

## TUTORIAL 8 | MAPBOX SCROLLYTELLING TEMPLATE

Zoom: **Wednesday 06.22, 6:30-7:30pm**  
<https://virginiatech.zoom.us/j/2981092726>

### Intro

In this week's tutorial you'll upload your Map Assignment 3 research layers from QGIS to Mapbox, adjust the style of your basemap to match, and then put your new map into the Mapbox scrollytelling template.

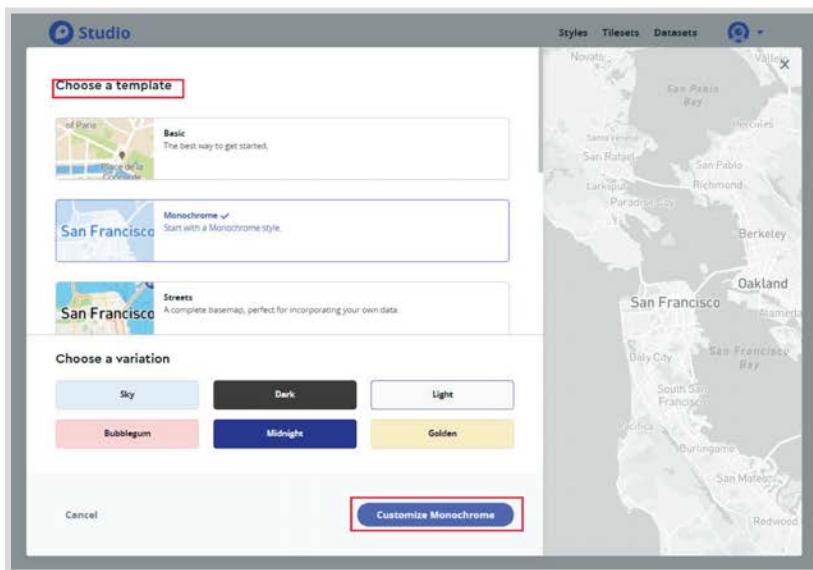
Before you start this tutorial, please first check out this page for more information on the Mapbox template:  
<https://demos.mapbox.com/scrollytelling/>

For a refresher on how to upload layers to Mapbox, please scroll back through Tutorials 6 & 7.

**Note: throughout this tutorial keep in mind that it can take Mapbox a while to update!! This means that your published changes might not be visible for several minutes.**

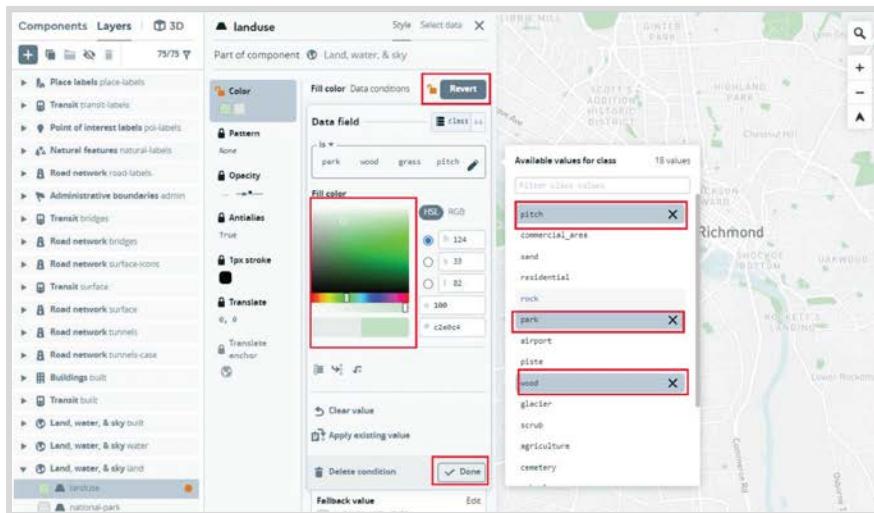
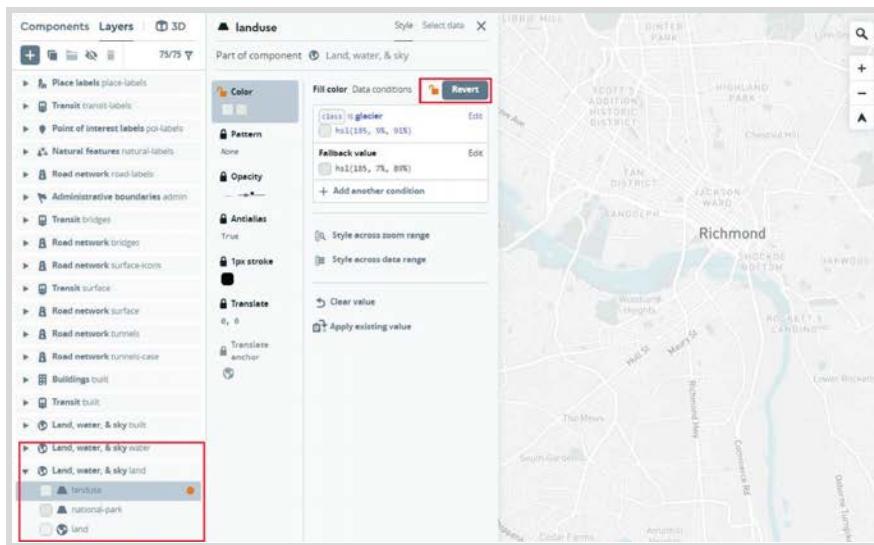
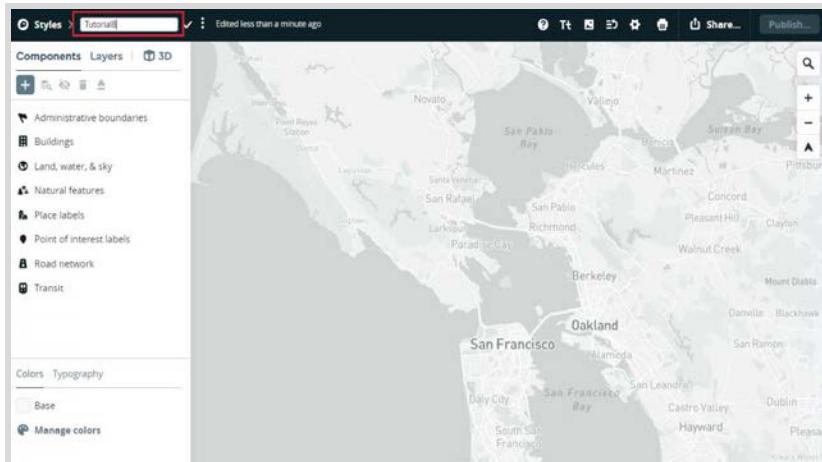
### Step 1: Upload your layers to a new Mapbox style

