Javascript Libraries for Web Mapping: Cartographic Experiments and Selected Review

Maps are ubiquitous on the internet. From navigational tools like Google Maps to infographics produced by news outlets like the New York Times, the art and science of creating web applications with maps as interfaces has flourished and impacts our daily lives. A wide range of tools are available for bringing geospatial data to life on the internet. Some require little to no training to display, edit and share map features, such as ArcGIS StoryMaps and Google MyMaps. These are limited in terms of functionality and customization. On the other hand, professionals use some very abstract tools to create custom infographics and other sophisticated applications from scratch, requiring specialized training. In the middle are numerous libraries catering to the growing populations of web developers interested in cartography and cartographers with some web development background.

To the student of Thematic Cartography interested in web mapping, it is important to assess the available tools not only in terms of their usefulness for producing a map, but also for producing a well-designed map. There is often a tradeoff between ease of use and flexibility: how much the library handles “under the hood” versus what level of customization it makes possible without getting in the way.

Javascript is “the programming language of the Web,” running in the browser on every website but the most basic. We can safely say that most if not all interactive maps found on the internet rely on it to some degree. A number of Javascript libraries make it practical for the aspiring cartographer with an intermediate level of familiarity with the language (like myself) to readily produce interactive maps from raw geospatial data. I investigated several such libraries, with a focus on cartographic design, appropriate map elements, and simple, logical code. To assess the capacity of each to produce a well-designed map, I attempted to replicate as closely as possible the proportional symbol map of Florida’s counties that won our class’s map design competition using several libraries, only adding very basic interactivity. Within reasonable time constraints for this project, I had varying degrees of success. Some substantial differences emerged.

Four JS libraries stand out from my experience and provisional research: Leaflet, Open Layers, D3 and the ArcGIS Javascript API. Articles and discussions available online generally compare them from the perspective of a software developer, rather than a cartographer. **Leaflet** a popular and widely used tool for adding an interactive, zoomable map to a webpage. It is admired for being “lightweight” and simple, containing “all the features most developers need” but not addressing more advanced uses. It is very easy to get a basic map up and running. Because it is so popular, it is easy to look up help in online forums. My experiment with Leaflet was largely successful, and informed by some previous experience with it.

**Open Layers** is often compared to Leaflet. Similar in purpose and overall design, it is considered more powerful but heftier and harder to learn, more suitable “for complex GIS application development.” It is far less popular than Leaflet but highly regarded among a more specialized community. There is a lively Leaflet versus Open Layers debate online. Open Layers was new to me, but I was also successful with it.

**D3** (Data Driven Documents) is a data visualization library with broad applications, geographic and otherwise. It is the professional standard for custom infographics, and the subject of many how-to books. It includes tools for rendering geographic information, but not much built-in functionality related to maps. Many highly skilled developers use it to produce incredibly sophisticated geovisualizations that would be clumsy or impractical with Leaflet or Open Layers, especially animations. As I found out the hard way, it is hard to do much beyond a very simple map without a solid grounding in general D3 concepts. My efforts were only partly successful.

The **ArcGIS Javascript API** is organized along the lines of Leaflet or Open Layers, but it is more feature rich. It makes it easy to create a web map and apply some of the same powerful data visualization features one would find in ArcGIS Pro, but it seems less thoroughly customizable, trading design flexibility for ease of use and an ESRI look and feel. It is optimized to use data from ArcGIS Online, and integrate with various ESRI products. I did not have time to produce a map using the ArcGIS Javascript API, and my knowledge of it comes from reading the documentation rather than getting my hands dirty.

Cite SotryMaps, Google MyMaps,

<https://www.w3schools.com/Js/>

<https://www.geoapify.com/leaflet-vs-openlayers>

<https://stackshare.io/stackups/leaflet-vs-openlayers>

https://gis.stackexchange.com/questions/33918/choosing-openlayers-or-leaflet

https://www.d3indepth.com/geographic/

At the most basic level, some allow a user, without GIS training, to display, customize, and share geospatial data, including ESRI StoryMap and Google MyMaps (citation). On the other hand, there are

There are many tools ava

Javascript is “the language of the web” (citation). All modern web applications use it for client side scripting, and all modern browsers run it. Any interactive map one may find on a website makes uses of it.