

TUTORIAL 7 | JUMP INTO MAPBOX

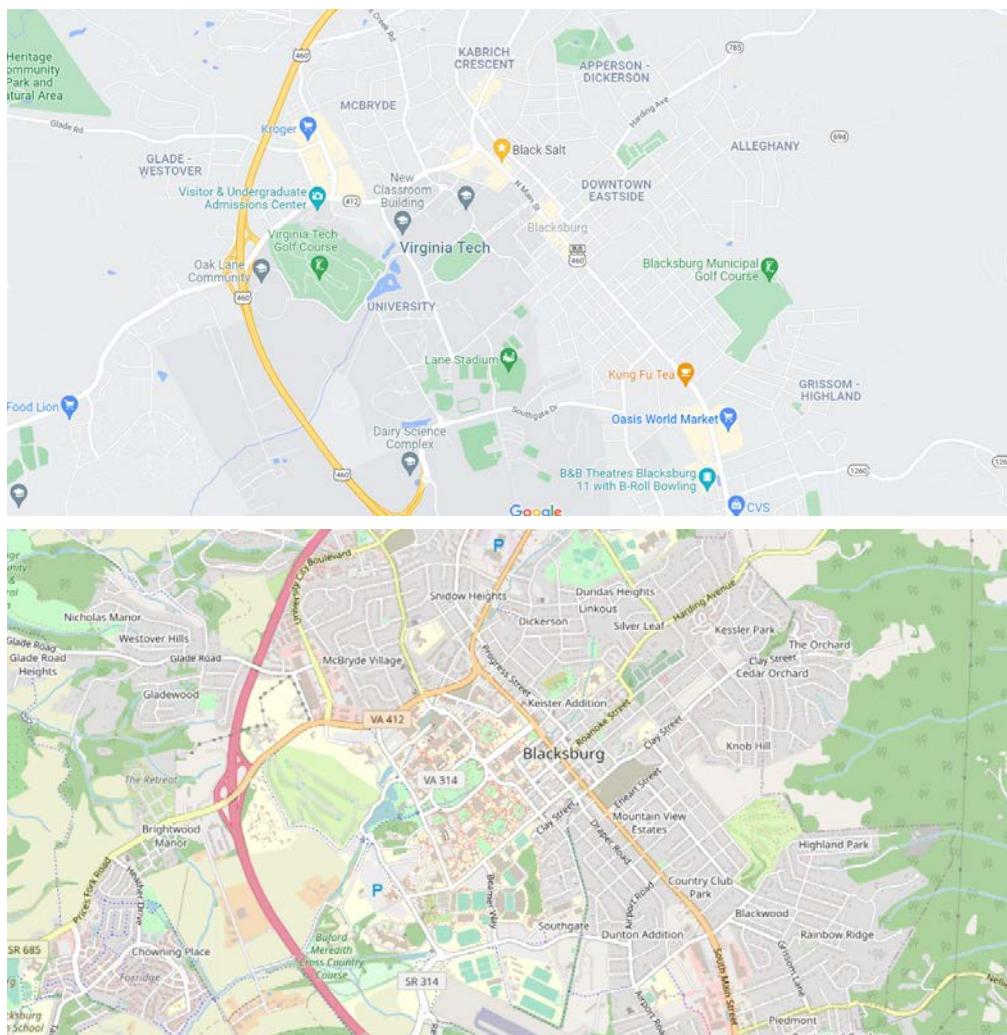
Goals

- 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

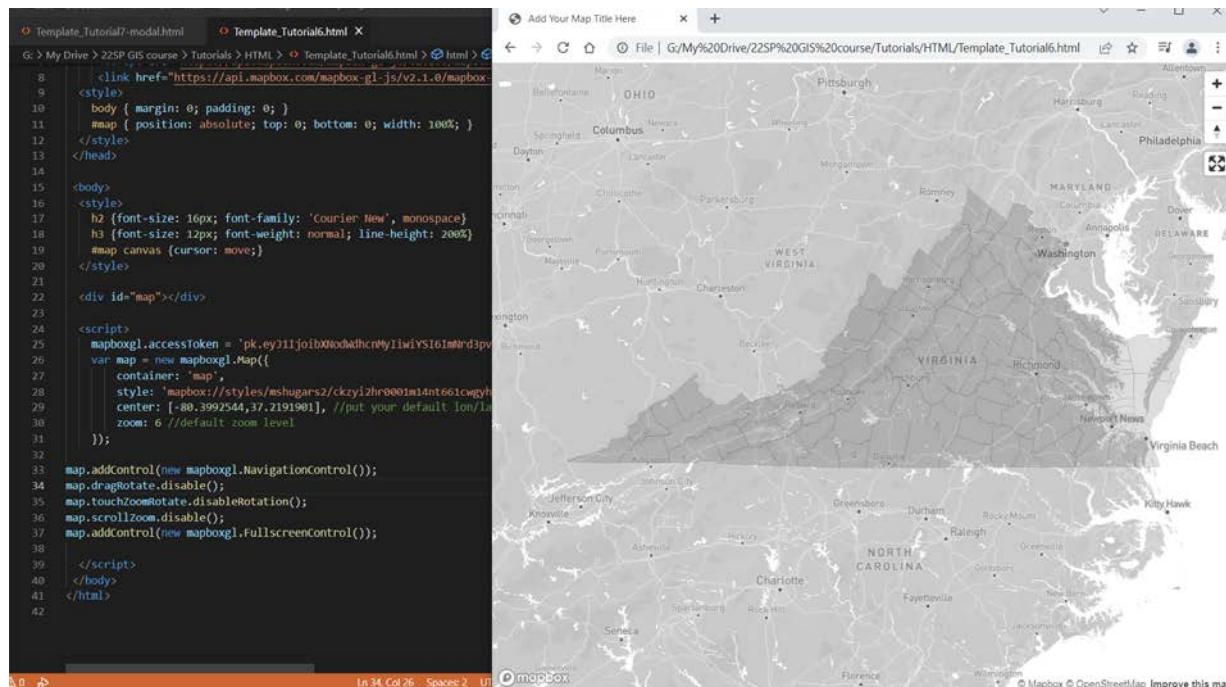
- **Studio:** where you create and edit basemaps
- **Tileset:** map image data that styles by zoom level; generally, basemap information
- **Dataset:** as opposed to tilesets, datasets contain a specific set of data (eg location of community centers, or industrial land use polygons)
- **Access token:** 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.



```
G: > My Drive > 22SP GIS course > Tutorials > HTML > < Template_Tutorial6.html > html >
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  </style>
13
14
15  <body>
16    <style>
17      h2 {font-size: 16px; font-family: 'Courier New', monospace}
18      h3 {font-size: 12px; font-weight: normal; line-height: 200%}
19      #map canvas {cursor: move;}
20    </style>
21
22    <div id="map"></div>
23
24    <script>
25      mapboxgl.accessToken = 'pk.eyJ1IjoiYXodWdnchmMyIiwiYSI6ImRd3py';
26      var map = new mapboxgl.Map({
27        container: 'map',
28        style: 'mapbox://styles/mshugarsz/czry12hr0001ml4nt661cwyh',
29        center: [-88.3992544, 37.2191901], //put your default lon/lat
30        zoom: 6 //default zoom level
31      });
32
33      map.addControl(new mapboxgl.NavigationControl());
34      map.dragRotate.disable();
35      map.touchZoomRotate.disableRotation();
36      map.scrollZoom.disable();
37      map.addControl(new mapboxgl.FullscreenControl());
38
39    </script>
40  </body>
41 </html>
```

