7578073	33289 22712 32673 22282
2	020	NULL	16820	NULL	A	1866599905	14035672	+38.0241840	-078.5535056	100636 82359 107405 87831
3	020	NULL	NULL	NULL	A	1153727336	8535515	+37.7879047	-080.0086689	16240 15174 15157 14012
4	020	NULL	40060	NULL	A	920142596	8424838	+37.3361313	-077.9732177	12712 9171 12953 9824
5	020	NULL	31340	NULL	A	1227404444	12730183	+37.6303621	-079.1478477	32244 24747 31775 24381
6	020	NULL	31340	NULL	A	863744566	3204517	+37.3707253	-078.8109404	15054 11578 15707 12263
7	020	548	47900	47894	A	67273573	244125	+38.8783374	-077.1007034	21484 153109 233464 166912
8	020	NULL	44420	NULL	A	2504515631	9987990	+38.1678073	-079.1466816	73726 68628 75079 69533
9	020	NULL	NULL	NULL	A	1370512659	14049862	+38.0689876	-079.7328980	4686 4362 4307 3933
10	020	NULL	31340	NULL	A	1950314965	42043879	+37.3124079	-079.5279466	69175 63281 78376 70090

3f Export the joined shapefile as an “ESRI Shapefile”, saved in a new folder so you can easily zip its various parts. Make sure to set your **export** CRS to “**EPSG 3857 PseudoMercator**”. This is Mapbox's projection system



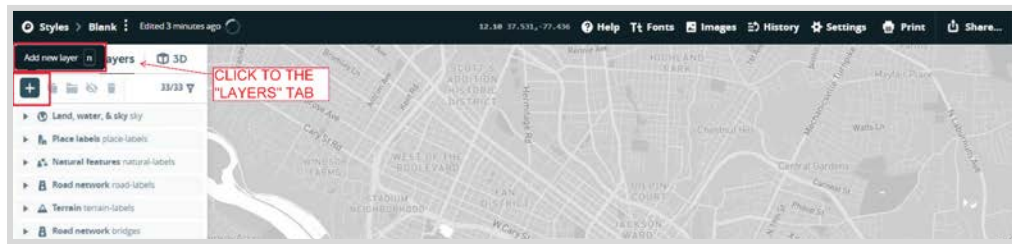


3g Zip your Shapefile files.

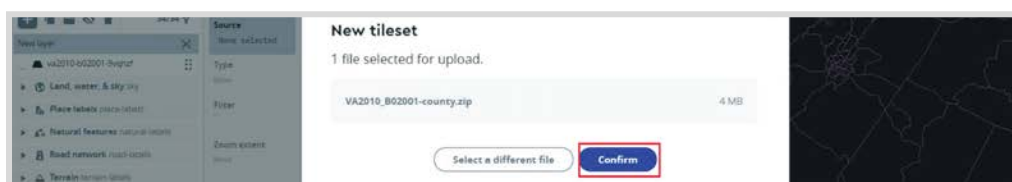
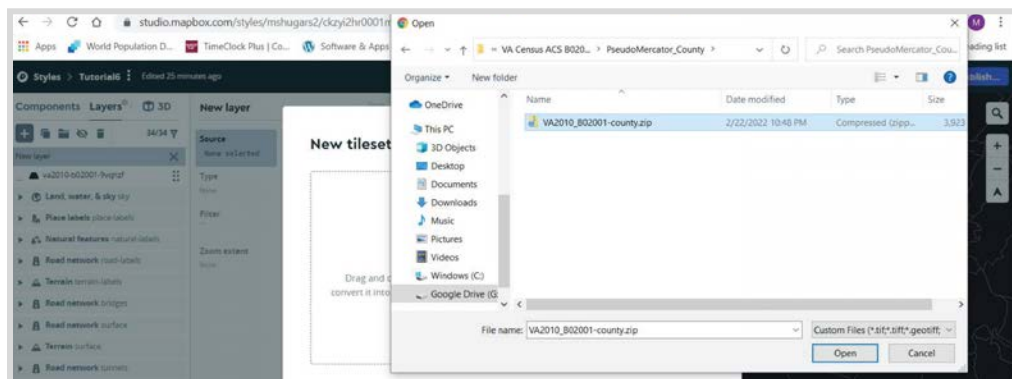
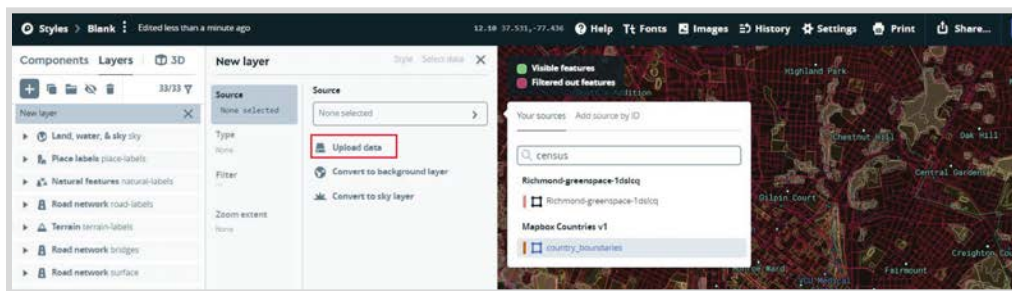


