Anna Wegbreit

May 4, 2018

Web Mapping Final Project

**Code Workflow and Outline of Instructions:**

The first step to creating this web map was Data Collection and manipulation. For these GeoJSON files, manipulation meant re-naming the files to end in .js and opening them in the computer’s basic text notepad application. I brought them into notepad because opening multiple large GeoJSON files within Brackets took an infuriatingly long time. I assigned variables to each GeoJSON file as a way to quickly reference them throughout the code. The notepad trick was a sanity-saving trick I learned during this process.

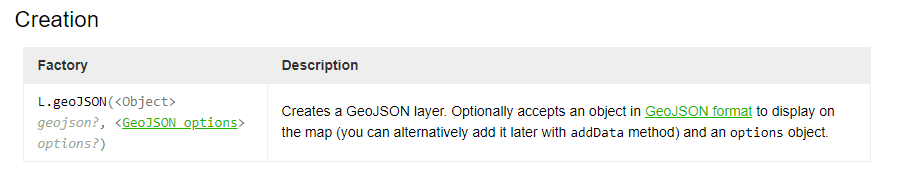
Lastly, before starting on the map content, I made sure to go over the metadata to review the properties and attributes within each file. Going over the metadata is not just best-practices for GIS data, but it was essential to see how the property names were formatted so I could refer to specific properties when filtering for specific GSI tools later.

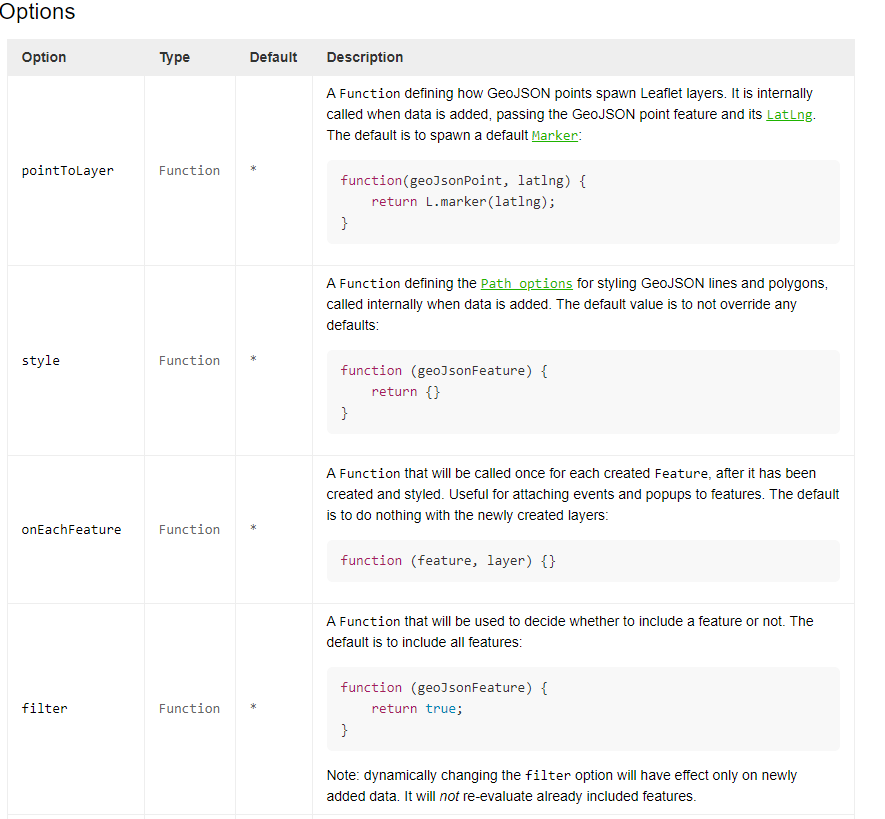
Instructions:

1. Create new <script> tags within the HTML so that the script can locate the GeoJSON files being used. Each file has already been assigned a variable.

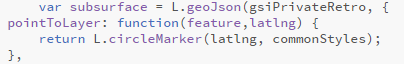


1. Create a GeoJSON layer with L.geoJSON and bring in the GeoJSON code as an object. There are several important Options to include when creating the GeoJSON layer including pointToLayer, style, onEachFeature, and filter.

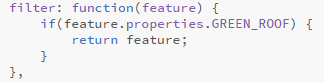




1. Use the pointToLayer function as the first option in the L.geoJson layer. This function allows the individual points to “spawn” as Leaflet layers. When points data is called it passes the points and their LatLng locations. For point data the default output is a Marker or a Circle Marker.



1. The Filter function is a critical option to include when creating layers to represent different classifications of GeoJSON data. Without the filter function every point feature in the GeoJSON would be represented identically, and the map would not communicate anything other than the features’ loction. The filter function works best with a conditional logic statement. For each layer I used the filter function and an “if statement” to separate the categories of GSI types from their properties.