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:

The screenshot shows the Mapbox Account dashboard. At the top, there are links for Dashboard, Tokens, Statistics, Invoices, and Settings. On the left, there are two main sections: "Design a custom map style" with a "Create a map in Studio" button and a thumbnail image, and "Create a web map with Mapbox GL JS" with an "Install Mapbox GL JS" button. Below these is a section for "Access tokens" with a "+ Create a token" button. On the right side, there's a sidebar titled "Tools & resources" containing links for Integrate Mapbox, Design in Mapbox Studio, Documentation, and Help. The sidebar also displays account information: mshugars2, Pay-as-you-go plan, and current billing period usage (27 / 50,000 free loads).

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

This screenshot is similar to the previous one, showing the Mapbox Account dashboard. However, a dropdown menu has been opened from the user profile icon in the top right corner. The "Studio" option is highlighted with a red box. Other options in the dropdown include "Documentation" and "Sign out". The rest of the interface remains the same, with the "Design a custom map style" and "Create a web map with Mapbox GL JS" sections visible on the left, and the "Tools & resources" sidebar on the right.

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

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

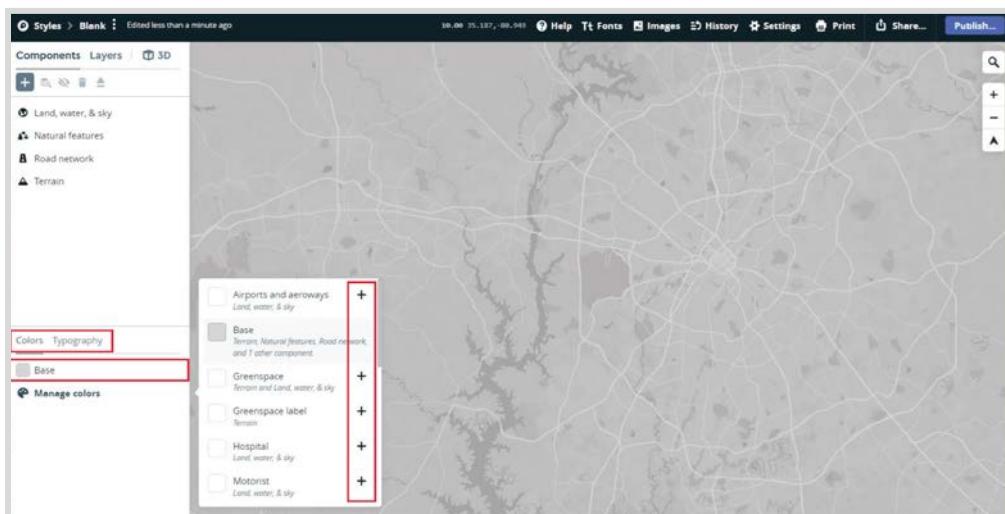
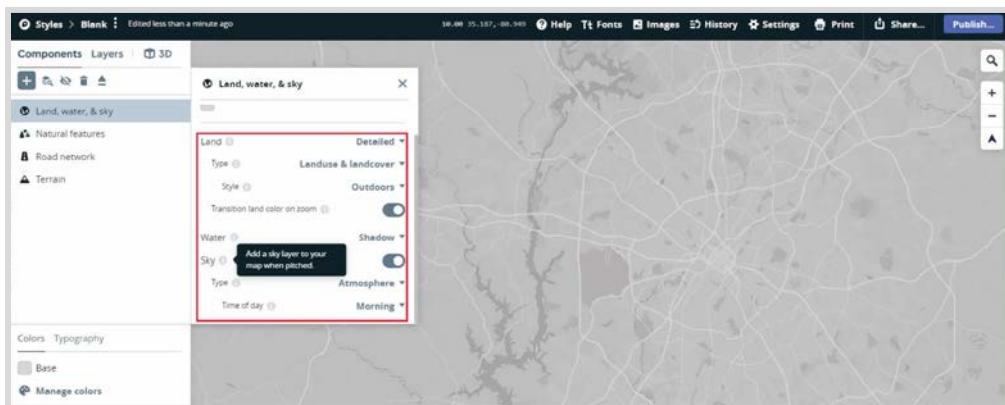
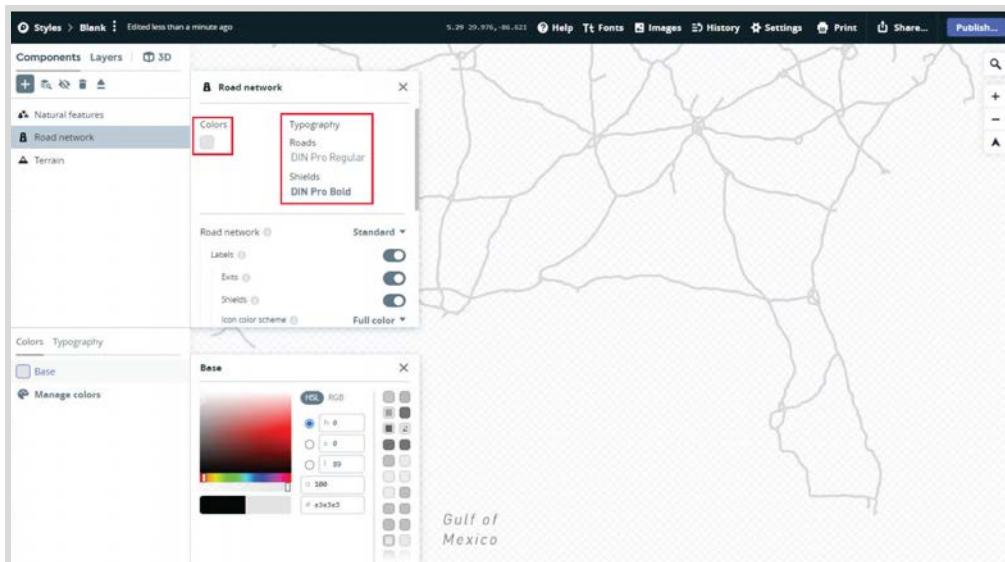
The top image shows the Mapbox Studio interface with the 'Styles' tab selected. A blue button labeled 'New style' is highlighted with a red box. The bottom image shows a modal dialog titled 'Choose a template'. It lists several pre-made styles: 'Basic' (a map of Paris), 'Monochrome' (a map of San Francisco), 'Streets' (a map of San Francisco), 'Outdoors' (a map of a canyon), 'Satellite Streets' (a map of New York), 'Navigation' (a map of New York), and 'Blank' (a blank white space). The 'Blank' option is highlighted with a red box. A blue button labeled 'Customize Blank' is at the bottom right of the dialog.

