[PD-05-032] JavaScript for GIS

Abstract

JavaScript (which has no connection to the Java computer language) is a popular high-level programming languages used to develop user interfaces in web pages. The principle goal of using JavaScript for programming web and mobile GIS applications is to build front-end applications that make use of spatial data and GIS principles, and in many cases, have embedded, interactive maps. It is considered much easier to program than Java or C languages for adding automation, animation, and interactivity into web pages and applications. JavaScript uses the leading browsers as runtime environments (RTE) and thus benefits from rapid and continuously evolving browser support for all web and mobile applications.

Keywords: application, development, JavaScript, mobile, programming, script, web

Author & citation

Swift, J. (2020). JavaScript for GIS. The Geographic Information Science & Technology Body of Knowledge (3rd Quarter 2020 Edition), John P. Wilson (ed.).
DOI: 10.22224/gistbok/2020.3.5.

Explanation

1. Definitions
2. What is JavaScript
3. JavaScript for GIS Application Development
4. Future Trends in Geospatial Programming with JavaScript

1. Definitions

client-side (adjective): when an action takes place in a client-server computer network, such as running a computer program on a client's computer

client-side (noun): the end user's local computer rather than a remote server

cascading style sheets (CSS): a document with a .css extension that contains markup language for adding colors, spacing, fonts, etc. (style) to web documents such as HTML5 web pages

codebase: source code written by humans, including markup language, scripting language, or programming, used to construct, or build, a given software component, application, or entire system

front-end (noun): the user interface that a web or mobile application user directly interacts with, such as a web page or graphical user interface, and may also be referred to as the client-side of the application

front-end (verb): front-end web development typically utilizes HTML5, CSS, and JavaScript to integrate information into a graphical user interface so end users can work with the data

Git: a free version control system or software tool for managing the contents of a code repository

just-in-time compilation (JIT): dynamic processing of data after a computer program has started execution

library: a collection of shared programming code, such as files, programs, routines scripts or functions that can be used to facilitate the development of other software programs and applications (Swift and Goldberg 2019)

model–view–controller (MVC): common software design pattern for creating user interfaces that separates internal application data from information that the user sees< runtime: the timeframe when a computer programming is running, beginning when the program is launched and ending when it is closes

markup language: a human-readable computer language that uses tags written as text to annotate or define items in a document file, meaning it can contain standard words rather than programming syntax

scripting language: an interpreted computer programming language that is translated into machine code at the time it is run, executed to accomplish specific tasks without being compiled

script: a series of computer programming commands that can be saved in a computer file to facilitate automation of execution of tasks, or that could be executed manually one-by-one in a terminal program or emulator

web application framework: a software framework such as ASP.NET or Ruby on Rails, created to help developers build web applications such as web API's and web services and often providing libraries for simplifying programming tasks such as accessing and configuring data

2. What is JavaScript?

JavaScript is one of the most popular high-level programming languages used to develop user interfaces in web pages (Stackoverflow 2020). The principle goal of using JavaScript for programming web and mobile GIS applications is to build front-end applications that make use of spatial data and GIS principles, and in many cases, have embedded, interactive maps (Huang 2019; Quinn 2018; Latif et al. 2016). JavaScript has no connection to the Java language. It is considered much easier to program than Java or C languages for adding automation, animation, and interactivity into web pages and applications. JavaScript uses the leading browsers as runtime environments (RTE) and thus benefits from rapid and continuously evolving browser support for all web and mobile applications. JavaScript applications can be readily optimized for different screen sizes, mobile devices, and user scenarios. A front-end developer must ensure that web pages and web applications are responsive and perform optimally, meaning that they appear correctly to the user and

operate as expected (Huber and Demetz 2019; Ricker and Roth 2018; Roth 2017).

JavaScript conforms to the ECMAScript specification, meaning that it follows a general-purpose programming standard accepted world-wide (ECMAScript 2020). Though JavaScript is sometimes called ECMAScript, this was the parent standard Brandon Eich used in 1995 to create this language for Netscape Navigator (Severence 2012). Eich's main goal was to create a simple language for creating dynamic web pages. JavaScript's original code name was Mocha, was launched as LiveScript, then renamed JavaScript soon after. Microsoft released their own, similar version called Jscript, also based on the ECMAScript standard, to avoid trademark conflict.