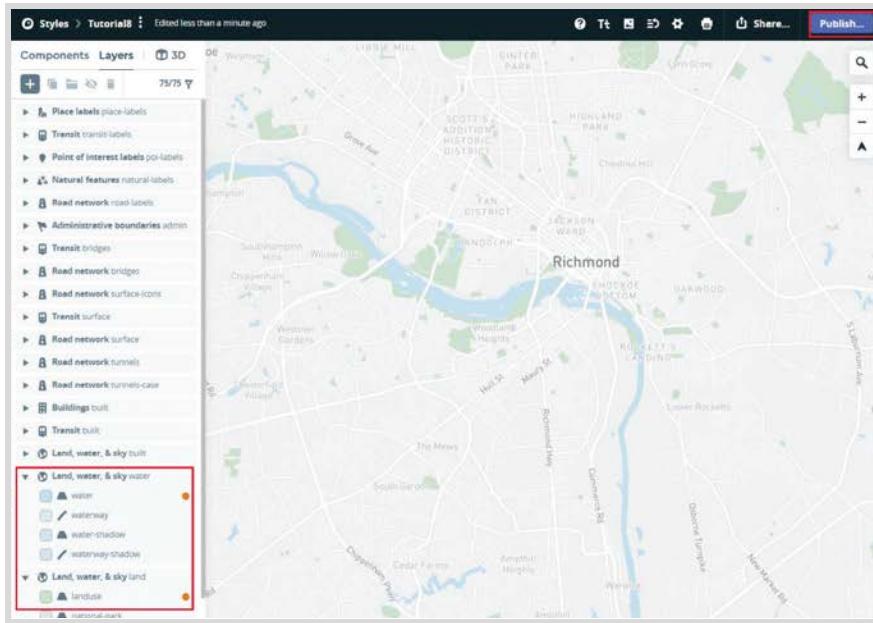
First, create a new Mapbox Style (basemap) to display your data. Either start with a blank map like we did before, or start with a template:



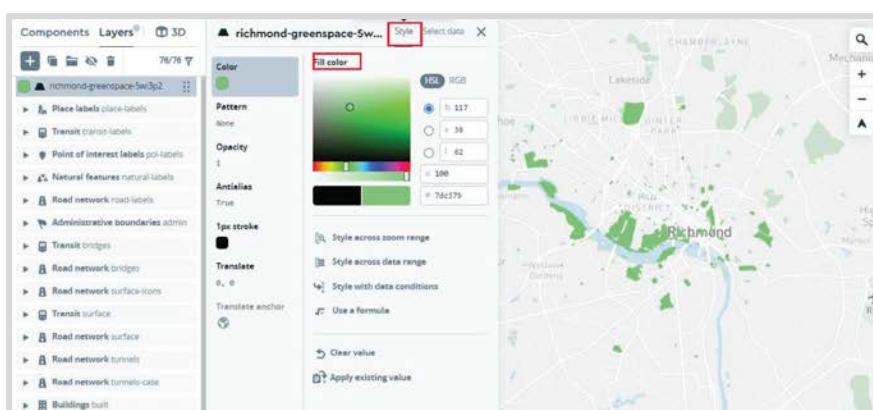
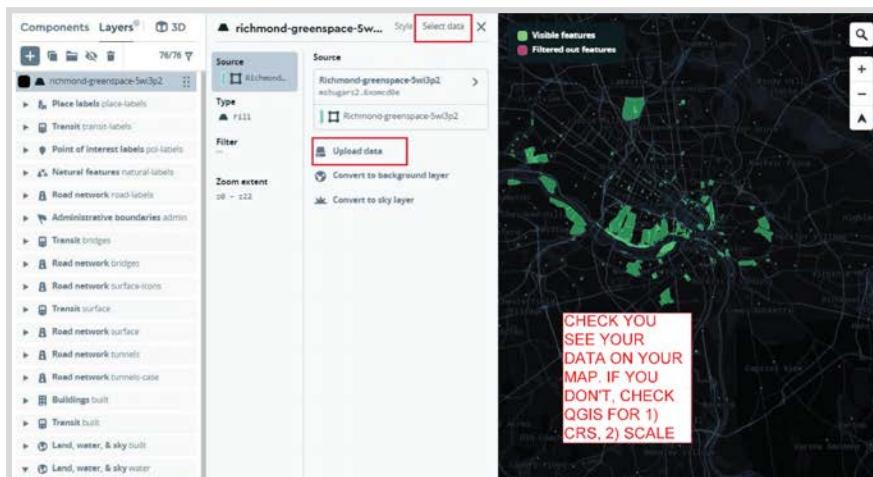
# GIS for Designers