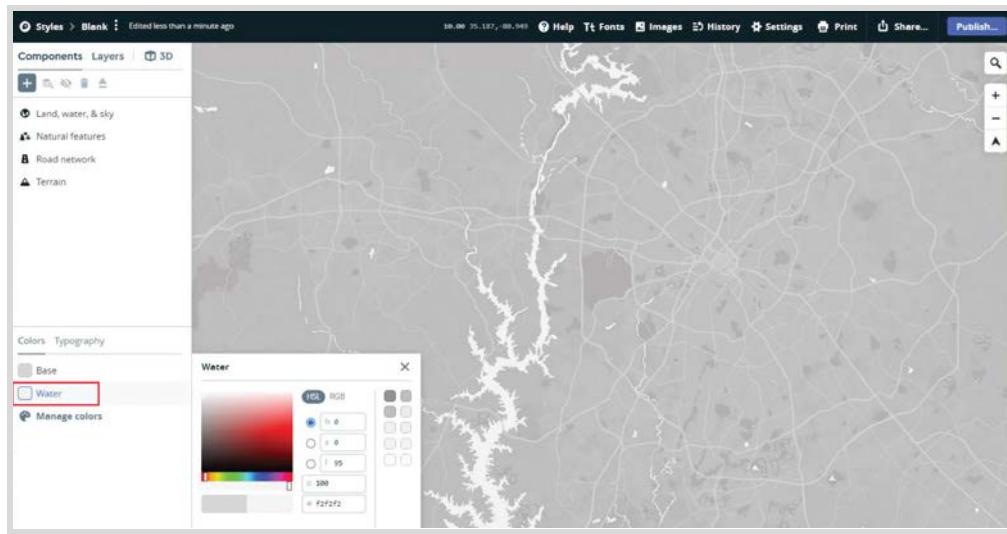
2b Now, add the following **Components**:

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

The screenshot shows the 'Components' panel in Mapbox Studio. A red box highlights the '+' button. A modal dialog titled 'Choose component to add to your style' is open, listing components: 'Data visualization', 'Land, water, & sky' (highlighted with a red box), 'Natural features' (highlighted with a red box), 'Place labels', 'Point of interest labels', 'Road network' (highlighted with a red box), 'Satellite imagery', 'Terrain' (highlighted with a red box), 'Transit', and 'Walking, cycling, etc.'

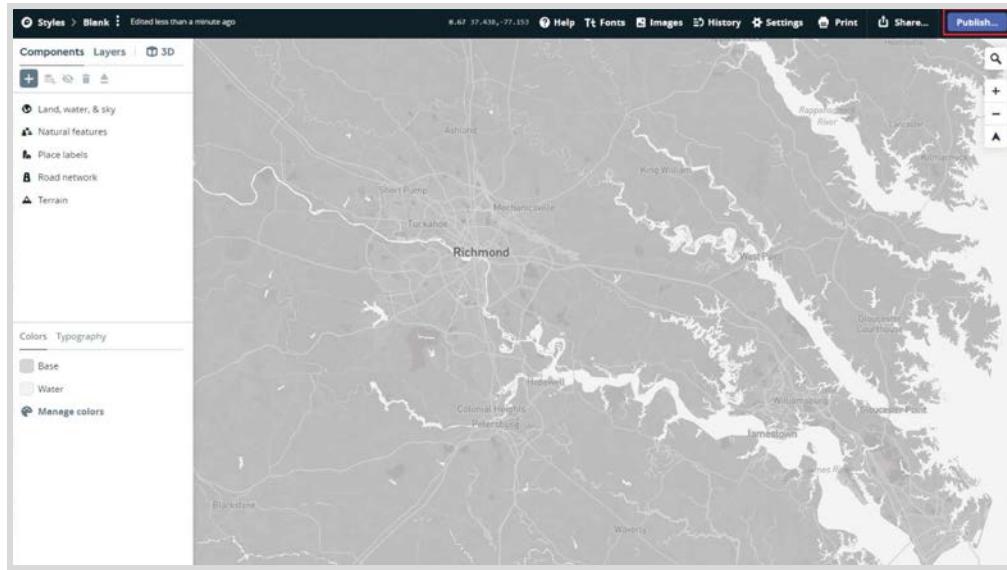
Play around with different styling options by clicking on these components and scrolling down to see the options. **Note: Label components are separated from Line and Polygon components.** This means that you will style the label (font, color, etc.) separately from the polygons that they refer to.





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.

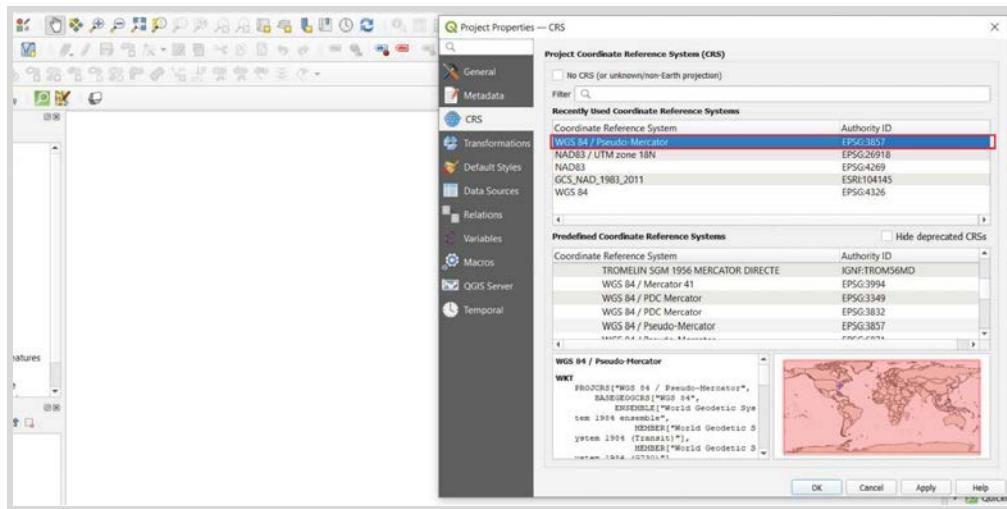
2c **Publish** your new Map style. This makes your changes live on the web, so you can see your updated map when you embed it. Note that when you make changes to your map, you will not be able to see these changes in your embedded map until you've Published them.



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 Mercador**. 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>

Select the year and layer you are interested in from the dropdown menus below and click "Submit" for a list of the available geographic areas.

Select year: 2010

Select a layer type: Counties (and equivalent)

Submit

Access our FTP site for additional downloading options

County and Equivalent (2000)
Select a State: All states in one national file ▾ Download

County and Equivalent (2010)
Select a State: Virginia ▾ Download

Source: US Census Bureau, Geography Division

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

B02001	All	2019	2018	2017	2016	2015	2014	2013	2012
ACS 1-Year Estimates Detailed Tables	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
ACS 5-Year Estimates Detailed Tables	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>					

File Type: CSV

What You're Getting:

- 2 .csv files (metadata)
- 2 .csv files (data)
- 2 .txt files (table title)

Compressed Size Estimate: 10.2 kB

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**).

