

## TUTORIAL 9 | MAPBOX SCROLLYTELLING TEMPLATE

### Goals

- Explore Mapbox scrollytelling template.
- Create draft chapters in template.
- Learn how to edit and view template web page.

### Intro

In this week's tutorial you'll put your Mapbox layers into the Mapbox scrollytelling template. You'll also cartoon out the rest of your scrollymap for the final Map Assignment 3 – planning out which layers you'll need to create and which parts of the final story are missing.

Before you start this tutorial, first check out this page for more information on the Mapbox template:

<https://demos.mapbox.com/scrollytelling/>

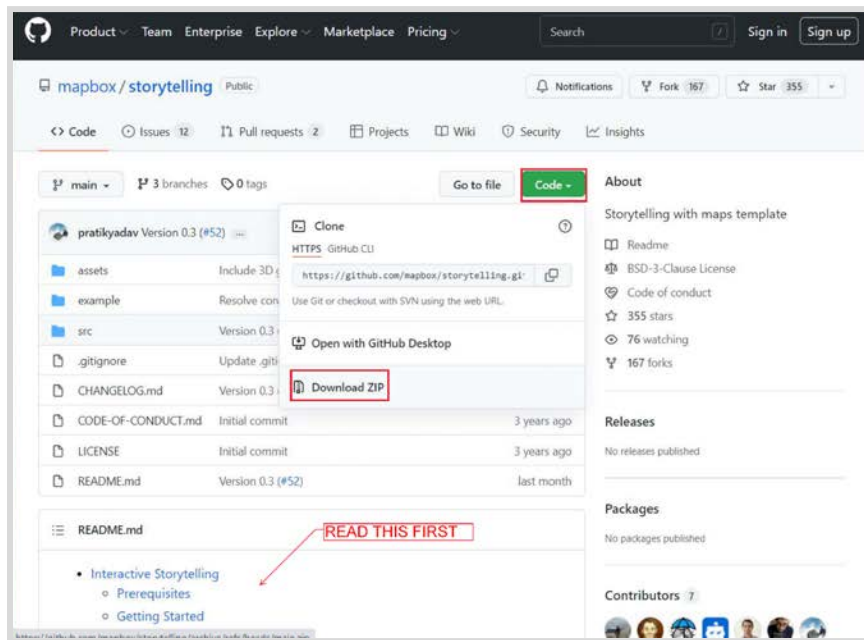
**Note: throughout this tutorial keep in mind that the Mapbox website can take a while to update!! This means that your published map changes might not be visible in the template for several minutes.**

**Step 1: Download and install Visual Studio Code (this will be our code editor):**

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

**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. It's short and informative. This tutorial roughly follows the instructions, but you will find additional feature options for this Template:



2a First, we're going to preview an example storymaps.

**Download and unzip** an example final project from the Data folder:

<https://drive.google.com/drive/folders/13BL083YBijwuX9PsXCNRx4QYfSowi01P?usp=sharing>

Inside the "src" folder, open the "index.html" file in your web browser (Google Chrome or similar). Scroll down to preview the example.