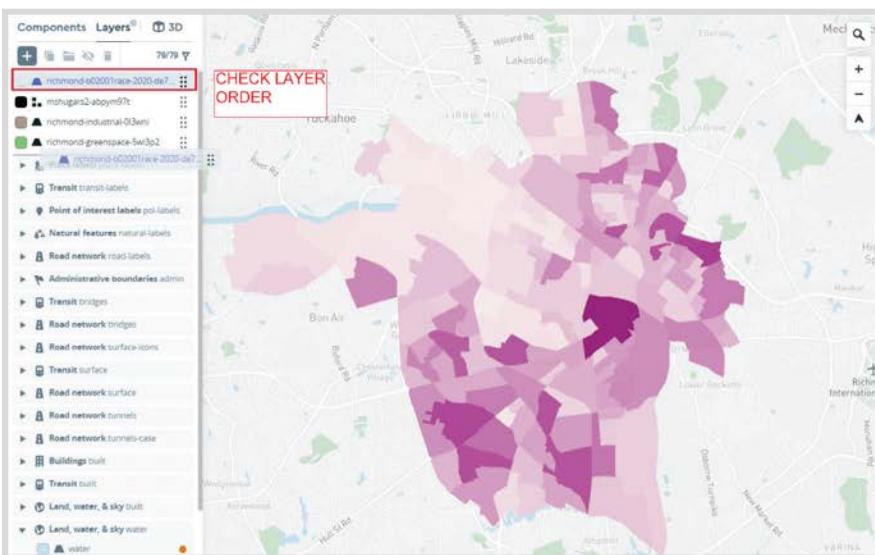
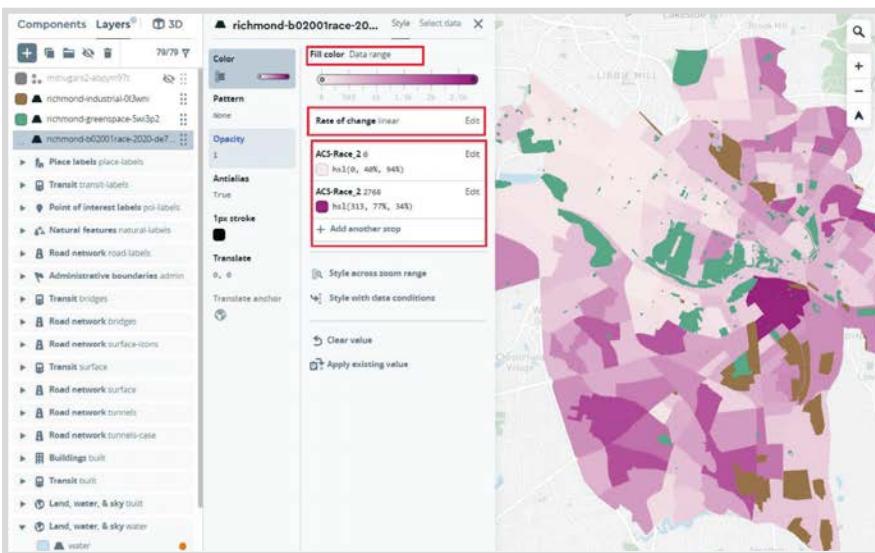
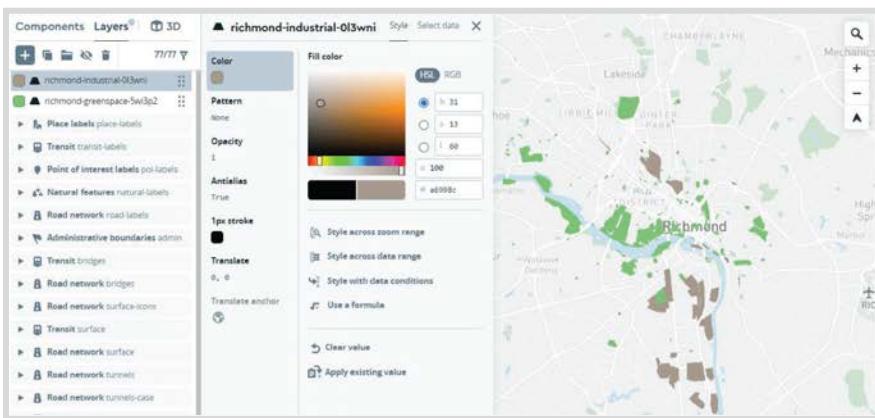
## VT | A+D | SU22