A	B	C	D	E	F	G	H	I	J	K	L	
1	B02001_001E	B02001_001M	B02001_002E	B02001_002M	B02001_003E	B02001_003M	B02001_004E	B02001_004M	B02001_005E	B02001_005M	B02001_006E	B02001_006M
2	Estimate/Total	Margin of Error/	Estimate/Total/White alone	Margin of Error/	Estimate/Total/	Margin of Error/						
3	2965	20	2741	194	119	153	0	12	0	12	0	1
4	5523	625	3699	549	1509	435	6	14	0	17	0	1
5	2459	241	1883	212	447	157	0	12	12	16	0	1
6	6733	727	3842	652	2329	578	0	17	14	22	0	1
7	2854	368	2163	379	629	211	9	15	0	12	0	1
8	4126	405	3233	347	826	188	33	38	4	6	0	1
9	5146	581	2856	373	2169	479	82	69	0	17	0	1
10	3476	413	1995	337	1456	336	0	12	0	12	0	1
11	3	5	0	12	-	-	0	12	0	12	0	1
12	2	2	0	12	-	-	0	12	0	12	0	1

A	B	C	D	E	F	G
1	Total2013	WhiteOnly2013	GEO_ID			
2	33289	22712	0500000US51001			
3	100630	82359	0500000US51003			
4	16240	15174	0500000US51005			
5	12712	9171	0500000US51007			
6	32244	24747	0500000US51009			
7	15054	11578	0500000US51011			
8	214861	153109	0500000US51013			

GIS for Designers
VT | A+D | FA22

	STATEFP10	COUNTYFP10	COUNTYN10	GEOID10	NAME	CBSAfp10
1	51	001	01480091	51001	Accomack	NULL
2	51	003	01675170	51003	Albermarle	16620
3	51	005	01673675	51005	Alleghany	NULL
4	51	007	01497770	51007	Amelia	40060
5	51	009	01480095	51009	Amherst	31340
6	51	011	01497238	51011	Appomattox	31340
7	51	013	01480097	51013	Arlington	47900
8	51	015	01480098	51015	Augusta	44420
9	51	017	01673638	51017	Bath	NULL
10	51	019	01674818	51019	Bedford	31340
11	51	021	01946413	51021	Bland	NULL
12	51	023	01674418	51023	Botetourt	40220

	A	B	C	D	E	F	G
1	51	Total2013	WhiteOnly2013	GEO_ID			
2	51	33289	22712	5000000US51001			
3	51	100638	82359	5000000US51003			
4	51	16240	15174	5000000US51005			
5	51	12712	9171	5000000US51007			
6	51	32244	24747	5000000US51009			
7	51	15054	11578	5000000US51011			
8	51	214861	153109	5000000US51013			
9	51	73726	68628	5000000US51015			
10	51	4686	4362	5000000US51017			
11	51	69175	63281	5000000US51019			
12	51	6795	6442	5000000US51021			
13	51	33076	31297	5000000US51023			
14	51	17220	1149	5000000US51025			
15	51	23920	23032	5000000US51027			
16	51	17126	10688	5000000US51029			
17	51	54967	45262	5000000US51031			
18	51	28757	19109	5000000US51033			
19	51	29979	29286	5000000US51035			
20	51	7205	3027	5000000US51036			
21	51	12478	8415	5000000US51037			
22	51	320430	222794	5000000US51041			
23	51	14191	12860	5000000US51043			
24	51	5199	5041	5000000US51045			

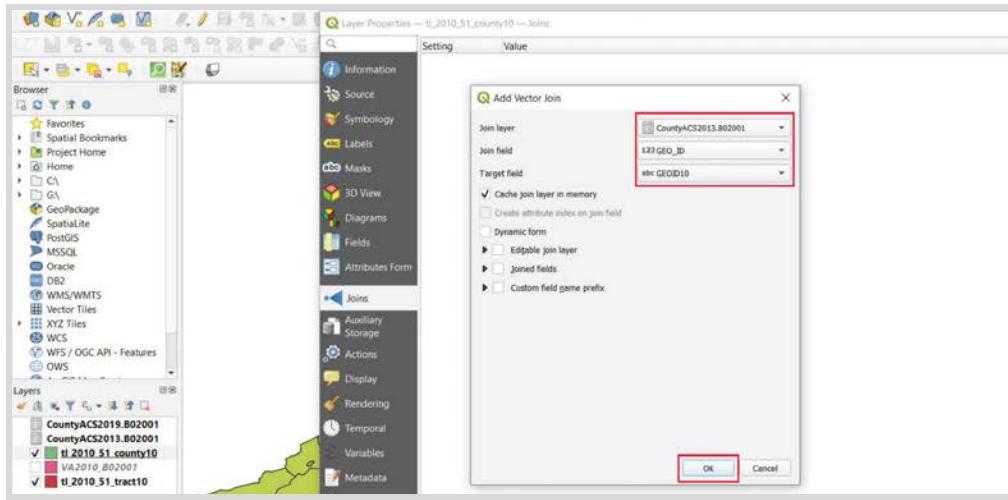
	A	B	C
1	C10	51017	
2	1	Total2013	WhiteOnly2013
3	2	33289	22712
4	3	100638	82359
5	4	16240	15174
6	5	12712	9171
7	6	32244	24747
8	7	15054	11578
9	8	214861	153109
10	9	73726	68628
11	10	4686	4362
12	11	69175	63281
13	12	6795	6442
14	13	33076	31297
15	14	17220	1149
16	15	23920	23032

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.

Untitled Project — QGIS

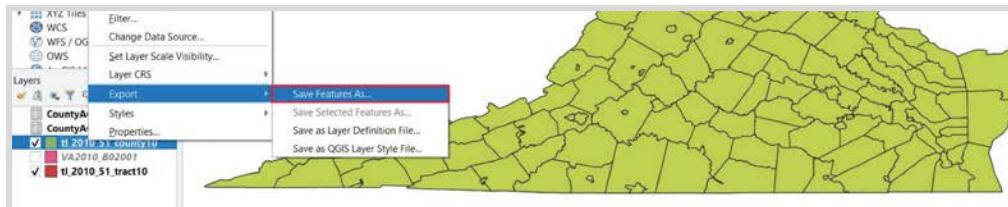
- Project
- Edit
- View
- Layer**
- Settings
- Plugins
- Vector
- Raster
- Database
- Web
- Mesh
- Processing
- Help