JavaScript became a mainstream language when Asynchronous JavaScript and XML (Extensible Markup Language or AJAX) was developed in 2005 by Jesse James Garett (Garrett 2005). AJAX is a combination of web technologies that can be used to build applications that can quickly update a user interface without reloading an entire web page or application (Huang 2019; MDN 2020; Roth 2017). In 2009, the world's leading browsers adopted HTML5 (Hypertext Markup Language Version 5) as the core language that defines properties and behaviors of the World Wide Web. The fact that JavaScript works directly with HTML5 played a significant role in it becoming a leading computer language for the web used world-wide.

Over the last decade, there has been an explosion of publicly available web and mobile GIS applications that use open source as well as proprietary JavaScript application programming interfaces (APIs) and libraries. JavaScript APIs are collections of library modules that provide GIS software functionalities through programming (Chow and Yuan 2019). Since JavaScript is now an essential language for web and mobile application development, this entry provides an overview of commonly used programming API's and libraries that support its use. JavaScript usage in web GIS programming is also discussed in Swift and Goldberg (2019), and in mobile GIS programming in Huang (2019). The conclusion of this entry highlights several anticipated future trends and challenges in geospatial application development using JavaScript.

3. JavaScript for GIS Application Development

JavaScript gradually became popular among the GIS application development community due to standardization across browsers and operating systems, the open development community, and ease of> integration with other web scripting languages to build interactive web and mobile geospatial applications (Huang 2019; Latif et al. 2016). Many leading geospatial software companies are heavy users of JavaScript in their front-end web and mobile mapping application offerings, including Esri and Google. Open source and proprietary GIS software often provide JavaScript application programming APIs that allow developers to easily access spatial and non-spatial data (Chow and Yuan 2019; Huang 2019; Sack 2017). Also, the use of JavaScript facilitates modular integration of advanced geospatial analysis functionality and application user interface customization. Commercial vendors promote such APIs as a marketing tool as well as a way to crowdsource growth in software functionality.

3.1 Architecture

JavaScript is a client-side, object-oriented scripting or front-end programming solution that works directly with HTML5 and CSS (Cascading Style Sheets) in a browser to deliver an application to a web page. Modern JavaScript is often just-in-time compiled (JIT) into native machine code in all major JavaScript implementations (Aycock 2003). Client-side web GIS programming is explained in Sack (2017), Quinn (2018), and Swift and Goldberg (2019), including how a browser typically runs software applications. Concerning JavaScript for GIS application development, the client is usually a browser or mobile application where most tasks performed by JavaScript do not require access to a web server. If running JavaScript on a server (server-side) is desired, the JavaScript runtime platform Node.js, for example, enables JavaScript to be deployed (Brown 2020).

Commonly in web page development, JavaScript code is written between <script> and </script> tags in an HTML5 page, which identifies a client-side script (program code). The code between these tags contains the scripting statements in JavaScript or points to an external JavaScript code file with the extension .js. When a client-side JavaScript program, or script, executes, a user only sees the resulting application built with JavaScript in a web page or mobile application, not the script. Quinn (2018) describes Web GIS architectures in detail, and Sack (2017) explains GIS client applications.

Figure 1 illustrates the basic components of a typical JavaScript for GIS application development project. In general, an application may consist of an HTML5 document, with added JavaScript and link tags, which load a GIS-based JavaScript API code to create an instance of a web map. The HTML5 document may also include code or script for CSS specific to GIS style components and perhaps widgets. The JavaScript and CSS code inside the HTML5 document state the version of each language being used and contain links to separate JavaScript and CSS files. These individual files contain all of the instructions for making GIS components such as maps in the application, and styling the way application components are rendered or displayed, respectively.