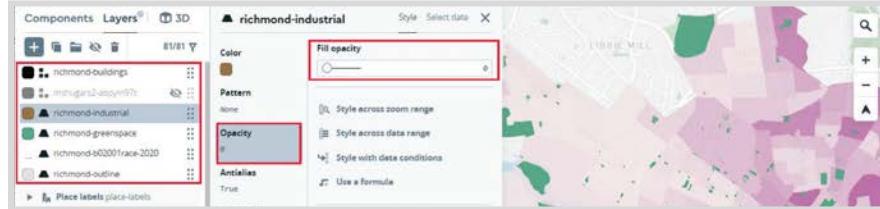
Next, upload your QGIS data layers to Mapbox (refer to Tutorial 6). Remember to set your CRS to **EPSG 3857: Pseudo Mercador** when you export:





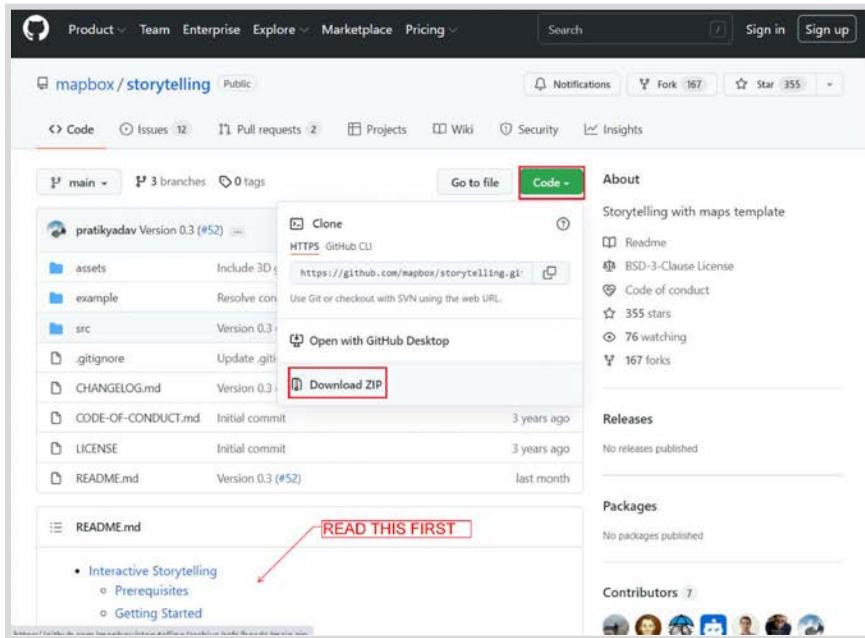
**Note:** If your QGIS layer is over 20MB large after zipping it, Mapbox will be unable to display it fully on the map. In this case, you may need to re-export your layers in separate parts. You can also try the QGIS tool Vector > Geometry Tools > Simplify.