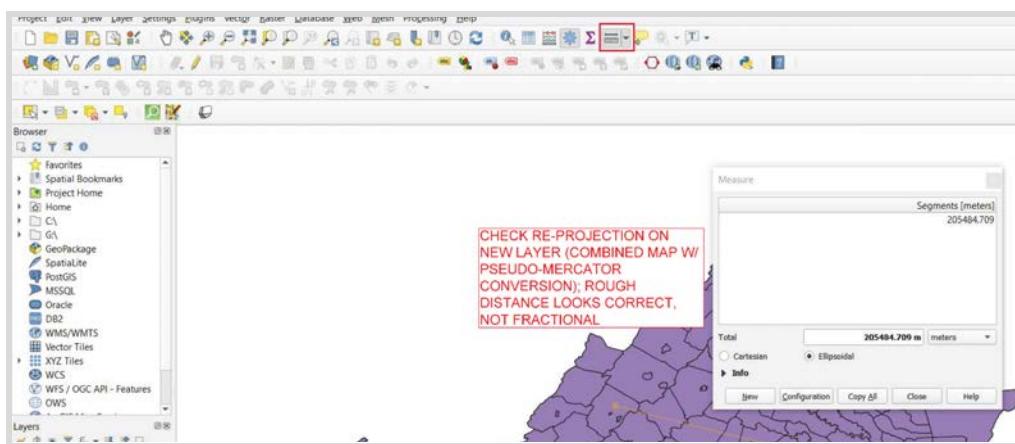
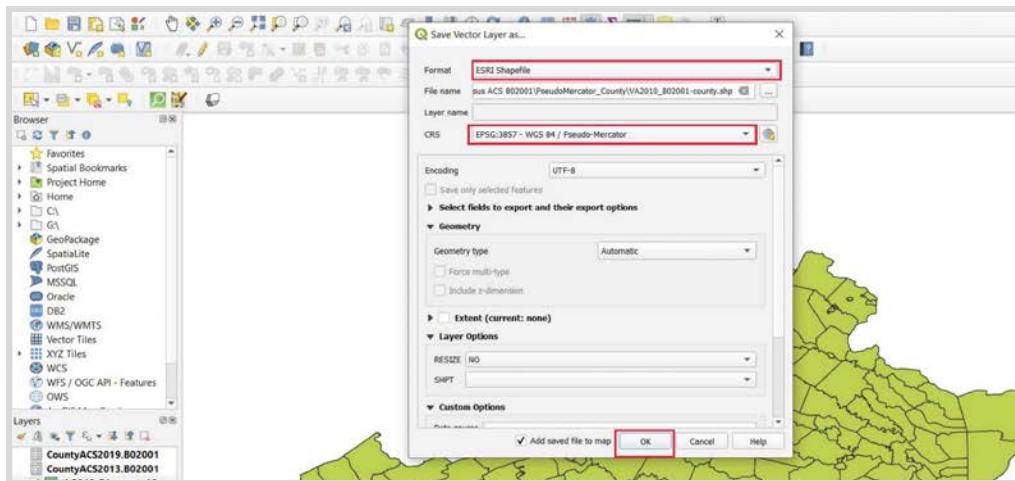
- Data Source Manager
- Create Layer
- Add Layer
- Add Vector Layer... **Ctrl+Shift+V**
- Add Raster Layer... **Ctrl+Shift+R**
- Add Mesh Layer...
- Add Delimited Text Layer... **Ctrl+Shift+T**
- Add PostGIS Layers...
- Add Spatialite Layer...
- Add MSSQL Spatial Layer...
- Add DB2 Spatial Layer...
- Add Attribute Table



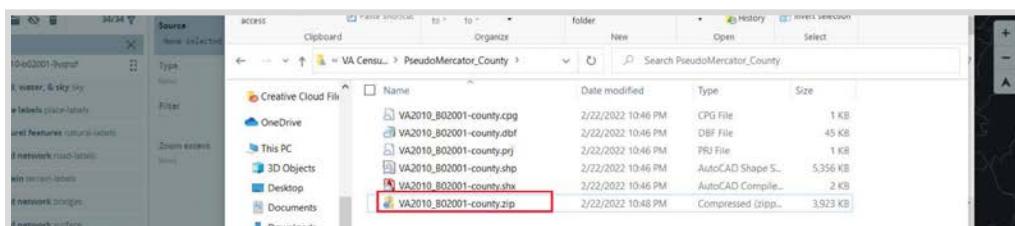
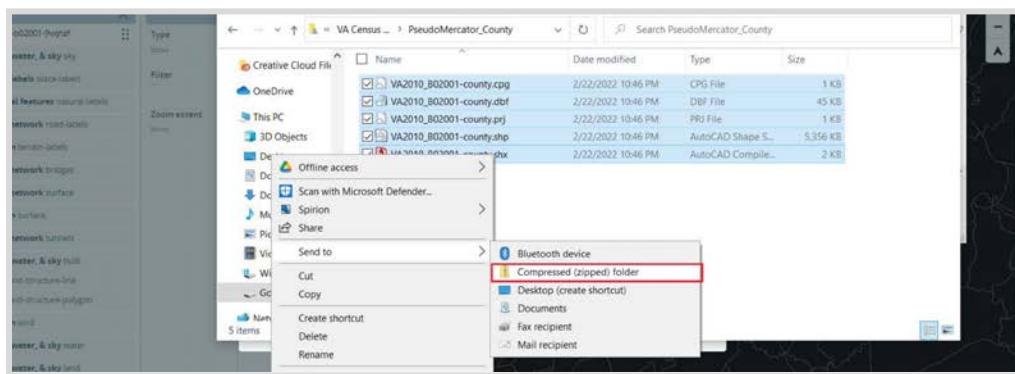
	MTFCC10	CSAFTP10	CBSAFTP10	METONVP10	FUNCSTAT10	ALAND10	AINWATER10	INTPTLAT10	INTPTLON10	CS2013.B02001_Tl_2013.B02001_Whit.CS2019.B02001_Tl_2019.B02001_Whit
1	020	NULL	NULL	NULL	A	1164189289	2228794013 + 37.7659435	-075.7578073		33289 22712 32673 22882
2	020	NULL	16820	NULL	A	1866599005	14035672 + 38.0241840	-078.551505073		100636 82359 107405 87831
3	020	NULL	NULL	NULL	A	1153727136	8535515 + 37.7879047	-080.0066889		16240 15174 15157 14012
4	020	NULL	40060	NULL	A	92042596	8424808 + 37.33631313	-077.9732177		12712 9171 12953 9824
5	020	NULL	31340	NULL	A	1227484444	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	-077.1057034		214861 153109 233464 166912
8	020	NULL	44420	NULL	A	2504515631	9987990 + 38.1678073	-079.1466816		73726 66628 75079 69533
9	020	NULL	NULL	NULL	A	1370512659	14049862 + 38.0689876	-079.7328980		4686 4362 4307 3933
10	020	NULL	31340	NULL	A	1990314965	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



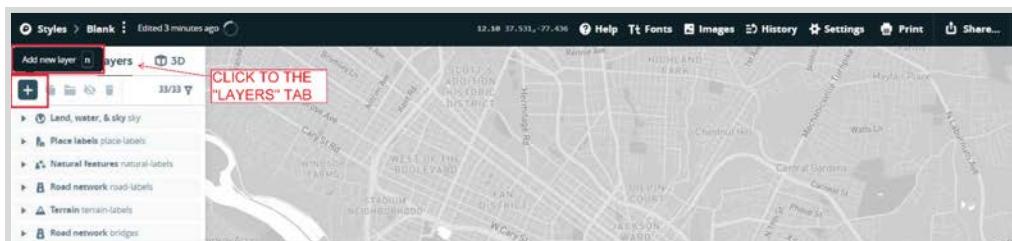


3g Zip your Shapefile files.