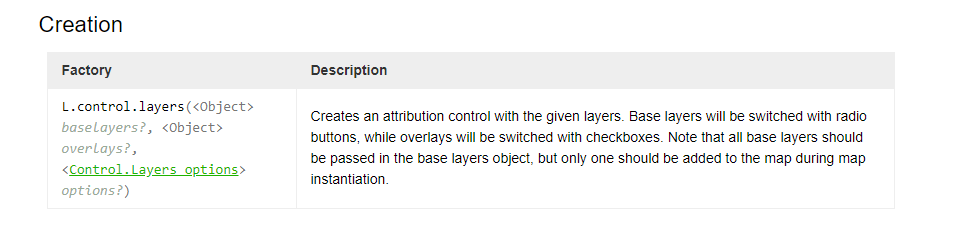
1. Use the Style function to style the appearance of GeoJSON points, lines and polygons. This function does not have a set default value other than to not override any other intentional or default styles. However, this function sets the Path options for the object as it is called internally throughout the script. In this lab I used the style function to create visually appealing circle markers to represent GSI implementations in the city. Instead of setting the radius to a uniform value, I called the getRadius function and used the count of the specific GSI tool featured in the layer as a multiplier value for the size of the marker. Styling the radius of each circle marker this way is what makes this map a proportional symbol map.



1. The onEachFeature option is the last option I included in the L.geoJSON layer. OnEachFeature is a function that is called for each feature. In this map I used it to attach an event and later to create Popups for each feature.
   1. In this map when the mouse moves over a feature the feature changes in appearance until the mouse moves off of the feature. These mouse events are made possible by the .on method which creates an event listener function to listen for a ‘mouseover’ or ‘mouseoff’ event to happen. When the first event happens (“mouseover”) the feature’s appearance changes thanks to the .setStyle method which calls a Style function for the feature while the mouse remains over it.
   2. The .on method is used again to listen for the “mouseout” event, to call the setStyle function again to return feature back to its original style. If I had more time to refine this map I would employ a more liberal use of commonStyles to help keep the overall styling consistent, as well as keep the script more compact!



* 1. In the watersheds layer I included an additional method in the onEachFeature option. My goal was to identify which watershed each GSI project was located in, so I used similar “mouseover” events to highlight the particular watershed, and also used the .bindPopup method to attach an instantly available label for the watershed.

1. To create a clickable legend for this map I used L.control.layers. To hold all of the layers in the control option I created a variable “sourceLayers” to be included as the overlay object that gets passed through. I had several frustrations with the control layer because I would like to be able to nest groups of layers into a single layer that could be clicked on and off all at once, in addition to choosing the individual layers to make visible. 
   1. The way I worked around the lack of nested source layers was to create layer groups for each dataset. You will notice that in the creation of each GeoJSON layer instead of .addTo(map) I added the layer to three different layer groups. If I were to continue this project my goal would be to create a better work around to nesting layers within the control.
   2. The control layer was also the location where I added a scale bar for reference.



**Reflection:**

My general reflection for this course is that I am extremely glad I took the class; I learned so much in such a short amount of time.

Coming into this class I had a lot of interest in learning to code in Javascript because I knew it was one of the more popular languages and coding looks good on a resume. I had tried Codecademy before but did not take it super seriously and would get stalled over little things.

Overall I realized that creating a webmap, or creating any type of code, is almost like creating art in the sense that you can really tell how many hours go into it and there are no shortcuts—especially when you are still learning how to code. Although there are many great online resources I was able to eventually utilize, at first I struggled to use those resources because I was uncomfortable and unfamiliar with the language and how to read about reading code. Not only did I learn a lot in this course but I learned enough to feel much more confident to go onto Stackexchange or Geonet or any other online discussion board and be able to understand at least some of what is being discussed.

The part of the course I struggled with the most was around the mid-point of the semester, especially on the midterm problem set. While I felt comfortable in the beginning having a ‘template” to put my own variable names and values into, I really struggled when it became time to start creating code without an actual example in front of me. The strategies that helped me the most were having the videos to walk through the new concepts (being able to go at my own pace with the tutorials was much easier to follow than if we had gone through all the new concepts in class like that). And being able to use my code from previous weeks and other student’s Show and Tell examples as inspiration or even just to compare when checking for errors. A personal strategy I found helpful was looking at the different parts of the code in chunks, as opposed to looking at it like a set of lines with variables, operators, properties, methods, loops, functions, callers, handlers and other elements we discussed. By being able to see one “chunk” of the code as the part that filtered for values in this range, or the part that changed the popups, or as the part that was affected when I clicked on it, I was able to focus on that section and play around with it until I grasped what it was doing, even if I was still unsure of the exact term used to describe it. While that may not be best practices of coding, it helped me get through the labs I struggled with the most.

I am already looking forward to using this type of web mapping in my GIS career. Once I feel comfortable enough with hosting and Github I would like to create my own portfolio website to update with projects I create throughout the rest of the PSM. I also look forward to using elements of web mapping at work, where I am a GIS Technician. Currently when I want to make a web map at work I use ArcGIS Online. Although the Arc Online platform is easier to use than lines of script, it was great to be able to make that connection of what was going on in both web maps when thinking through a problem. I also have realized that Arc Online has a quite a few limitations as far as its capabilities and interface. Having Javascript as an additional tool I can use for web mapping will be helpful when I need more freedom to customize the appearance and abilities of a client’s web map.

If there were a Web Mapping 2 class offered I would be interested in learning about making a personal online portfolio and customizing it to best showcase my GIS capabilities and past projects. Also, for students who do not already have a job in GIS (or are looking to move up in their GIS career) having a personal website to show potential employers is a huge benefit for job seekers. I would also be interested in learning the basics of adding more analysis elements to the map for the user to access. For example, how to add a search bar function or how to let the user turn filters on or off.

In the end I spent a lot of hour on this course but I am so glad I did because I gained so much from it and I truly feel that my time investment paid off. Thanks again!