**Note:** All layers you want to use should be shown, with their opacity set to 0:

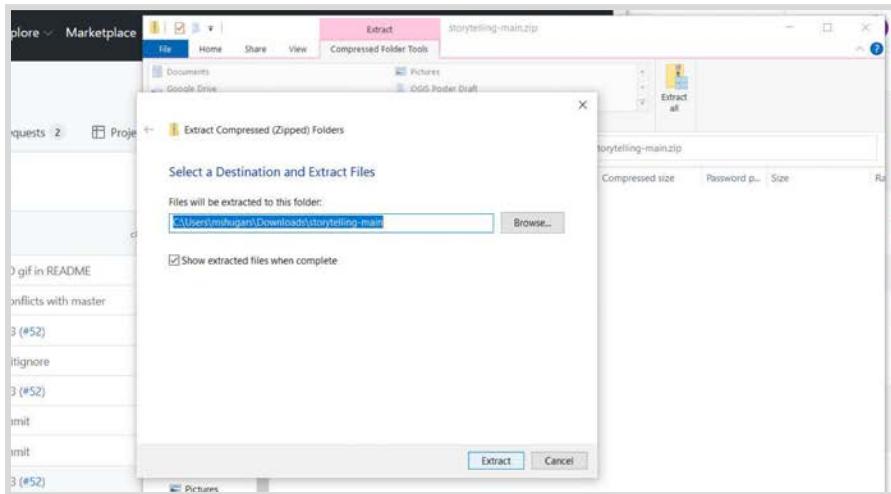


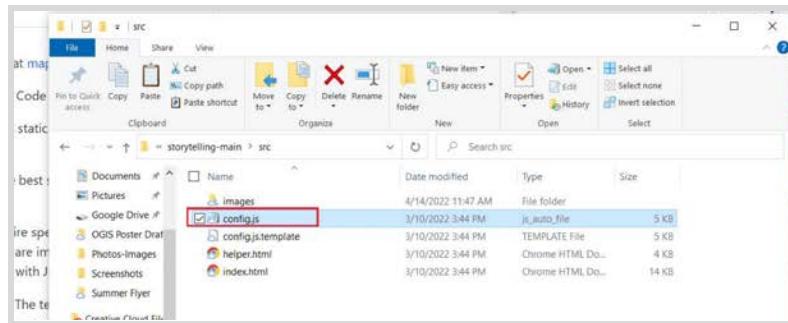
**Step 2: Download the zipped Mapbox Scrollytelling template file from: <https://github.com/mapbox/storytelling>**

**Note:** Read through the README.md part of the Github page first; this tutorial roughly follows the instructions here, but you will also see feature options for this Template:



Unzip the file and open the folder “src/”. From here, copy “config.js.template” and rename it “config.js”. Open this new file in Visual Studio Code:





**Note:** also open your “index.html” file in Visual Studio to preview the results in your web browser as you go.

### **Step 3: Update the config file**

As before, add your Mapbox access token url and new Mapbox style url to the template. Update the “title”, “subtitle”, “byline”, and “footer” with your project information:

```
File Edit Selection View Go Run Terminal Help config.js - Visual Studio Code 08 - config.js config G: My Drive > 22SP GIS course > Tutorials > ScrollTelling > storytelling-main > src > config.js > config
1 var config = {
2   style: 'mapbox://styles/mapbox/streets-v11',
3   accessToken: 'YOUR_ACCESS_TOKEN',
4   showMarkers: true,
5   markerColor: '#3FBC1E',
6   //projection: 'equirectangular',
7   //Read more about available projections here
8   //https://docs.mapbox.com/mapbox-gl-js/example/projections/
9   inset: true,
10  theme: 'dark',
11  use3dTerrain: false, //Set true for enabling 3D maps.
12  title: 'The Title Text of this Story',
13  subtitle: 'A descriptive and interesting subtitle to draw in the reader',
14  byline: 'By a Digital Storyteller',
15  footer: 'Source: source citations, etc. cbx Created using <a href="https://github.com/charlesbennett91/storytelling">Storytelling</a>',
16  chapters: [
17    {
18      id: 'slug-style-id',
19      alignment: 'left',
20      hidden: false,
21      title: 'Display Title',
22      image: './path/to/image/source.png',
23      description: 'Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.',
24      location: {
25        center: [-122.418398, 37.75948],
26        zoom: 8.5,
27        pitch: 60,
28        bearing: 0
29      },
30      mapAnimation: 'flyTo',
31      rotateAnimation: false,
32      callback: '',
33      onchapterEnter: [
34        // [
35        //   layer: 'layer-name',
36        //   opacity: 1,
37        //   duration: 5000
38        // ]
39      ],
40      onchapterExit: [

```

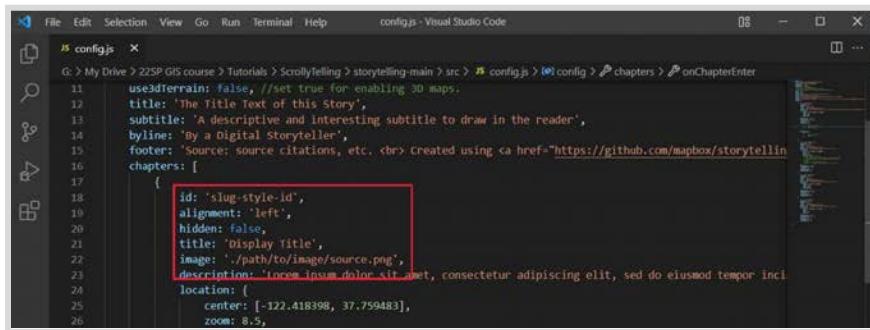
#### Step 4: Add 4 chapters to your map as practice

You'll see that the template includes 4 chapters by default.

The default chapters 1 and 2 walks us through several of the chapter features, so first let's take a look at this. Note that, in javascript, you can comment out text with two backslashes (//) – this means that the code compiler won't read that part as code. Programmers use this to embed comments in their code. In chapters 1 and 2 of this template you can see several comments that either give you optional code (in chapter 1), or explain optional features (in chapter 2).

So, let's look at the parts of one of these chapters. First, we see "id" – this is used to refer to the chapter in other parts of your code. For example, it can be used as an anchor, which means that a hyperlink at the top of the page could take you straight down to the beginning of the chapter. This is a useful way to navigate long scrolling pages like this.