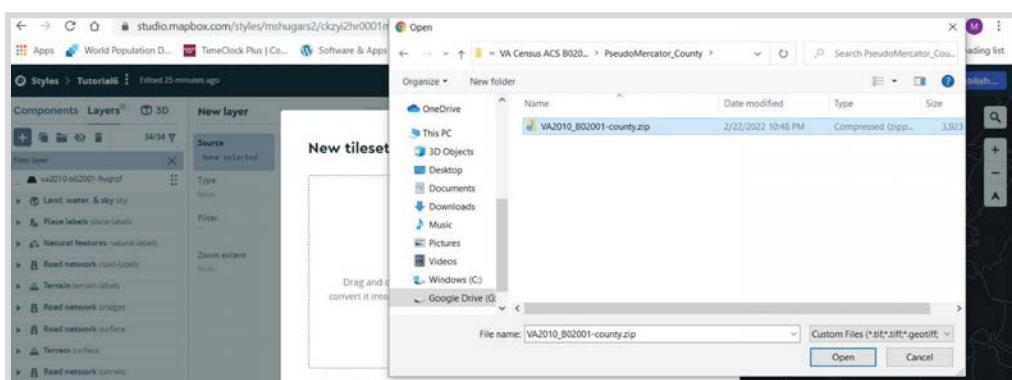
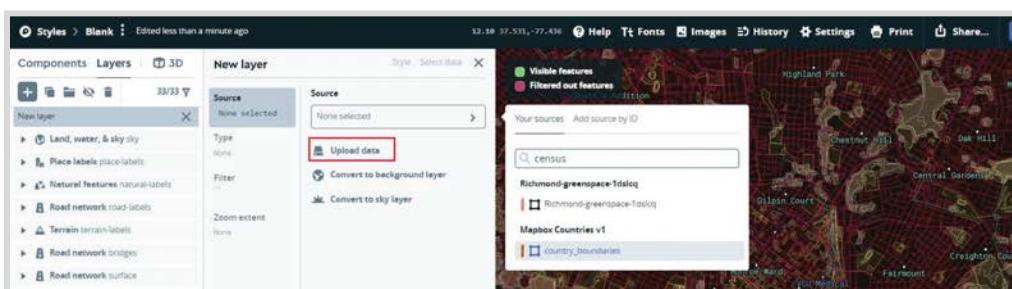


Step 4: Add your zipped Shapefile as a layer to your Mapbox base map.

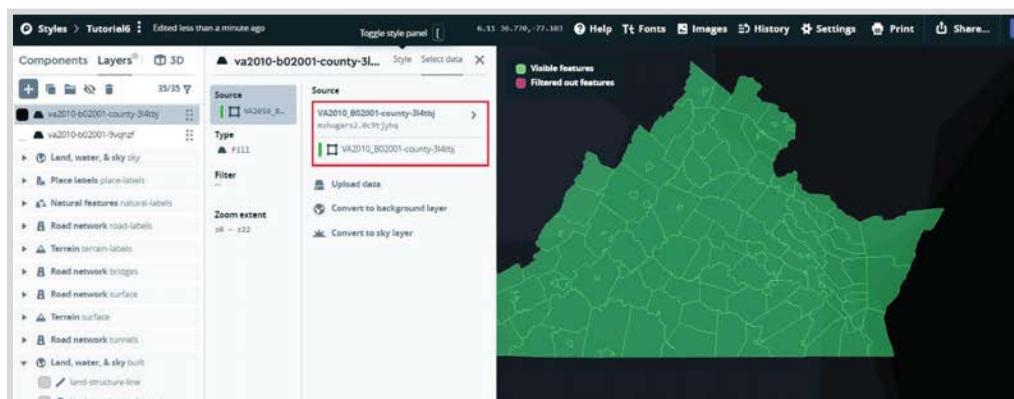
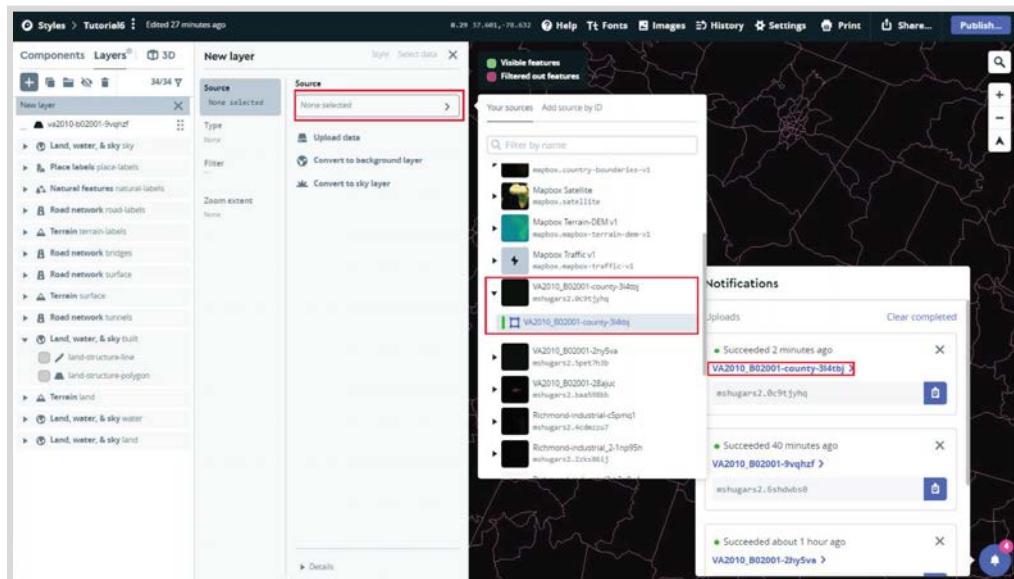
4a Under the **Layers** tab in your Studio view, click the plus sign “Add New Layer”



