Curriculum Development and Pedagogy for Teaching Web Mapping

By

Carl M. Lemke Oliver Sack

A dissertation submitted in partial fulfillment of

the requirements for the degree of

Doctor of Philosophy

(Geography)

At the

UNIVERSITY OF WISCONSIN–MADISON

2018

Date of final oral examination: 7/27/2018

The dissertation is approved by the following members of the Final Oral Committee:

Robert E. Roth, Associate Professor, Geography
Kristopher N. Olds, Professor, Geography
Ian A. Muehlenhaus, Associate Faculty, Geography
Leema K. Berland, Associate Professor, Curriculum and Instruction

Table of Contents

Acknowledgements

I am most grateful to Dr. Robert Roth for his steadfast and superb advising, feedback, encouragement, and flexibility, all of which enabled this dissertation to come to fruition. The dissertation has also benefitted from the contributions of my other committee members, Dr. Ian Muehlenhaus, Dr. Kristopher Olds, and Dr. Leema Berland, as well as the helpful feedback of Dr. A-Xing Zhu and Dr. Qunying Huang on the proposal. My gratitude further extends to Tanya Buckingham for providing a stimulating and supportive work environment in the UW Cartography Lab, to Robin Tolochko for enabling and supporting the collection of data for this project while instructing the lab component of the Interactive Cartography and Geovisualization course, and to Dr. Rich Donohue for teaching me how to code in the first place and providing the initial inspiration for this dissertation. Thanks also to Fond du Lac Tribal and Community College, and especially to Dr. Anna Fellagy, for supporting my pursuit of the dissertation's completion while a faculty member there, and to my wife, Zee Lemke, for spousal support. Finally, thank you to my interview participants, and to all of the students who took Interactive Cartography and Geovisualization at UW–Madison between 2014 and 2017, without whom this dissertation would not have been possible.

List of Figures

List of Tables

I. Introduction: Teaching Cartography in the 21st Century

Abstract

This chapter introduces the subject matter of web mapping education. Section 1.1 outlines recent changes to the discipline and craft of Cartography, especially those driven by the advancement of internet technologies. Section 1.2 provides a more detailed explanation of the sweeping changes that have taken place in cartographic practice due to the expansion of online media since the early 2000s. Section 1.3 presents the parallel web-driven changes occurring in higher education. Section 1.4 explores possible reasons for the lag in adoption of web development skillsets within cartography and GIS curricula. Finally, Section 1.5 presents the research questions and outlines the remainder of the dissertation.

1.1 Overview

Cartography is changing. In the 20th Century, Cartography was:

- Scientific and objective;
- Used for presentation of information;
- Based on centralized, authoritative datasets curated by institutions;
- Conducted by draftsmen (and a very few women);
- Done using drafting tools, film, and (in the final decade of the century) desktop graphic design software;
- The production of maps on durable media.

In the 21st Century, Cartography is:

- Reflexive, subjective, and/or utilitarian;
- Used for exploration, analysis, synthesis, and presentation of information;
- Inundated by disparate, voluminous geographic information from volunteers and sensors as well as greater accessibility to institutional datasets through the internet;
- Conducted by graphic designers, computer programmers, and amateurs;
- Done using GIS software, desktop graphic design software, online applications, web servers, computer code, and the Open Web Platform;
- The production of both printed maps and *web maps,* or maps intended for digital viewing on a diverse variety of devices and screen sizes, including and especially web maps that support a rich set of human-computer interactions (Buckingham and Dennis Jr., 2009; Crampton, 2010; Goodchild, 2007; Harley, 1989; Harvey, 2012; MacEachren, 1994; MacEachren and Kraak, 1997; Reichenbacher, 2003; Rød et al. 2001; Roth et al., 2017; Wood, 2003a; Wood, 2003b; Wood and Fels, 1992).

Has cartographic education kept up with the changes? In many respects, yes: as cartography has reasserted its theoretical importance for map design in the digital era, there have been numerous efforts aimed at recalibrating curricula to new student and professional demands (Woodruff, 2011). These efforts include the joint International Cartographic Association—Open Source Geospatial Foundation “Geo for All” initiative to promote collaboration between universities in developing educational resources (http://www.geoforall.org/), and the University Consortium for Geographic Information Science’s ongoing revision and expansion of the Geographic Information Science and Technology Body of Knowledge (http://gistbok.ucgis.org/). Much of the change over the past two decades has been driven by the spread of the internet, creating an enormous need for web programming skills in fields that employ cartographers (Jancer, 2017; U.S. News and World Report, 2017). Nonetheless, many academic Cartography and GIScience programs have struggled to incorporate web development into their curricula. For example, all 13 University of Wisconsin System four-year campuses offer GIS courses, three offer undergraduate GIS majors, and nine offer a minor, concentration, and/or certificate in GIS; yet only five campuses—including only 2 of the 3 GIS major programs—offer a course in making maps for the web at the time of this writing.

This dissertation leans on instructional design theory and empirical case studies to bridge the gap between the geospatial industry's demand for interactive web mapping skills and the instruction of those skills within collegiate GIScience programs. It aims to point the way forward to integrating the modern web as part of the art, science, and technology of cartography and GIS. The remainder of this chapter will motivate the need for such a dissertation.

1.2 Changes in Cartographic Practice

Rumors of Cartography's death have been greatly exaggerated (Hermansen, 2010; Wood, 2003a).