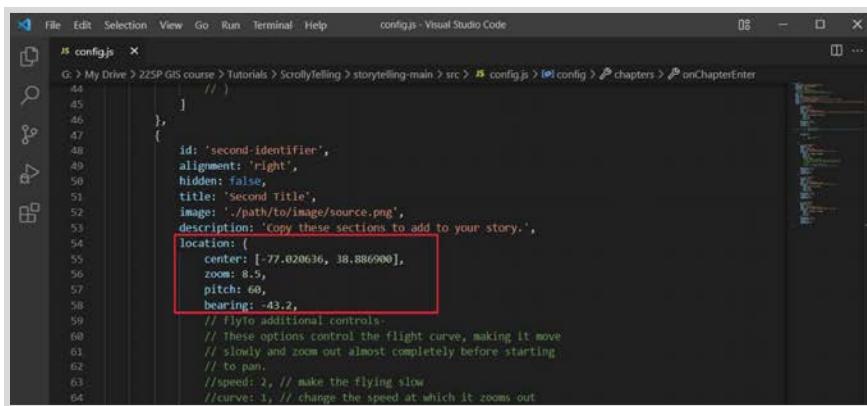
Next, we see "alignment" – which refers to the chapter text box alignment; "hidden" – can you see the text; "title" – what is the title of your chapter; "image" – chapter image; "description" – chapter content. That's all pretty straightforward text editing. For more text editing options, like font, size, and color, check out the index.html file.



A screenshot of Visual Studio Code showing the config.js file. The code defines a chapter object with properties: id ('slug-style-id'), alignment ('left'), hidden (false), title ('Display Title'), image ('./path/to/image/source.png'), and description ('Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.'). The location object contains center coordinates (-122.418398, 37.759483), zoom level (8.5), pitch (60), and bearing (-43.2). A red box highlights the chapter object.

```
11  use3dTerrain: false, //set true for enabling 3D maps.
12
13  title: 'The title Text of this Story',
14  subtitle: 'A descriptive and interesting subtitle to draw in the reader',
15  byline: 'By a Digital Storyteller',
16  footer: 'Source: source citations, etc. <br> Created using a href="https://github.com/mapbox/storytelling'
17  chapters: [
18    {
19      id: 'slug-style-id',
20      alignment: 'left',
21      hidden: false,
22      title: 'Display Title',
23      image: './path/to/image/source.png',
24      description: 'Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.',
25      location: {
26        center: [-122.418398, 37.759483],
27        zoom: 8.5,
```

After that, we see the section that locates the chapter on your map, "location". You can see the center will be your lon/lat; zoom level; pitch and bearing (discussed below).



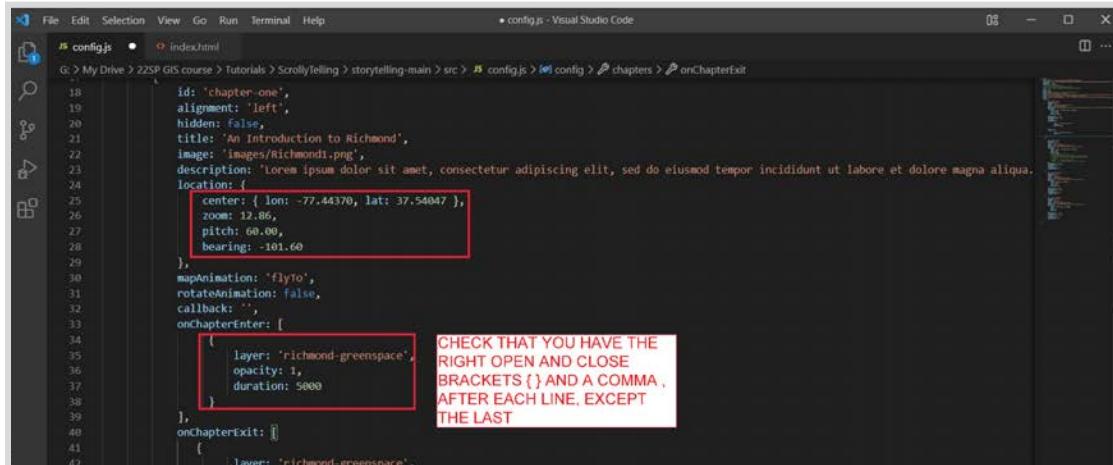
A screenshot of Visual Studio Code showing the config.js file. The code defines a chapter object with properties: id ('second-identifier'), alignment ('right'), hidden (false), title ('Second Title'), image ('./path/to/image/source.png'), and description ('Copy these sections to add to your story.'). The location object contains center coordinates (-77.020636, 38.886900), zoom level (8.5), pitch (60), and bearing (-43.2). A red box highlights the chapter object.

```
44  ],
45  ],
46  },
47  {
48    id: 'second-identifier',
49    alignment: 'right',
50    hidden: false,
51    title: 'Second Title',
52    image: './path/to/image/source.png',
53    description: 'Copy these sections to add to your story.',
54    location: {
55      center: [-77.020636, 38.886900],
56      zoom: 8.5,
57      pitch: 60,
58      bearing: -43.2,
59      // flyto additional controls
60      // These options control the flight curve, making it move
61      // slowly and zoom out almost completely before starting
62      // to pan.
63      //speed: 2, // make the flying slow
64      //curve: 1, // change the speed at which it zooms out
```

Next, we see a few options for "mapAnimation" (how to animate between chapters), "rotateAnimation", and "callback"; for more information on these options, refer to the README.md on Github. For now, leave them as-is.