Step 4: Add your zipped Shapefile as a layer to your Mapbox base map.

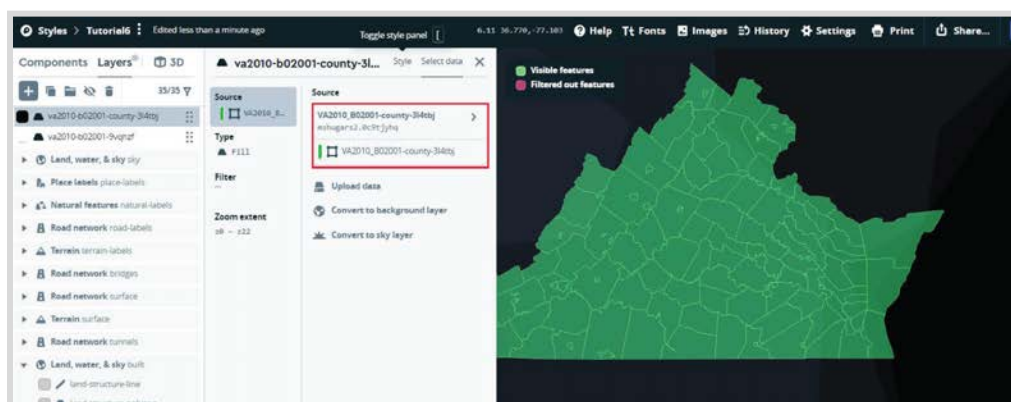
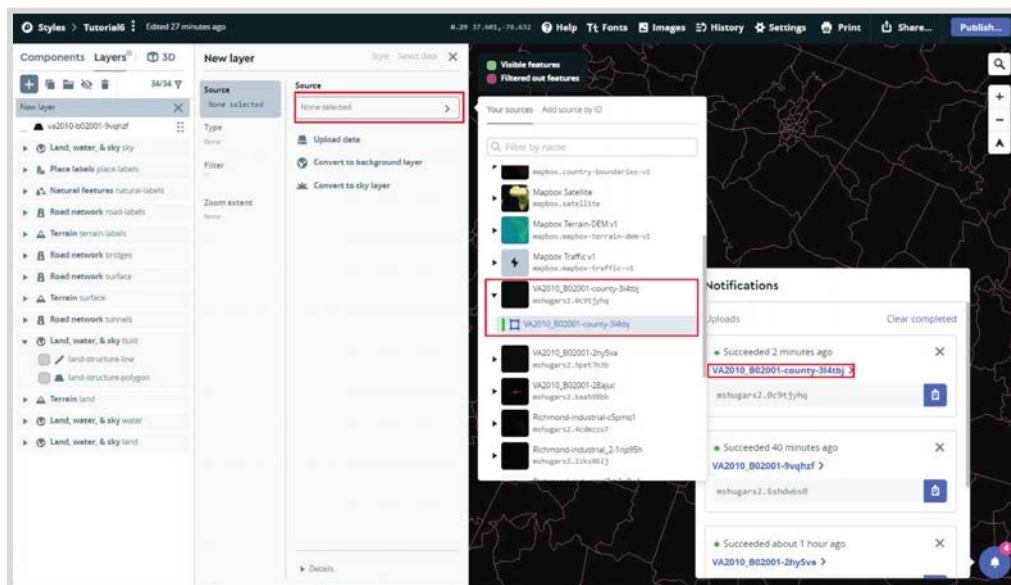
4a Under the **Layers** tab in your Studio view, click the **plus** sign “Add New Layer”