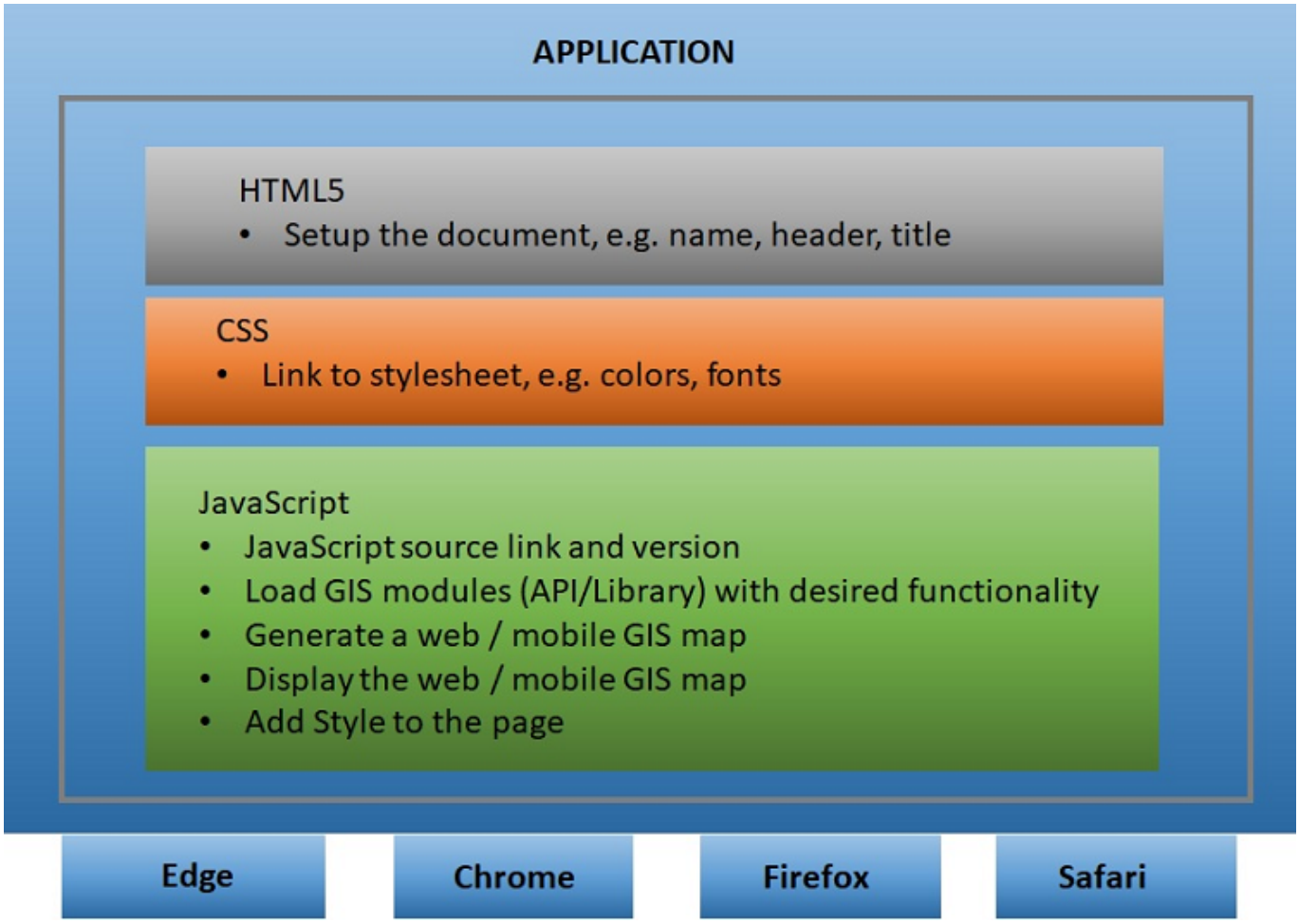


Figure 1. Example main components of a JavaScript for GIS scripting (development) environment. Source: author.

In a standard web application, a web GIS developer's job is to create a web page that has a header, fonts, colors, pictures, perhaps animations, and an embedded map the end-user can easily interact with> (Quinn 2018; Roth 2017; Sack 2017). In terms of application architecture, the developer must break the idea for the application into small components, then translate these pieces into instructions that a computer understands. In a web GIS application, HTML5 provides the basic structure of the web page, CSS controls the formatting of the components, and JavaScript guides the behavior of all of the different elements inside the web page. Typical uses of JavaScript functionality include popup boxes on your screen, interactive games, animations, and security features. Interactive maps are embedded inside web pages by writing JavaScript functions specific to each particular GIS software API or library, within the HTML5 code. In contrast, map images only require and HTML <img> tag.