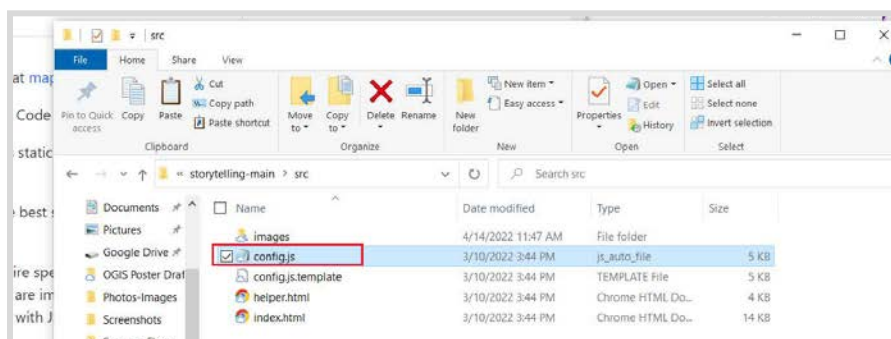
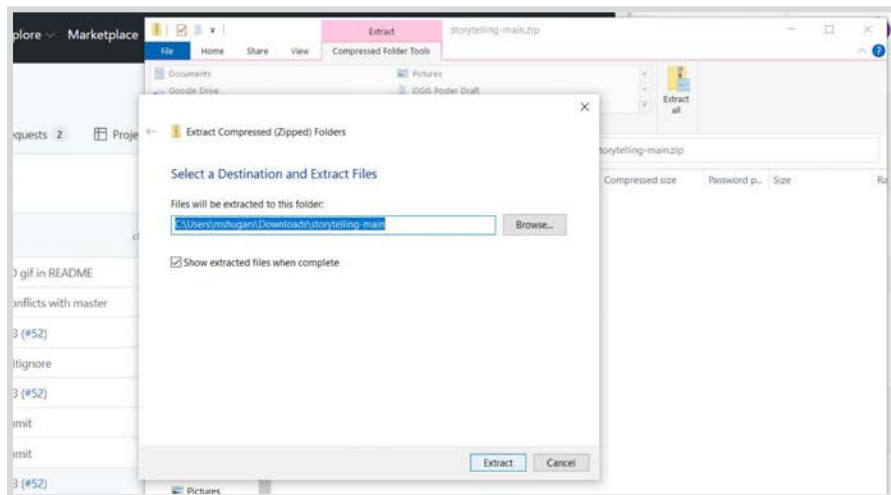
2b Now, open the example's **config.js** file in Visual Studio Code. Although the index.html file is what your web browser reads, the config.js file will contain your edits, text, layers, and so on. If you open the index.html file in Visual Studio, you'll see the link of the config.js file. Basically, the html file links in and displays the config.js information.

Scroll through the example's config.js file side by side with the scrollymap in your web browser. Pay attention to each "Chapter" section. Though it looks complicated, take a few minutes to suss out the plain English involved. Find the locations of each chapter's text, the different map layers turned on and off using the opacity (set to 1 for on and 0 for off). See how images are added to the template through image links, and check out how some of those images are stored in the "assets" folder and use local links (folder paths) instead.

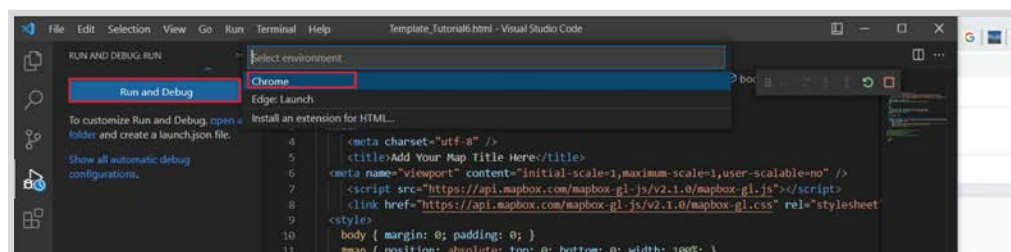
### Step 3: Open and update the config.js file.

We'll start with some basic edits to the config file, such as adding in your Mapbox access token and your Map Style url (the map you created on Mapbox in the previous tutorial). Then, we'll take a closer look at the structure of the "chapter" sections, and you'll add in some chapters of your own.

3a First, unzip the Mapbox template and open the folder "src". Src is short for source – your main code source folder. Here, rename the "config.js.template" file to "**config.js**". You may get a message about changing the file type. Say "ok", change file type from .template to .js.