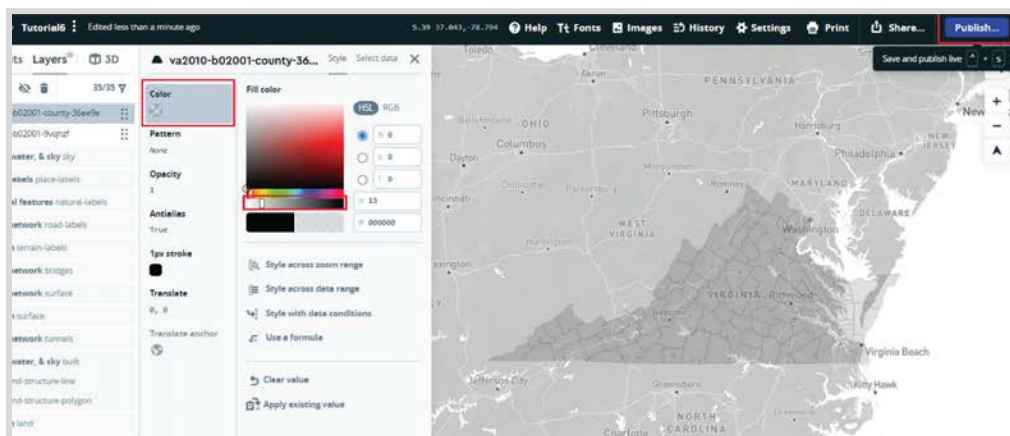
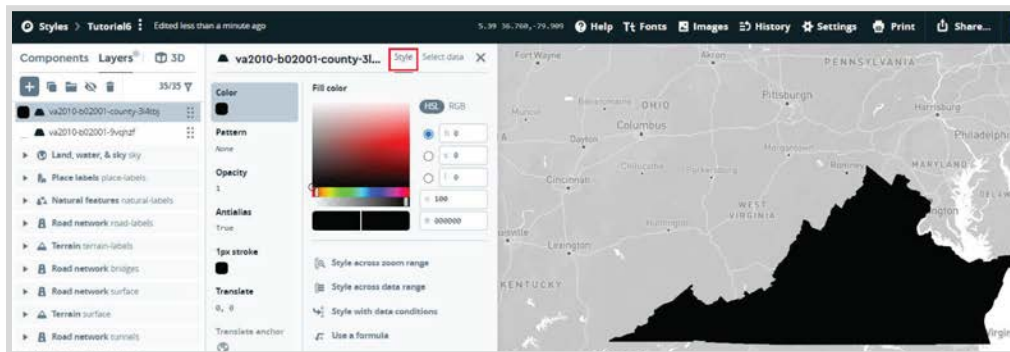
4b In Source, select “Upload Data” and select your zipped shapefile.



4c Once the upload has processed (check the lower right of your screen), scroll down to your newly uploaded tileset (**check the notification list for its name**), **click** on it, and select the **square shapefile symbol sublayer**. You should see the outline of your state show up in bright green.



4d Click the **“Style”** tab, and for now simply slide the fill color opacity to around 20%. In the next Tutorial, we’ll do fancier styling on this layer. For now, you want to see the ghostly shape of the state with its counties outlined in black.