There is strong support in the GIS application development communities toward the use of both proprietary and open-source JavaScript libraries as well as APIs. Numerous open-source JavaScript libraries are publically available and consist of compilations of code that can be used by application developers to simplify writing programs. Many JavaScript libraries have been developed specifically for designing and building interactive user interfaces across platforms and devices. One of the most popular is ReactJS, a front-end

JavaScript Library for building web and mobile user interfaces (Facebook 2020). React was designed by Jordan Walke to support building responsive and scalable applications quickly. It handles the GIS data layers that are displayed, and can also be used to create modular (reusable) user interface components. Developers can easily create sophisticated applications that allow the data on the web maps to change without reloading an entire web page.

The most commonly used development platform for hosting JavaScript for GIS APIs and libraries, and applications created from these by developer communities, is GitHub at github.com. GitHub is a website that provides cloud-based hosting for Git repositories. Github was initially designed as a hosting service, and also provides command-line tools, a web-based interface for managing code, software version control, and standardized access and collaboration functionality such as wikis. GitHub offers free accounts for open source projects, as well as commercial, professional, and enterprise accounts that support privacy and other features that support businesses. Accompanying code development documentation and data can also be managed in GitHub, though it is not as common for handling binary data (e.g., images). GitHub facilitates new programmers' participation, lowering barriers for open source developers, and fostering social interaction through user-profiles and peer interaction through repository activities. Users who want to contribute to an existing repository or use it in their development can "fork" the project repository, meaning obtain a copy and create a new repository in an individual account. Users can "clone" their forked repository if they want a copy on a local computer to modify it. Using Git commands, a local copy can be "committed" and then "pushed" back into a forked repository in their account. Many major GIS software development communities have joined GitHub, including Microsoft, Esri, QGIS, Mapbox, OpenGeo, and GeoServer.

Chow and Yuan (2019, Tables 1 and 2) provide lists of GIS-related API's related to JavaScript programming for GIS, and Sack (2017, Table 1) includes map data formats. Table 1 provides a selection of the most common JavaScript APIs and libraries used in GIS programming by GI Scientists and geospatial application developers. Some can also be used for different purposes, such as Angular, which is a general web application framework, while Leaflet is a web mapping library (Huang 2019; Ricker and Roth 2018; Roth 2017; Swift and Goldberg 2019). Some offer both open source components and commercial access to additional functionality and often, customer support services (O/P in Table 1). This summary is a starting point for beginning JavaScript for GIS developers. Chow and Yuan (2019) provide a detailed description of commonly used, platform-specific software development kits (SDKs) designed for programming mobile GIS applications.

Table 1. Selected JavaScript APIs and Libraries Typically Used in Web and Mobile GIS Applications

Application / Library	Open Source / Proprietary (O / P)	Link
Angular	O	angular.io
Apache Cordova	O	cordova.apache.org
Carto.js, Carto VL (Carto/CartoDB)	O / P	carto.com

Application / Library	Open Source / Proprietary (O / P)	Link
Cesium	O	cesium.com/cesiumjs
D3 (Data Driven Documents)	O	d3js.org
Electron	O	www.electronjs.org
Esri API for JavaScript	O / P	developers.arcgis.com/javascript
Express	O	expressjs.com
Google Maps API	O / P	developers.google.com/maps/documentation
Ionic2	O	ionicframework.com
jQuery API	O	api.jquery.com
Leaflet	O	leafletjs.com
MapBox	O / P	mapbox.com
MapQuest.js	O	business.mapquest.com/products/mapguest-js
Node.js	O	nodejs.org
OpenLayers	O	openlayers.org
OpenWeather Map API	O / P	openweathermap.org
ReactJS / Native (JavaScript Diaries)	O	react.js
Sencha	O	www.sencha.com
Turf.js	O	turfjs.org
Vue.js	O	vuejs.org

3.2 Use Cases

Novel uses of JavaScript for collecting, organizing, and preparing geospatial data are currently being implemented in open source communities as well as by proprietary software vendors. For example, the OpenStreetmap initiative represents a truly open, global effort to create a map or the entire earth based on crowdsourced data. In 2004 this project was started by Steve Coast (Coast 2013), and today includes millions of users and collaborators, such as Craigslist (Farivar 2012). OpenStreetMap data are added as tiles to a web application by referencing a representational state transfer (REST) Uniform Resource Locator (URL), explained in Chow and Yuan (2019) and Swift and Goldberg (2019). JavaScript libraries that utilize OpenStreetMap tiled web maps include Leaflet and OpenLayers (Table 1). Esri uses JavaScript in the development of popular Web GIS toolkits such as App Studio and Web AppBuilder (Esri 2020). Developers can create applications with virtually no programming, or customize applications by starting with templates written in JavaScript, up to writing applications using JavaScript from scratch. Using these toolkits, Esri continues to invent useful tools to show what can be done with its configurable products. In 2016 Esri released Survey123, a template designed for creating surveys and polls created using App Studio (Esri 2016). Another essential example use of JavaScript libraries is in spatial analysis, such as the popular open-source library Turf.js, which can be added to a web page as a client-side plugin or run with Node.js on a server. Turf.js includes commonly used spatial operations, spatial statistics, and data classification tools. There are many open-source and proprietary JavaScript libraries that provide spatial analysis functionality that can be integrated with web and mobile GIS applications.