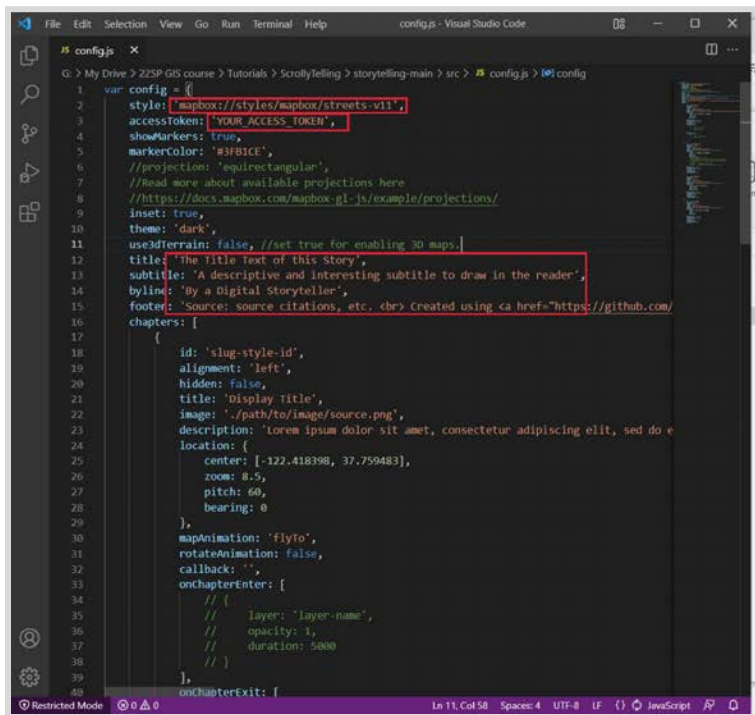


3b From the src folder, open **both the "config.js" and "index.html" files in Visual Studio**. You will only edit the config.js file, but you'll need to index.html to preview the results in your web browser as you go. *Note that until you add in your own access token and the url of your base map, the index file will not show up correctly in your web browser.* As you saw in the example, the .html file controls the web page display, while the config.js file, which is written in javascript rather than html, controls the back end of things (which layers load when, the text, and so on).





3e Last, update the “title”, “subtitle”, “byline”, and “footer” with your project information. You’ll find these in lines 13-16.



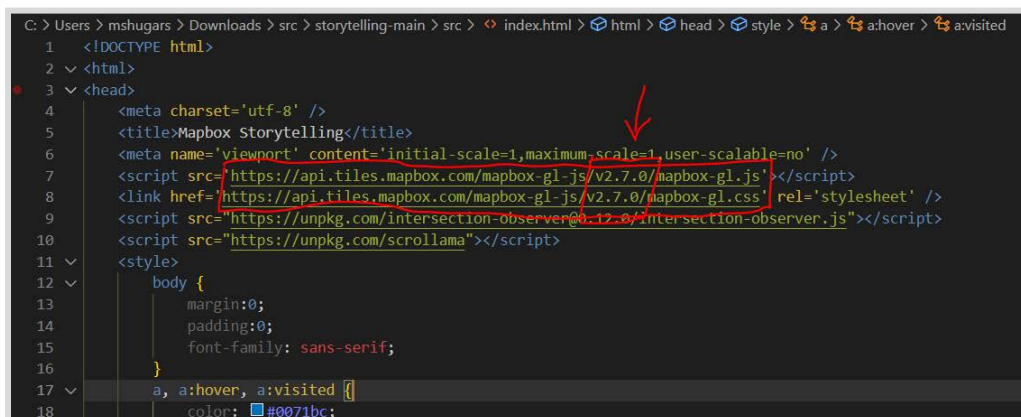
```

1  var config = {
2    style: 'mapbox://styles/mapbox/streets-v11',
3    accessToken: 'YOUR_ACCESS_TOKEN',
4    showMarkers: true,
5    markerColor: '#3FB365',
6    //projection: 'equidistant',
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. <hr> Created using <a href="https://github.com/'
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 e
24       location: {
25         center: [-122.418398, 37.759483],
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: f

```

Check that the code still works and looks right by saving the file and reloading the map web page in Chrome.

**NOTE:** If your map is not showing up when you try to load it, make this slight edit to your *index.html* file: Change v2.7.0 (shown below) to v2.0.0 in BOTH lines (script and link). Resave your index file, and try to reload your map.



```

1  <!DOCTYPE html>
2  <html>
3  <head>
4    <meta charset='utf-8' />
5    <title>Mapbox Storytelling</title>
6    <meta name='viewport' content='initial-scale=1,maximum-scale=1,user-scalable=no' />
7    <script src='https://api.tiles.mapbox.com/mapbox-gl-js/v2.7.0/mapbox-gl.js'></script>
8    <link href='https://api.tiles.mapbox.com/mapbox-gl-js/v2.7.0/mapbox-gl.css' rel='stylesheet' />
9    <script src='https://unpkg.com/intersection-observer@0.12.0/intersection-observer.js'></script>
10   <script src='https://unpkg.com/scrollama'></script>
11   <style>
12     body {
13       margin:0;
14       padding:0;
15       font-family: sans-serif;
16     }
17     a, a:hover, a:visited {
18       color: #0071bc;