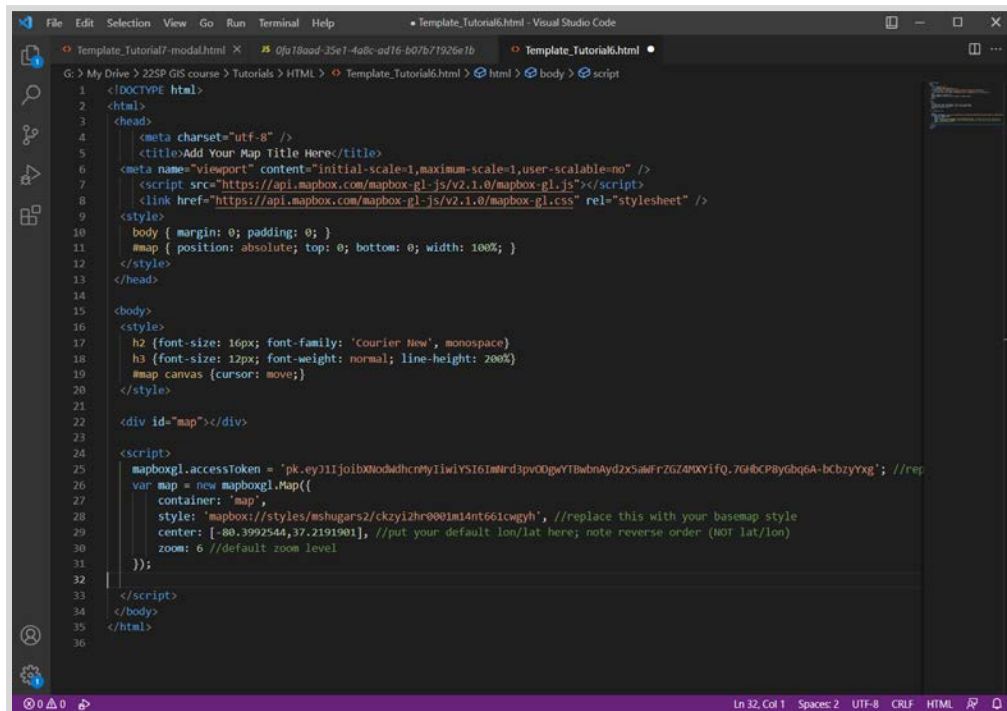


4e **Publish** your basemap changes (upper right button).

Step 5: Download and install Visual Studio Code (this will be our code editor):

<https://code.visualstudio.com/download>

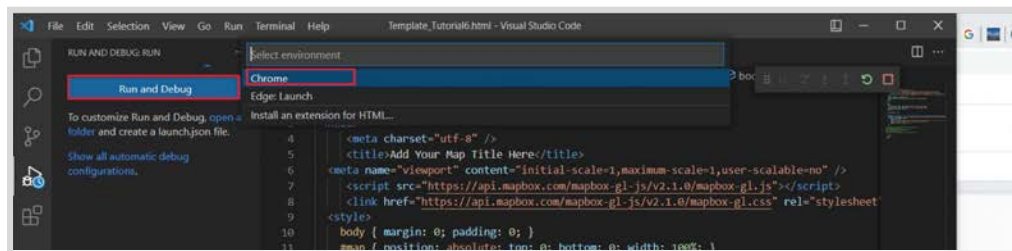
5a Download the **HTML template** (Canvas > Files > Templates > “Template-Tutorial6-7”) for this tutorial and open it in Visual Studio:

A screenshot of the Visual Studio Code editor interface. The main editor window displays an HTML file named 'Template_Tutorial6.html'. The code is a template for a web page that includes a map. It features a head section with a meta charset, a title placeholder, a viewport meta tag, and links to Mapbox GL JS and CSS. The body section contains CSS for a h2 header, a h3 header, and a map canvas. A script section initializes a Mapbox GL map with a specific access token, container, style, center coordinates, and zoom level. The status bar at the bottom indicates the current line and column (Ln 32, Col 1) and the file encoding (UTF-8).