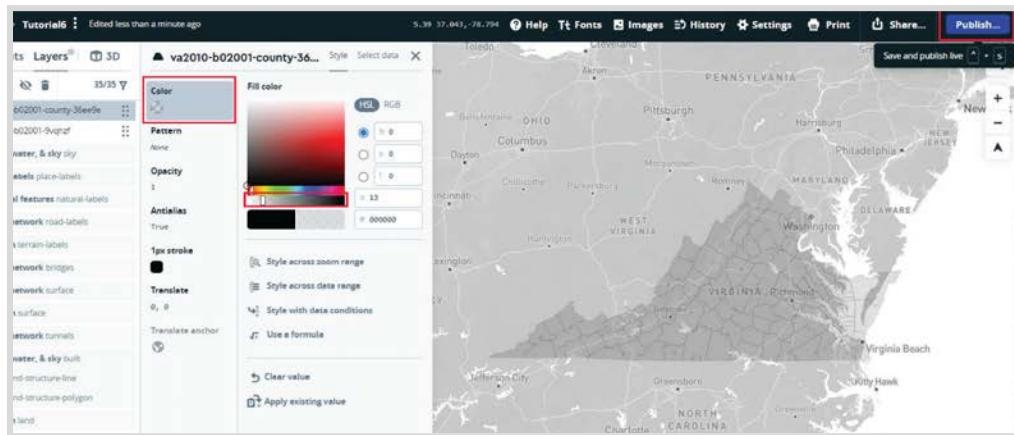
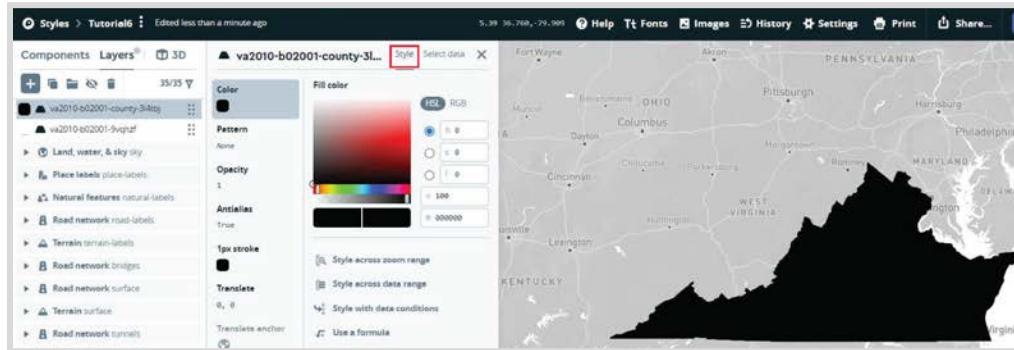
4b In Source, select “Upload Data” and select your zipped shapefile.



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.

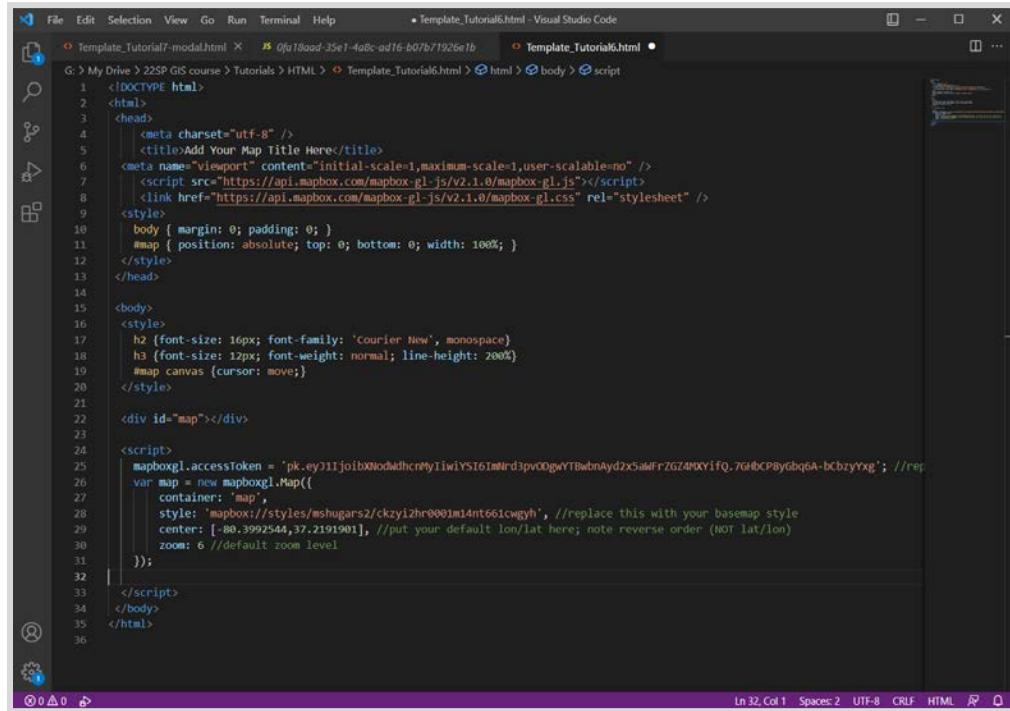


4e Publish your basemap changes (upper right button).

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:

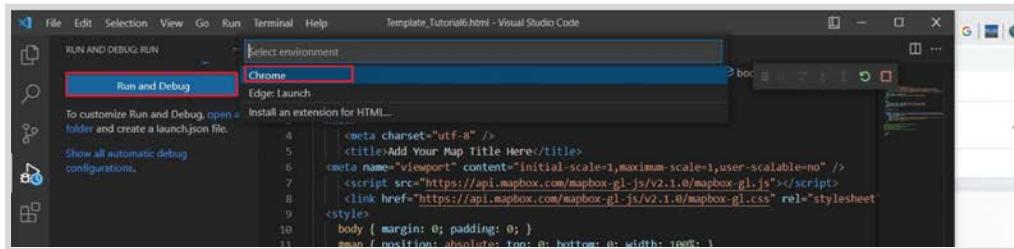


```
File Edit Selection View Go Run Terminal Help • Template_Tutorial6.html - Visual Studio Code
G: My Drive > 22SP GIS course > Tutorials > HTML > Template_Tutorial6.html > Template_Tutorial6.html •
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8" />
5     <title>Add Your Map Title 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   <style>
10    body { margin: 0; padding: 0; }
11    #map { position: absolute; top: 0; bottom: 0; width: 100%; }
12  </style>
13 </head>
14 <body>
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 </body>
21 <div id="map"></div>
22 <script>
23  .mapboxgl.accessToken = 'pk.eyJ1IjoibXlodKdhcnMyIiwibT6ImNrd3pvODgwYTBwbnAyd2x5aiFrZG74MDYifQ.7GhbCP8yBqg6A-bCbzyXg'; //replace this with your access token
24   var map = new mapboxgl.Map({
25     container: 'map',
26     style: 'mapbox://styles/mshugars2/ckzyi2hr0001m14nt66icwgyh', //replace this with your basemap style
27     center: [-80.3992544,37.2191901], //put your default lon/lat here; note reverse order (NOT lat/lon)
28     zoom: 6 //default zoom level
29   });
30 </script>
31 </body>
32 </html>
```

Ln 32 Col 1 Spaces: 2 UTF-8 CRLF 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):

```
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   <style>
10    body { margin: 0; padding: 0; }
11    #map { position: absolute; top: 0; bottom: 0; width: 100%; }
12  </style>
13 </head>
14
15 <body>
16   <h2>font-size: 16px; font-family: 'Courier New'; monospace</h2>
17   <h3>font-size: 12px; font-weight: normal; line-height: 200%</h3>
18   <#map canvas (cursor: move);>
19   </style>
20
21   <div id="map"></div>
22
23 <script>
24   mapboxgl.accessToken = "pk.eyJ1IjoibXodWdhcnMyIiw1YS16ImNr3pv0OgvY1bwbnAyd2x5awFrZGZ4MXyfQ.7GhBCP8ygbq6A-bcbzyVxp";
25   var map = new mapboxgl.Map({
26     container: "map",
27     style: "mapbox://styles/mshugars2/ckwzoc1b11eur14ni0ooxn8dm", //replace this with your basemap style
28     center: [17.207, 10.88], //put your default lat/lon here
29     zoom: 8 //default zoom level
30   });
31
32
33 /*map.addControl(new mapboxgl.NavigationControl());*/
```

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.