```

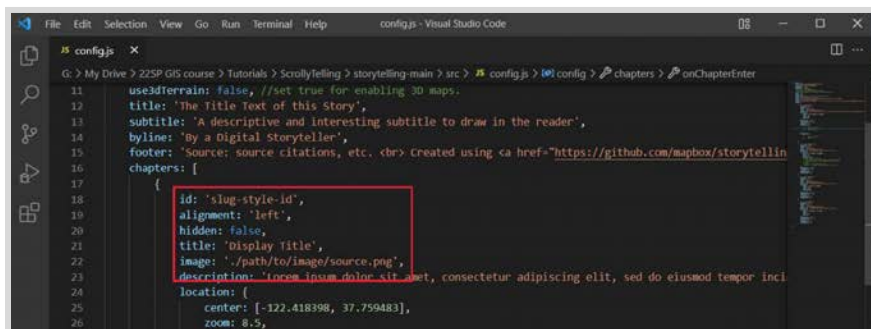
#### Step 4: Update the template chapters.

In the previous tutorial, you uploaded several layers at different geographic levels (state and city) to Mapbox. You'll now practice turning these layers on and off in the Mapbox scrollytelling template. For this, we'll look at the next section of the config.js file: the chapters.

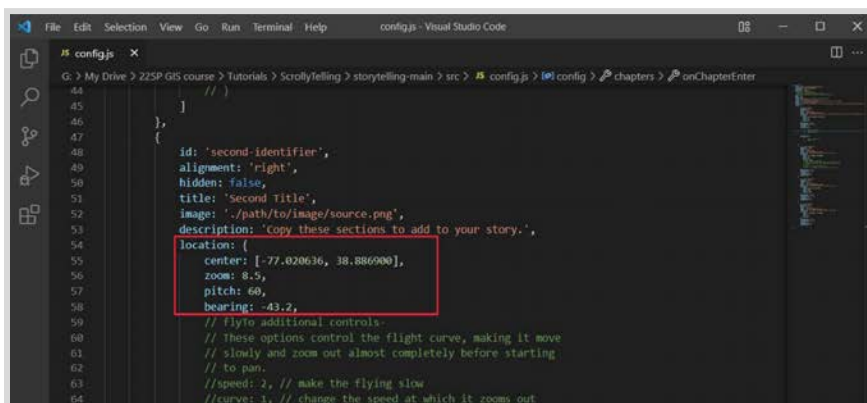
In the example, you saw that each chapter corresponded to a text box change and background map change in the scrollymap. You'll see that the config.js template file includes 4 default chapters, though you will add more for your final project. The default chapters 1 and 2 walk us through several of the chapter features, so we'll start with a closer look at these two. 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).

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 on the page; “hidden” – can you see the text box; “title” – what is the title of your chapter; “image” – chapter image, if used; “description” – chapter text content. That's all pretty straightforward text editing. For more text editing options, like font, size, and color, check out the index.html file and the template documentation.



After that, we see the section that geographically locates the chapter, “location”. You can see the screen will center on lon/lat. Here you can also control each chapter’s zoom level, pitch, and bearing (discussed below).



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.

4a You’ll start by **modifying the four default chapters in the template**. For each chapter, turn on one of the layers you uploaded to Mapbox. Make sure to set the opacity to “0” again “on chapter exit”. **Note that your layer names must exactly match the layer names on Mapbox.** You’ll also need to set the center, zoom, pitch, and bearing for each layer (see the next step).

Chapter 1: Virginia state racial change layer (NW Change)

Chapter 2: City level census layer

Chapter 3: City level vector layer

Chapter 4: One other city level layer (eg point or raster data)

```

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',