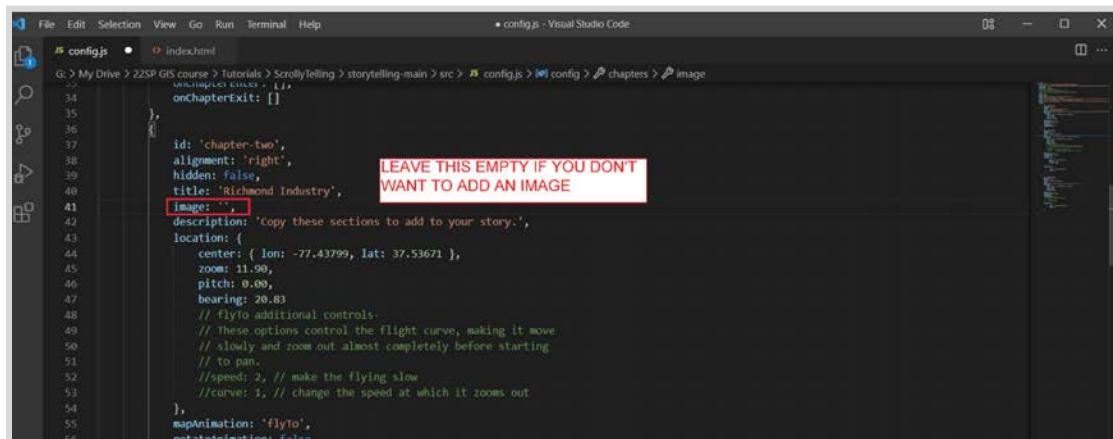
The last two sections are “onChapterEnter” and “onChapterExit”, which refers to transitions to and from your chapter. This is where you can instruct the map to turn on and off certain layers when moving between chapters. You’ll also see “duration”, which refers to the time in milliseconds to turn on or off the layer. Opacity 1 means it’s fully visible, Opacity 0 means it’s fully invisible. Make sure you turn on and off the appropriate layers when entering and leaving each chapter.

Fill out the information for each chapter. Set your layer transitions, trying out different speeds. Make sure each “id” name has no spaces or special characters. Keep these simple.



A screenshot of Visual Studio Code showing the file config.js. The code defines a chapter object with an id of 'chapter-one'. It includes properties for alignment ('left'), hidden status, title ('An Introduction to Richmond'), image URL ('images/richmond1.png'), and a long descriptive text ('Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.'). The 'location' property contains a nested object with 'center' coordinates (-77.44370, 37.54047), 'zoom' (12.86), 'pitch' (60.00), and 'bearing' (-101.60). The 'mapAnimation' is set to 'flyTo', 'rotateAnimation' is false, and there's an empty 'callback'. The 'onChapterEnter' section contains a block of code that toggles the visibility of a 'richmond-greenspace' layer with an opacity of 1 over a duration of 5000ms. The 'onChapterExit' section contains another block of code that turns off the same layer. A red box highlights the 'onChapterEnter' code, and a callout bubble says: "CHECK THAT YOU HAVE THE RIGHT OPEN AND CLOSE BRACKETS {} AND A COMMA , AFTER EACH LINE, EXCEPT THE LAST".

```
18     id: 'chapter-one',
19     alignment: 'left',
20     hidden: false,
21     title: 'An Introduction to Richmond',
22     image: 'images/richmond1.png',
23     description: 'Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.',
24     location: {
25       center: { lon: -77.44370, lat: 37.54047 },
26       zoom: 12.86,
27       pitch: 60.00,
28       bearing: -101.60
29     },
30     mapAnimation: 'flyTo',
31     rotateAnimation: false,
32     callback: '',
33     onChapterEnter: [
34       {
35         layer: 'richmond-greenspace',
36         opacity: 1,
37         duration: 5000
38       }
39     ],
40     onChapterExit: []
41   },
42   'layer': 'richmond_greenspace'
43 }
```