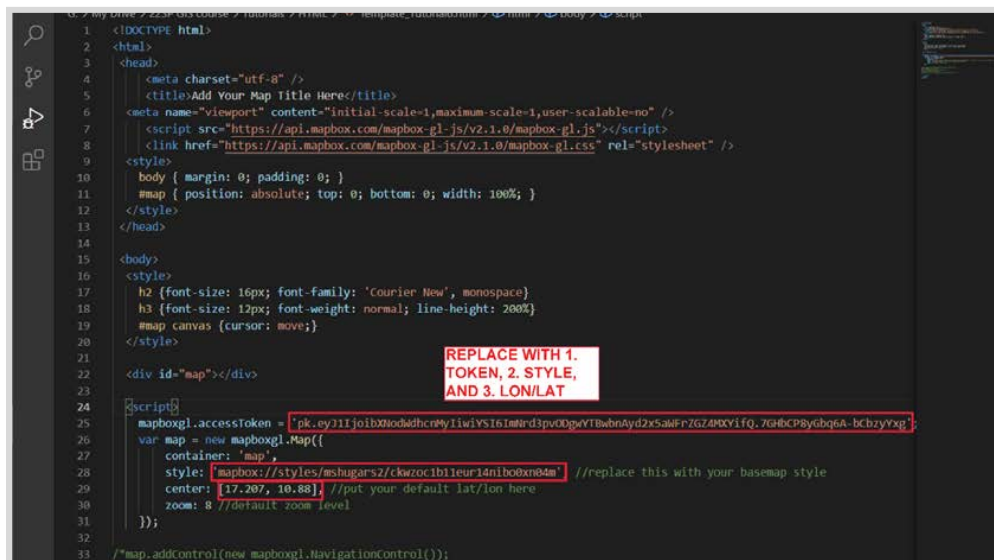
```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="utf-8" />
5   <title>Add Your Map Title Here</title>
6   <meta name="viewport" content="initial-scale=1,maximum-scale=1,user-scalable=no" />
7   <script src="https://api.mapbox.com/mapbox-gl-js/v2.1.0/mapbox-gl.js"></script>
8   <link href="https://api.mapbox.com/mapbox-gl-js/v2.1.0/mapbox-gl.css" rel="stylesheet" />
9 </head>
10 <body>
11   <div id="map"></div>
12 </body>
13 </html>
14
15 <style>
16   h2 {font-size: 16px; font-family: 'Courier New', monospace}
17   h3 {font-size: 12px; font-weight: normal; line-height: 200%}
18   #map canvas {cursor: move;}
19 </style>
20
21 <script>
22   mapboxgl.accessToken = 'pk.eyJ1IjoibXNodWdhcnMyIiwiaW51IHRnd3pvODgwYTUwbnAydzSawFrZGZ4ODYifQ.7Gh8CP8yC8q6A-bC8zyYng'; //replace this with your mapboxgl access token
23   var map = new mapboxgl.Map({
24     container: 'map',
25     style: 'mapbox://styles/mahugars2/ckzy12hr0001m4nt661cwgh', //replace this with your basemap style
26     center: [-80.3992544, 37.2191901], //put your default lon/lat here; note reverse order (NOT lat/lon)
27     zoom: 6 //default zoom level
28   });
29 </script>
30 </body>
31 </html>
```

Step 6: Update HTML code with your map.

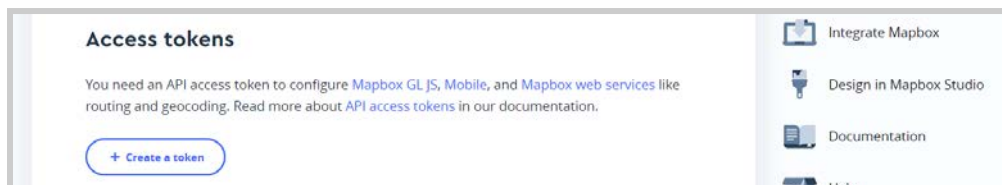
6a First, run your HTML in Chrome to see what it looks like by default. It might not show up until you've finished this step. Reload this page frequently to check the changes you've made:



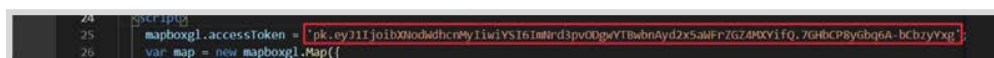
Next, you're going to update the **ACCESS TOKEN**, **MAP STYLE URL**, **CENTER**, and **ZOOM** in the HTML. Scroll down to this section (below the `<script>` tag):



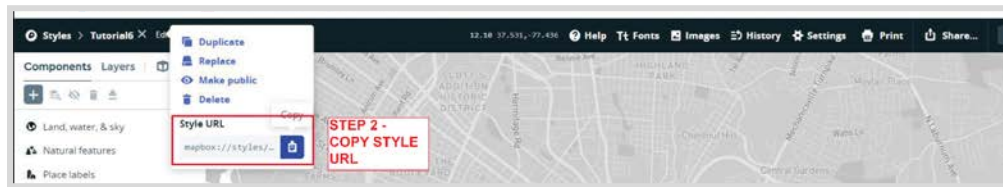
6b First, you need to create a Mapbox Access token from your Mapbox home page (<https://account.mapbox.com/>). Click **Create a token**. Name it anything and confirm.