```

CHECK THAT YOU HAVE THE RIGHT OPEN AND CLOSE BRACKETS {} AND A COMMA , AFTER EACH LINE, EXCEPT THE LAST

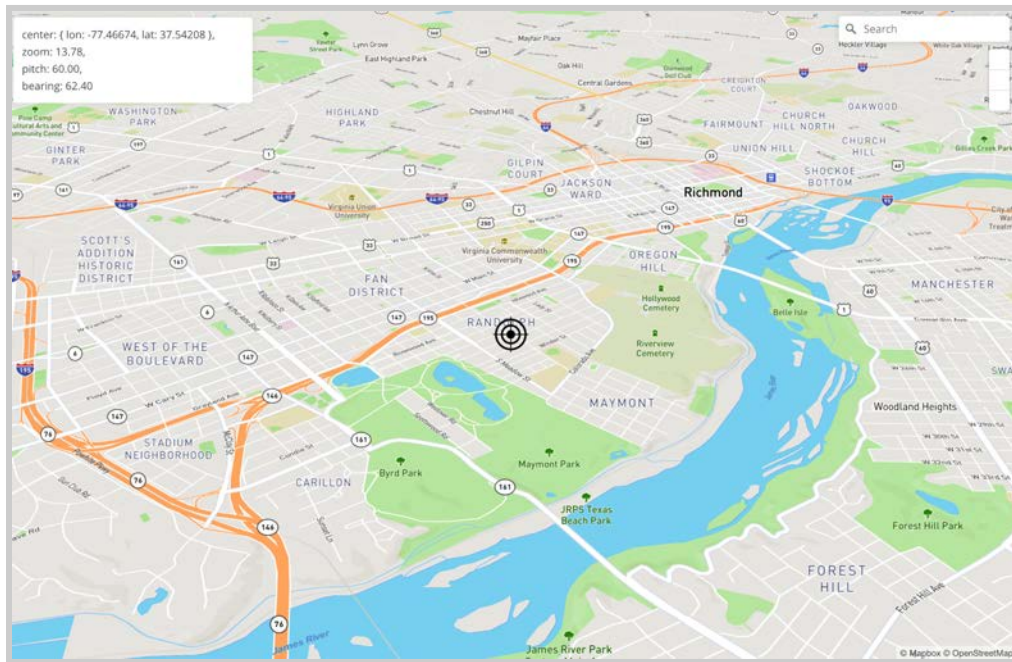
```

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

```

LEAVE THIS EMPTY IF YOU DON'T WANT TO ADD AN IMAGE

4b To calculate each chapter's Center, Zoom, and Pitch and Bearing (angle of the camera and direction of view) use Mapbox's location helper. After setting the view you want for each chapter, copy the numbers from the upper left. Use right click to tilt the map: <https://demos.mapbox.com/location-helper/>



4c For the “image” of each chapter, add a legend for each corresponding map. This will be a .png or .jpg exported from QGIS print layout, Photoshop, or any other image editing software. Make sure the colors in the legend match the colors in your Mapbox map style. I strongly recommend using **gradients** for multi-color variables like the census variables, since these are much cleaner than the blocks used by default in QGIS.

All images you use for this final scrollymap should be stored in the “images” folder within the “src” folder. Then, your file path will be “images/filename.jpg”. You don’t need to include “src” or any of the other folders that contain images, because the config file and the images folder are both in the same folder. Think of it as a local address that doesn’t require the zip code, or a local phone number that doesn’t include the area code. Your config.js code line might look like:

```
image: 'images/yourimagename.png',
```

4d Lastly, add placeholder text for the titles and descriptions of your chapters. You might pull some of this from your Map Assignment 2 text.

4e Once you have this set up, open your index.html file in your web browser to preview the results.



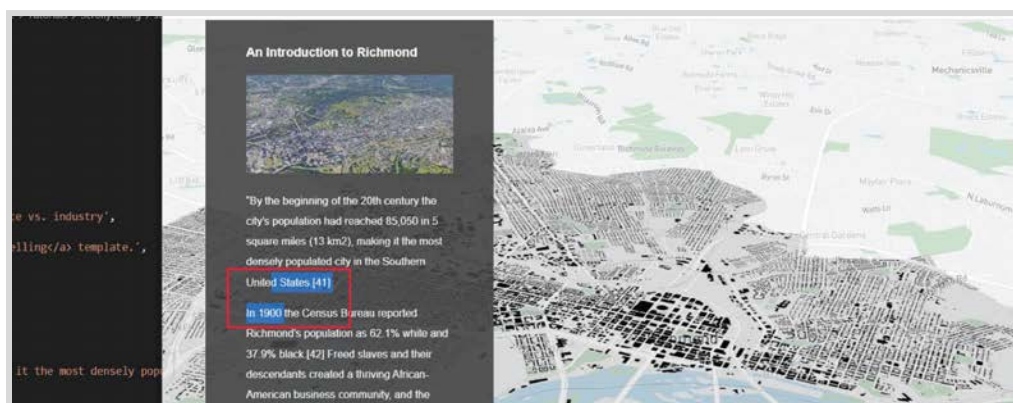
### Step 5: Try additional template features like text edits, images, videos, and chapter alignment.