A basic JavaScript for GIS application might consist of an interactive web GIS map embedded in a web page or mobile application and include additional web page functionality integrated with non-spatial static or dynamic data sources. JavaScript implements the interactivity between an end-user and a web GIS map within a web or mobile application. The user interface, as well as the map, are often both customized with JavaScript (Sack 2017).

There are many approaches to create such an application. For example, a static data source could consist of a set of landmarks deployed as a GIS layer on the map that the application developer does not want to edit or be edited by others. At the same time, dynamic content could be delivered behind-the-scenes in real-time or near-real-time, such as recent earthquakes or other events or social media posts (Esri Storymaps Team 2020). The data can be stored on or streamed from a web server and a mobile client using JavaScript object notation (JSON) or extensible markup language (XML) data formats (Huang 2019). GeoJSON is a popular geospatial extension that provides a standard way of storing geospatial data in JSON. Browsers translate GeoJSON into JavaScript. Spatial features such as points, line strings, and polygons, as well as their non-spatial attributes, are represented in GeoJSON. This example application could also include routing and navigation information coded in GeoJSON. Another approach might be to customize existing, configurable web or mobile GIS application templates using JavaScript using proprietary software packages. The following section provides some guidance for JavaScript programming.

3.3 Guidance for JavaScript for GIS Programmers

JavaScript is chosen for programming web and mobile GIS applications over traditional programming languages such as Python and the C languages, primarily because of its optimization options and> extensive support built into today's browsers. Nevertheless, figuring out what the best programming language is for a given situation can be a challenging task. When choosing your programming options, the following considerations offer a project planning guide for the web or mobile GIS application developer:

1. If you already have a useful codebase to start with, you should consider using the one you have in hand and building upon it. If you don't, there may be added cost in time if you add a new language to your project, depending on the coder's experience.
2. If you need to change to JavaScript from another language, there must be clear reasons to support this decision, such as a performance requirement, i.e., you need to optimize the runtime environment.
3. You could choose JavaScript if it fits your specific use case technical requirements, a list of actions, or event steps, and in particular, if your application should run client-side.
4. You want to share your project with other developers who work with JavaScript on similar GIS projects and become part of their community.

Huang (2019), Quinn (2018), Rey (2017), Sack (2017), and Swift and Goldberg 2019) offer guidance for programmers related to choosing between client-side versus server-side system architecture, and for selecting between differnent languages depending on the programming task. For example, both Python and JavaScript support writing code using simple functions and variables without needing class definitions. Python might be chosen over JavaScript, for instance, if classes and inheritance are required. Python is often used for backend data processing as an alternative to Node.js, whereas JavaScript is usually

chosen for front-end development. In regards to Java, Chow and Yuan (2019) and Huang (2019) explain how it can be used for server-side general, as well as spatial, web, and mobile application development. C is typically used to write specialized utilities or programs that can be run on a server, rather than front-end applications that browsers support. For example, C programmers develop operating systems like Microsoft Windows and Mac kernels. In the past, C might have been chosen over JavaScript if programs needed to run extremely fast and maximize computer memory usage.

When considering how or whether or not to use Javascript, code that can be compiled into a compact program might be more desirable than an interpreted program. Compiled languages can run extremely fast because the computer processor can execute them directly, while interpreters run through a program line by line to execute every command individually. A program interpreter reads and translates a program line by line on the fly, whereas a compiler works ahead to create a new file that contains a translation for the input program specific to the computer. Nowadays, JavaScript undergoes just-in-time compilation (JIT) into code during execution. Exactly when JavaScript code is compiled can vary based on the implementation. For example, in Chrome, a JavaScript program may start out using an interpreter, then the program can be compiled into an optimized native machine code if it includes functions that are repeatedly executed (Google 2020; Gordon 2020).