Now, **copy** the new Token number (also known as an API key!) to the HTML template, replacing the 'pk...' string **INSIDE the quotation marks** after `mapboxgl.accessToken`:



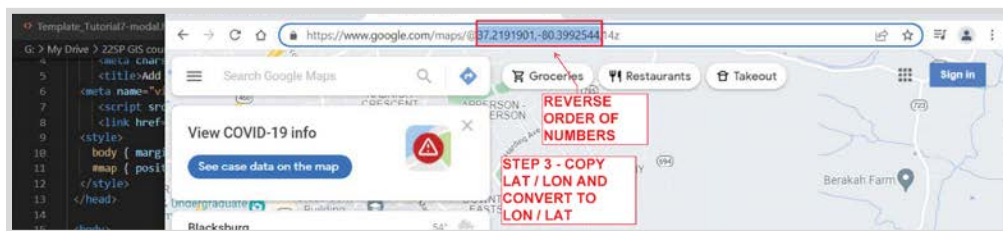
6c Second, add your **Map Style URL** to the HTML template. You can find the map style (eg basemap) url back in your Mapbox Studio. There are several places to find the url, but one is in the Style editor as shown below:



Replace the text **INSIDE** the quotation marks after *style* in the template:

```
27   container: 'map',  
28   style: "mapbox://styles/~" //replace this with your basemap style
```

6d Third, update the **Zoom level and Center** (Longitude / latitude starting point) in the template. Note that the template reads the center as Longitude / latitude, and NOT Latitude / longitude as it's shown in Google Maps. You need to **reverse** the numbers from Google Maps.



Replace the numbers after *center* in the template. Replace the number after *zoom* with an appropriate zoom level (try a few out to see what looks right by saving and reloading the map in Chrome).

```
29   center: [17.207, 10.88] //put your default lat/lon here  
30   zoom: 8 //default zoom level
```

6e **Check** that the code still works and looks right by **saving the file and reloading the map in Chrome**.

Step 7: Some final map edits

```
G: > My Drive > 22SP GIS course > Tutorials > HTML > Template_Tutorial6.html > html > body >
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="utf-8" />
5   <title>CHANGE MAP TITLE HERE
6   <meta name="viewport" content="initial-scale=1,maximum-scale=1,user-scalable=no" />
7   <script src="https://api.mapbox.com/mapbox-gl-js/v2.1.0/mapbox-gl.js"></script>
8   <link href="https://api.mapbox.com/mapbox-gl-js/v2.1.0/mapbox-gl.css" rel="stylesheet" />
9 <style>
10   body { margin: 0; padding: 0; }
11   #map { position: absolute; top: 0; bottom: 0; width: 100%; }
12   ADD CURSOR TYPE HERE
13 </style>
14 </head>
15
16 <body>
17   <div id="map"></div>
18
19   <script>
20     mapboxgl.accessToken = 'pk.eyJ1IjoibXVodWdhcnMyIiwiaSI6ImNrd3pvODgwYTbWbnAyd2x5aWFrZGZ4MXYif
21     var map = new mapboxgl.Map({
22       container: 'map',
23       style: 'mapbox://styles/mshugars2/ckzyixxxxxxxx', //replace this with your basemap styl
24       center: [-80.3992544,91], //put dyour default lon/lat here; note reverse order (NOT lat/
25       zoom: 2 //default zoom level
26     });
27     ADD MAP CONTROLS AND LIMITS HERE
28
29   </script>
30 </body>
</html>
```

7a Add some **map controls** for ease of use. You'll add Zoom tools (+ and -), a fullscreen button, disable tile control (so the map stays flat), and disable scroll zoom (so the viewer can only zoom in and out with the buttons).