Access tokens

You need an API access token to configure Mapbox GL JS, Mobile, and Mapbox web services like routing and geocoding. Read more about [API access tokens](#) in our documentation.

+ Create a token

Integrate Mapbox

Design in Mapbox Studio

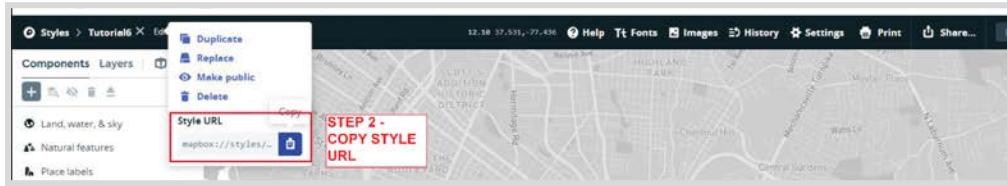
Documentation

Logout

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`:

```
24 mapboxgl.accessToken = "pk.eyJ1IjoibXodWdhcnMyIiw1YS16ImNr3pv0OgvY1bwbnAyd2x5awFrZGZ4MXyfQ.7GhBCP8ygbq6A-bcbzyVxp";
```

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:



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).



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</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  <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.eyJ1IjoibXNodWdhcnMyIiwiYSI6ImNrd3pvODgwYTBwbnAyd2x5aMFrZGZ4MXif
21 |   var map = new mapboxgl.Map({
22 |     container: 'map',
23 |     style: 'mapbox://styles/mshugars2/ckzyixxxxxxxxx', //replace this with your basemap style
24 |     center: [-80.3992544,91], //put your 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
30 </script>
```

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:

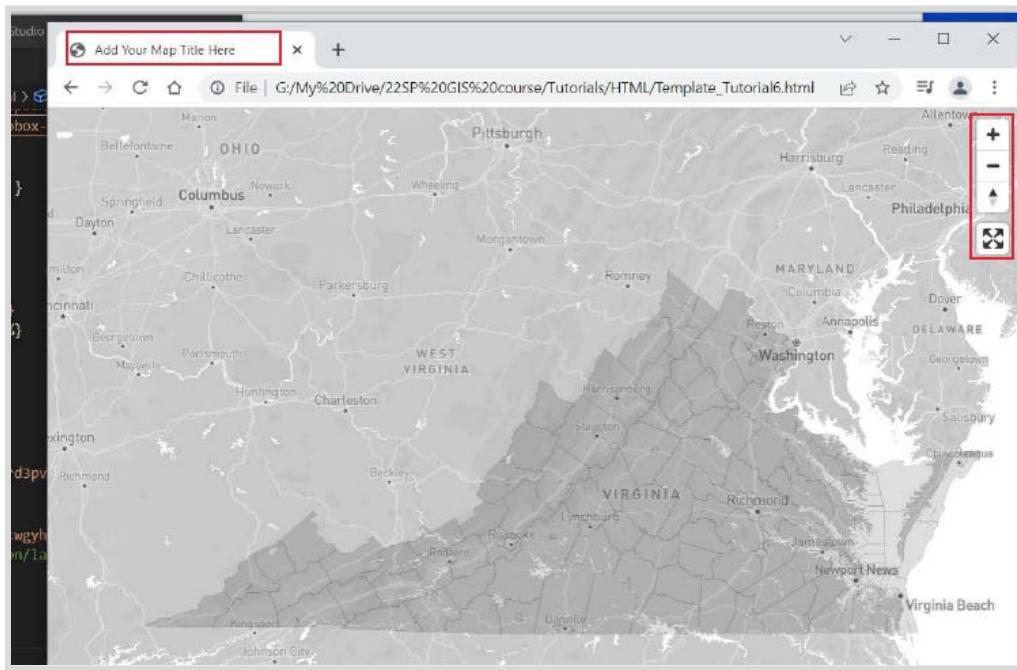
```
#map canvas {cursor: move;}
```

```
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</title>
```

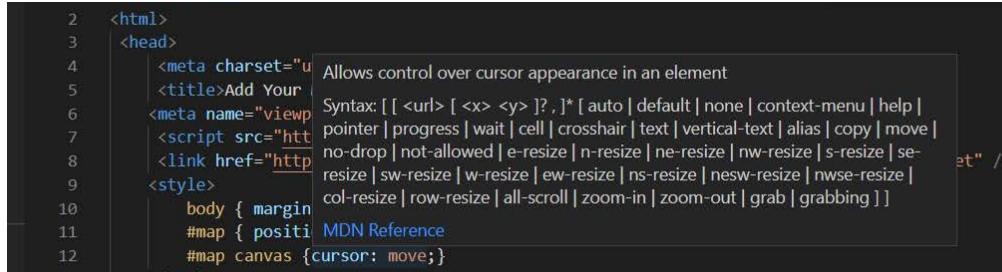
7d Now check out the changes in your Chrome map. Pan and zoom around to try out the new controls.



- 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:



A screenshot of Visual Studio's code editor showing a tooltip for the 'cursor' CSS property. The tooltip provides the following information:

- Allows control over cursor appearance in an element
- Syntax: [[<url> [<x> <y>]? ,]* [auto | default | none | context-menu | help | pointer | progress | wait | cell | crosshair | text | vertical-text | alias | copy | move | no-drop | not-allowed | e-resize | n-resize | ne-resize | nw-resize | s-resize | se-resize | sw-resize | w-resize | ew-resize | ns-resize | nesw-resize | nwse-resize | col-resize | row-resize | all-scroll | zoom-in | zoom-out | grab | grabbing]]
- [MDN Reference](#)

```
2  <html>
3  <head>
4  <meta charset="u Allows control over cursor appearance in an element
5  <title>Add Your Syntax: [[ <url> [ <x> <y> ]? , ]* [ auto | default | none | context-menu | help | pointer | progress | wait | cell | crosshair | text | vertical-text | alias | copy | move | no-drop | not-allowed | e-resize | n-resize | ne-resize | nw-resize | s-resize | se-resize | sw-resize | w-resize | ew-resize | ns-resize | nesw-resize | nwse-resize | col-resize | row-resize | all-scroll | zoom-in | zoom-out | grab | grabbing ]]
6  <meta name="viewp >
7  <script src="htt >
8  <link href="http >
9  <style>
10 body { margin >
11 #map { positi >
12 #map canvas {cu >
```

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/>