To edit things like font, color, add paragraph breaks, add additional images, and even embed videos or graphics into your chapter text blocks, you'll use simple HTML tags (text between two carrots < >) within the chapter description text. First off, you'll add some paragraph breaks and selectively edit the font and color.

Note: for extensive text changes, create paragraph styles in the <style> section of the main index.html file. Check out this [W3School page](#) for more info.

5a To create a paragraph break in your chapter description text, add </p> to your text:

```
8 km2), making it the most densely populated city in the Southern United States.[41] </p> In 1900 the Censu
```



5b To change the color or font weight locally, you can use a similar paragraph style tag, with **semicolons** between each adjustment, for example: <p style="color: blue; font-weight: eg, bold; font: serif">, where "blue" and "serif" can be any basic color or hex code, and any basic font or font type.

```
147
148 picture-in-picture" allowfullscreen></iframe></p><p style="color: red; font-weight: bold">Youtub
149
```



5c You can add additional images into your chapters with tags. To add an image into your chapter description text, first make a paragraph break with </p>. Then add an image with the <img> tag: <img src = "your image url here">, as shown below. Again, this simple file path works if you save all images in the "images" folder within the "src" folder.

```
2
3 ely populated city in the Southern United States.[41] </p>  "In 1900 the
4
5
```

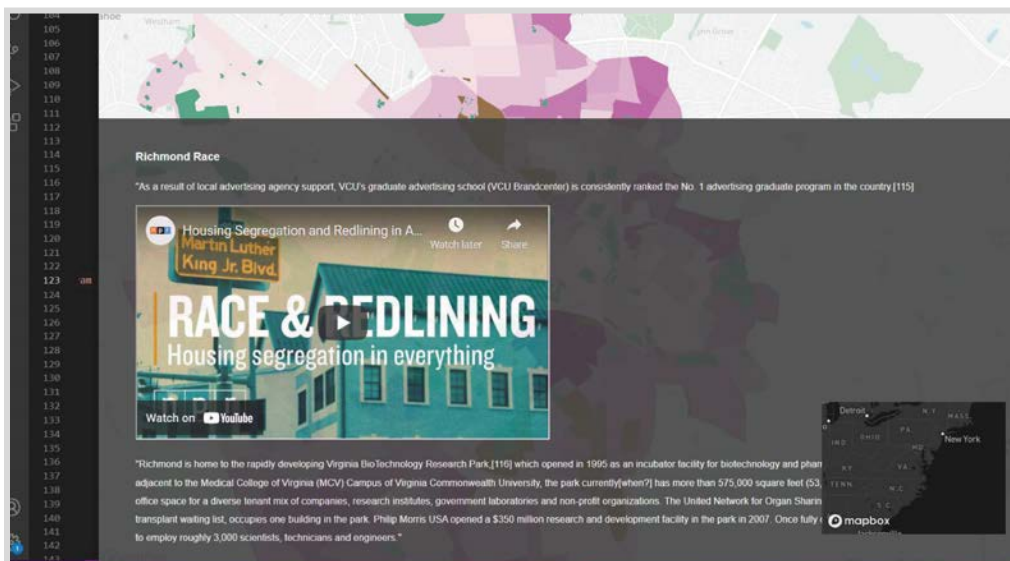
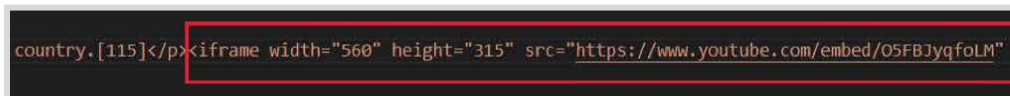


5d You can also add videos into your chapters with tags. This takes some troubleshooting to look right.

First, embed a video with the `<iframe>` tag. Iframes are windows that show another web page inside them; you can use these to show videos hosted elsewhere online, such as YouTube. Often these video hosting sites provide small snippets of “embed” code that you can copy and paste with a few grammatical modifications.

The iframe video tag has a few components: `<iframe width="560" height="315" src="videourlhere"></iframe>`. Notice that it uses an opening tag `<iframe>`, and also a closing tag which has a backslash, `</iframe>`. Notice also that you can specify the width and height in pixels of your video frame.