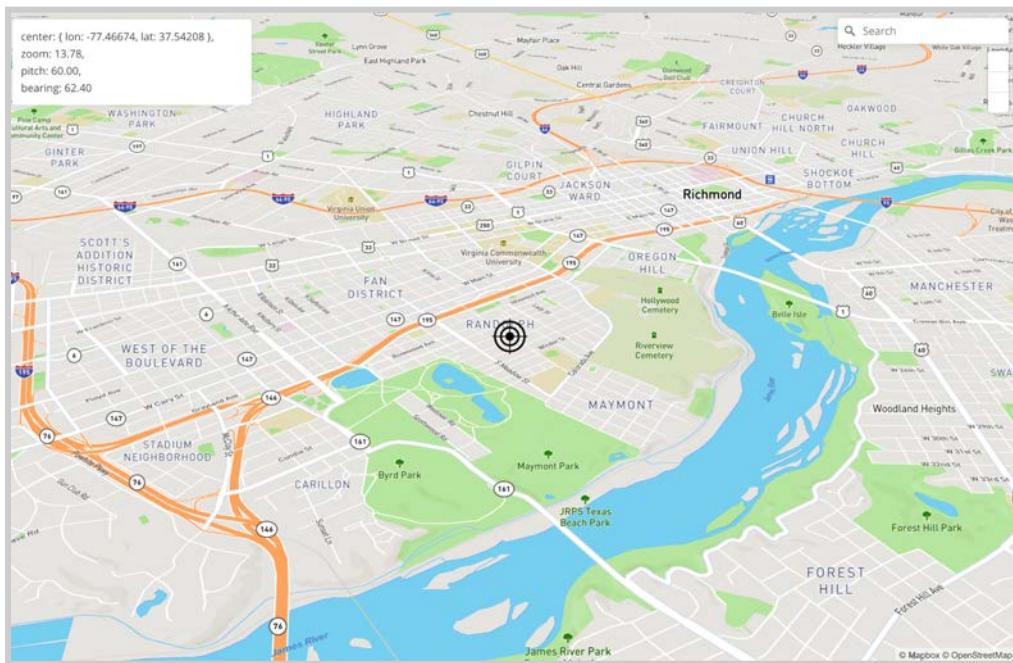


A screenshot of Visual Studio Code showing the file config.js. The code defines a chapter object with an id of 'chapter-two'. It includes properties for alignment ('right'), hidden status, title ('Richmond Industry'), and an empty 'image' URL (''). The 'description' field contains a note to copy sections for a story. The 'location' property contains a nested object with 'center' coordinates (-77.43799, 37.53671), 'zoom' (11.90), 'pitch' (0.00), and 'bearing' (20.83). The 'mapAnimation' is set to 'flyTo', and 'rotateAnimation' is false. A red box highlights the empty 'image' URL, and a callout bubble says: "LEAVE THIS EMPTY IF YOU DON'T WANT TO ADD AN IMAGE".

```
34   },
35   },
36   'id': 'chapter-two',
37   'alignment': 'right',
38   'hidden': false,
39   'title': 'Richmond Industry',
40   'image': '',
41   'description': 'Copy these sections to add to your story.',
42   'location': {
43     'center': { lon: -77.43799, lat: 37.53671 },
44     'zoom': 11.90,
45     'pitch': 0.00,
46     'bearing': 20.83
47   },
48   // flyTo additional controls:
49   // These options control the flight curve, making it move
50   // slowly and zoom out almost completely before starting
51   // to pan.
52   //speed: 2, // make the flying slow
53   //curve: 1, // change the speed at which it zooms out
54 },
55   mapAnimation: 'flyTo',
56   rotateAnimation: false
57 }
```

Along with Zoom level and Center, you can also choose a Pitch and Bearing (angle of the camera and direction of view); you can play around with these, or simply use Mapbox's location helper to find the location, pitch, and bearing (use right click to tilt the map):

<https://demos.mapbox.com/location-helper/>



Once you have this set up, open your index.html file in Chrome to preview the results.

#### Step 5: Cartoon your story map

Now comes a fun part: draw a series of panels (in your notebook or on your laptop) and sketch out your scrollytelling sequence in the panels – a cartoon set. Think about the order of information you want to present. Use your Map Assignment 3 maps as a basis for the story you want to tell.

Think about your story as having a beginning, middle, and end, like some of the examples we looked at. The beginning will tell us the general outline of the social issue that you are visualizing. It will tell us the where, when, and who. In other words, it will introduce (in text) the major data layers that you'll be showing. The middle will be your data layer maps, which start to set up possible explanations or correlations to describe the issue. The end will summarize your maps and conclude with some observations about the problem.

Think about how each of these parts of your story relates to a geographic level as well. For example, my story might be about the health impacts of pollution near New Orleans. I might start with a map zoomed out to show the entire area around New Orleans and the Gulf Coast, with the locations of petrochemical plants located on the map. The text here would describe the issue of airborne pollution on health, and the number of factories in this area, and the high cancer rates among residents: setting up the major datasets of my maps. The middle series of maps would be at a closer zoom level, and would show each of these variables, one by one and then maybe overlaid to compare the geographical location of each incidence: location of factories compared with location of high cancer rates. This middle section might also include quotes, statistics, and graphs (as embedded images) to support conclusions about the data correlation. The end might zoom back out, or might show all the layers overlaid together, and would conclude that the presence of these factories likely causes higher rates of cancer for these residents, and should be restricted.

Think about the sequence of your maps (or, panels in your cartoon set) as chapters, like in the template file.  
List out the following below each panel/chapter:

1. data layers shown
2. zoom level (county? state? national? city?)
3. brief text description (what part of your story does this map tell?)

**Note:** open the example maps in the Template folder (“example” folder; open the respective index.html files) for inspiration.

**Note:** In the next Tutorial, you’ll experiment with different styling options for your scrolllymap. Keep your map panel sequence loose and open to change, since you may decide to show different information or in a different order based on the capabilities of this template.

#### **Step 6: Screenshot map and cartoon panels**

For the Tutorial Assignment, please include a legible image of your cartoon panels, as well as screenshots of your scrolllymap at this stage. The screenshots should include each distinct chapter.

- no Bonus -