Add the following text after the close of your `var map` block, as shown above.

```
map.addControl(new mapboxgl.NavigationControl());
map.dragRotate.disable();
map.touchZoomRotate.disableRotation();
map.scrollZoom.disable();
map.addControl(new mapboxgl.FullscreenControl());
```

7b Now, to make it clear that you can pan around the map by clicking and dragging, change the **cursor type** to "move" (the crossed arrows) by adding the following in your `<style>` block:

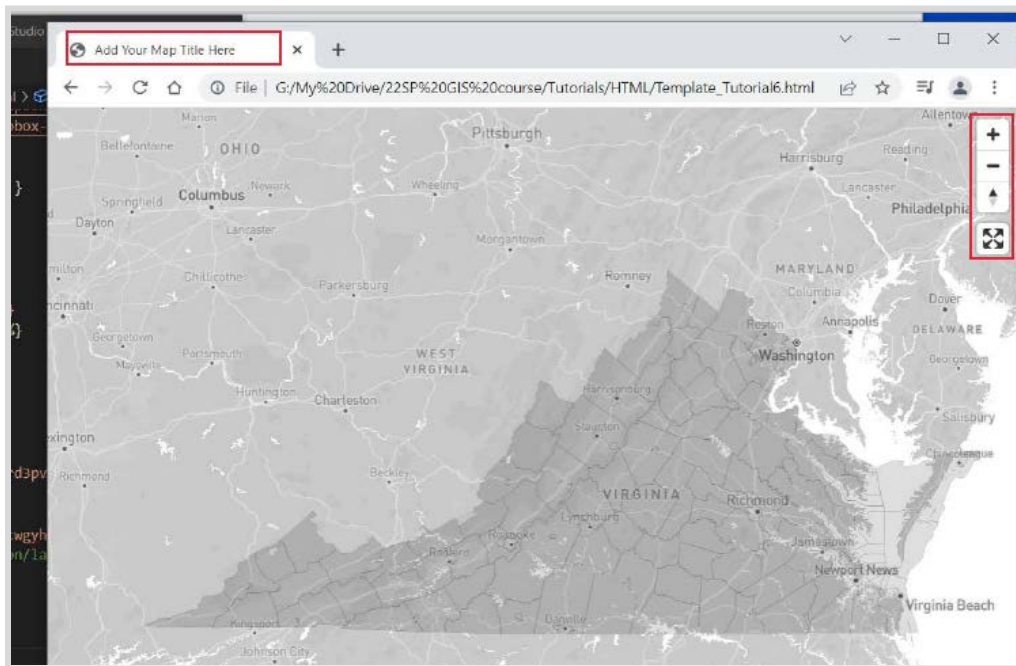
```
#map canvas {cursor: move;}
```

```
9 <style>
10   body { margin: 0; padding: 0; }
11   #map { position: absolute; top: 0; bottom: 0; width: 100%; }
12   ADD CURSOR TYPE HERE
13 </style>
```

7c Lastly, change your **map title**:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="utf-8" />
5   <title>CHANGE MAP TITLE HERE
```

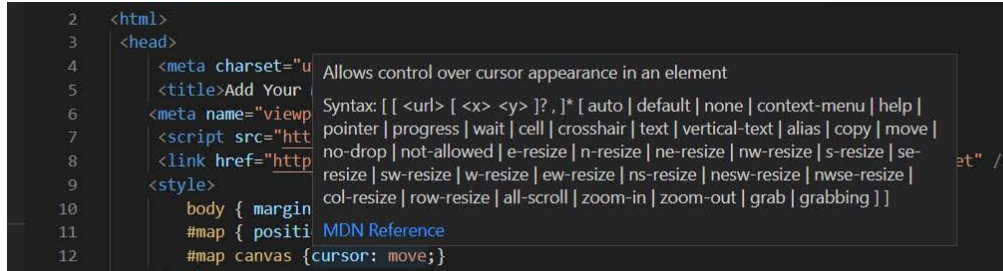
7d Now check out the changes in your Chrome map. Pan and zoom around to try out the new controls.



– Bonus –

Step 8: Try out some different cursor types and map control options.

8a Hover over the cursor section of your code in Visual Studio to see some different options:



8b Visual Studio won't be able to show you the map control options, since these are specific to Mapbox's language. However, you can see the possibilities in Mapbox's documentation page:

<https://docs.mapbox.com/mapbox-gl-js/api/markers/#navigationcontrol>

and

<https://docs.mapbox.com/mapbox-gl-js/api/handlers/>