4. Future trends in geospatial programming with JavaScript

JavaScript can be used to develop web pages, web applications, and mobile applications utterly independent of a GIS. As JavaScript usage rose with the internet over the last several decades, the geospatial application development community also embraced it for GIS. This is primarily due to its widespread adoption by all browsers as a web programming standard, its ease of use in GIS application development, and its free and open community. Recent advancements and demands for improved 3D, AR (augmented reality), and VR (virtual reality) geospatial applications provide significant opportunities for growth within the JavaScript for GIS development industries. Also, continued advancement in implementing high performance client-side geospatial analysis, spatial statistics, and spatial modeling are in high demand. Lastly, new and novel ways of gathering, managing, and processing big data using JavaScript and other client-side technologies is another possible frontier for web and mobile GIS applications development communities.

References

Aycock, J. (2003). A brief history of just-in-time. ACM Computing Surveys. 35(2)97-113.

Brown, E. (2020). Web Development with Node and Express Leveraging the JavaScript Stack. Second Edition. O'Reilly Media, Inc.

Chow, E. and Yuan, Y. (2019). GIS APIs. The Geographic Information Science & Technology Body of Knowledge (2nd Quarter 2019 Edition), John P. Wilson (Ed.).

Coast, S. (2013). Mapping my next adventure. Blog entry, retrieved August 16, 2020.

ECMAScript (2020). Language Specification. Retrieved on August 16, 2020.

Esri Storymaps Team (2020). Mapping the novel coronavirus pandemic. Retrieved on August 16, 2020.

Esri. (2016). Introducing Survey123 for ArcGIS. ArcGIS Blog. Retrieved August 16, 2020.

Facebook (2020). React. Facebook Open Source. Retrieved on August 16, 2020.

Farivar, C. (2012). Craigslist is on board: OpenStreetMap soars to new heights. ArsTechnica. Retrieved on August 16, 2020.

Garrett, J. J. (2005). Ajax: A New Approach to Web Applications. Retrieved on September 10, 2015.

Google. (2020). V8 JavaScrip Engine. Retrieved on August 16, 2020.

Huang, Q. (2019). Programming of Mobile GIS Applications. The Geographic Information Science & Technology Body of Knowledge (1st Quarter 2020 Edition), John P. Wilson (ed.).

Huber, S. and Demetz, L. (2019). Performance Analysis of Mobile Cross-platform Development Approaches based on Typical UI Interactions. In Proceedings of the 14th International Conference on Software Technologies (ICSOFT). 40-48. Prague, Czech Republic.

Latif M., Lakhrissi Y., Nfaoui E.H., and Es-Sbai N. (2016). Cross platform approach for mobile application development: A survey. In: 2016 International Conference on Information Technology for Organizations Development (IT4OD). Electrical and Electronics Engineers (IEEE), 1-5. March 30-April 1, 2016. Fez, Morocco.

MDN. (2020). MDN web docs: Ajax. Retrieved on August 16, 2020.

Quinn, S. (2018). Web GIS. The Geographic Information Science & Technology Body of Knowledge (1st Quarter 2018 Edition), John P. Wilson (ed).

Rey, S. J. (2017). Python for GIS. The Geographic Information Science & Technology Body of Knowledge (3rd Quarter 2017 Edition), John P. Wilson (ed).

Ricker, B., and Roth, R. E. (2018). Mobile Maps and Responsive Design. The Geographic Information Science & Technology Body of Knowledge (2nd Quarter 2018 Edition), John P. Wilson (Ed).

Roth, R. E. (2017). User Interface and User Experience (UI/UX) Design. The Geographi Information Science & Technology Body of Knowledge (2nd Quarter 2017 Edition), John P. Wilson (ed.).

Sack, C. (2017). Web Mapping. The Geographic Information Science & Technology Body of Knowledge (4th Quarter 2017 Edition), John P. Wilson (ed.).

Severance, C. (2012). JavaScript: Designing a Language in 10 Days. Computer, 45(02), 7-8.

Swift, J., and Goldberg, D. (2019). Web GIS Programming. In The Geographic Information Science & Technology Body of Knowledge (1st Quarter 2019 Edition), John P. Wilson (ed).