Make sure to also include a paragraph break `</p>` before AND after the iframe tags: `</p><iframe width="560" height="315" src="videourlhere"></iframe></p>`



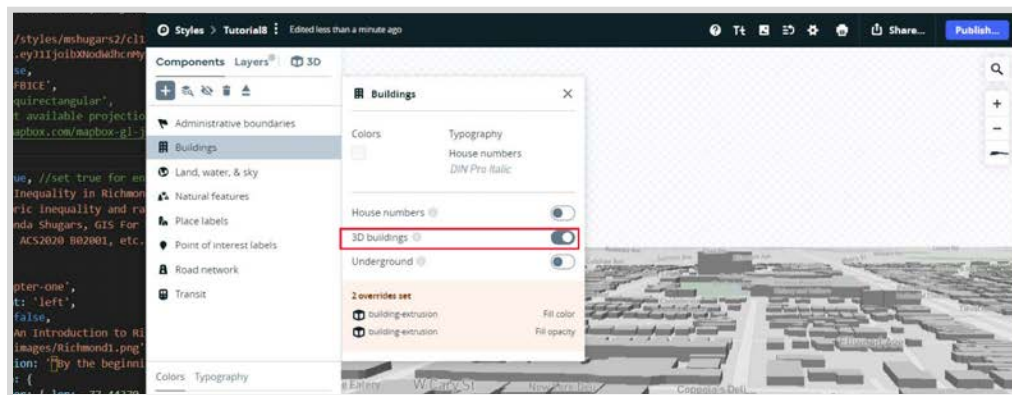
5e Add a full-span chapter. For emphasis, make one of your chapters span the full page by changing the ‘alignment’ of one chapter to ‘full’:

```
id: 'chapter-three-two',  
alignment: 'full',  
hidden: false,  
title: 'Richmond Race',
```

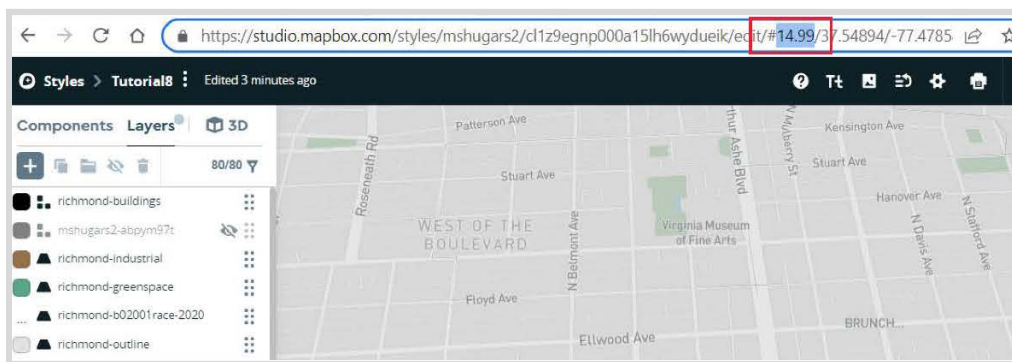
5f Add in a chapter with visibility turned off, so you can move the map along without a text box showing up. Sometimes you want to move through multiple maps without blocking each one with text. To do this, Adjust a chapter's **"hidden"** to be **"true"**.

## Step 6: add 3D buildings

We'll jump back to Mapbox for this one: turn on the 3D buildings and try rotating around them in your map. In your Mapbox style, under the “Components” tag, click on “Buildings” and toggle “3D buildings” on:



Make sure that the chapter in which you display them has a Zoom level **over 15**. You can see that they will disappear at a Zoom level lower than 15 (lower = more zoomed out):



Try adding these to a chapter where you've set the “rotateAnimation” to “true”, so that the camera slowly rotates around the buildings.

**Step 7: 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 2 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?)



**Step 8: Screenshot map and cartoon panels**

For the Tutorial Assignment submission, include:

1. a legible image of your cartoon panels
2. screenshots of your scrollymap including **at least** one image of **each of your 4 distinct chapters**.
  - a. make sure your screenshots also show all the components of steps 4-6: at least one embedded image, video, full span chapter, no show chapter, 3D buildings, text of a different color, and so on.
3. screenshots of your code file with each of the areas that have been edited. Highlight or outline the edits.

- no Bonus -