Though there is much to love about Denis Wood's (2003a) tirade celebrating the academic field's imminent demise, the economy has not just resuscitated but fueled the once-struggling discipline. In 2010, the Department of Labor's employment and training administration predicted a 35% growth rate in the number of overall mapping jobs, while in 2013 Google predicted 30% growth in demand for geoservices (Underwood, 2013). In 2016, U.S. News and World Report ranked "Cartographer" #1 out of the "Best Engineering Jobs" and #45 in its top 100 overall best jobs list (it has since fallen to #3 in the engineering category)

(Thatcher and Imaoka, 2018; U.S. News and World Report, 2017). The U.S. Bureau of Labor Statistics expects the number of cartographers and photogrammetrists (its job category for the field) to grow by almost 20% between 2016 and 2026 (Bureau of Labor Statistics, 2017). These are boom times for cartography, not its afterlife.

What explains the dramatic turnaround? When GIS came of age in the 1990s, it first extended, then replaced analytical cartography, the research focus built under the William Garrison-led quantitative revolution in geography (Barnes, 2004; Pickles, 2006). New digital and internet technologies delivering modifiable maps into every home and classroom further eroded the hegemony of academic and professional cartographers over mapping practice. Much discussion in the early 2000s revolved around a proposed 'democratization' of mapping through new open and accessible tools (Rød et al., 2001; Wood 2003b). Maps could no longer be divided between the disciplined creator and the passive user; thanks to interactive mapping technologies, the user becometh the map-maker (Crampton, 2010). Comparisons were made between new internet technologies and hand-drawn maps produced through innate human capabilities—comparisons that were perhaps somewhat ironic in their ignorance of digital divides, but nonetheless recognized the new competition that dedicated amateurs posed to the professionals (Goodchild, 2007; Ricker and Thatcher, 2017). Cartographers were faced with reinventing themselves or being put out to pasture.

Happily for the discipline, it quickly became apparent that while new digital tools might allow increasing numbers of people to make and share digital maps, they did not guarantee these maps would be *good* or *useful* (Wiseman, 2015). The problem became especially acute with new interactive web maps, many of which were developed quickly by technology industry professionals with computer science backgrounds and little exposure to cartographic design principles (Muehlenhaus, 2014). At the same time that cartographers were realizing they needed new tools in their toolbox, technology companies were realizing they needed to hire cartographers if their maps were to be useful and thus stand up to competition in the marketplace (O'Beirne, 2016). This unforeseen marriage between the two fields returned cartography to the fold of sought-after career paths, remaking it into a desirable high-tech field for upwardly-mobile job seekers (Underwood, 2013).

The tools of the trade that students of cartography must learn now include web development and software programming in additional to the design sensibilities and quantitative analysis skills traditionally taught by mapping instructors. However, many cartography and GIScience educators lack a complete understanding of these new tools and concepts, let alone an understanding of how best to teach them given limited time and resources. The question of *what* tools to teach has been addressed in part by the research of Donohue (2014) and Roth et al. (2014). Their research, reviewed in Chapter 2, describes the technologies and skillsets necessary to make a map on the *Open Web Platform,* the set of royalty-free technologies and standards that power the internet (W3C, 2015). The research questions outlined in Section 1.4 are aimed at exploring *how* these new tools can be included in cartography and GIS curricula to meet the current demands of the field.

1.3 Changes in Higher Education

There is a paradigm shift occurring in higher education.

Multiple social forces are creating this shift. One source of the shift is undoubtedly the bleeding of public universities of their remaining taxpayer funding and the largely consequent move toward a funding model reliant on generating greater tuition revenues (Mitchell et al., 2014). As a result of this funding decline as well as broader economic realities, more and more universities are looking toward applied and online programs designed to retrain working adults and professionals for higher-skill jobs in the workforce (Gardner, 2016). Such programs are increasingly offered through distance education platforms in order to attract students regardless of their geographic location, students who may not want to move their families or put their existing jobs at risk to obtain a degree from a world-class institution of higher learning (Bose, 2014).

According to data from the U.S. Department of Education, in 2012, approximately 2.6 million students were enrolled exclusively in distance education courses, or 12.5 percent of American college students (NCER, 2014). Between 2002 and 2012, the proportion of U.S. higher education institutions that considered online education critical to their long-term strategy rose from less than 50 percent to 60 percent, and the online enrollment average annual growth rate of 17.5% far surpassed the 2.7% average annual growth rate for overall higher education enrollments during the same time period (Allen and Seaman, 2013). Although the online growth rate may be slowing and even reversing slightly (Allen et al., 2016), online education has entered the mainstream and is here to stay. To take one Geography example, the Online Masters in GIS program at The Pennsylvania State University had a cumulative enrollment of 362 students from 48 states and 4 countries between 2005 and 2013 (Luo et al., 2014). Penn State's "Maps and the Geospatial Revolution" MOOC (Massive Open Online Course), which is non-degree-granting, enrolled over 12,000 students from 177 countries in its first offering, of which over 3,300 completed the course (Robinson et al., 2015).

However, these new programs would not be possible in their current form without a maturation of the technologies needed to offer effective, real-time distance education on a mass scale. Learning Management Systems (LMSs), Open Educational Resources, and MOOCs (Massive Open Online Courses) provide increasingly robust and usable environments for teaching both traditional and non-traditional student populations (Weller, 2014). All of these technologies are rooted in the Open Web Platform (W3C, 2015; see Section 2.1). Furthermore, the *concept* of 'Open' which underlies the web has been expanded beyond its esoteric roots in software development to the point of becoming a mainstream social and economic theory guiding development in many spheres, including education (Steele, 2012). The 'Open Everything' movement has created the impetus and justification for sharing curriculum materials in ways that would have previously been unthinkable to institutions concerned about guarding their intellectual property (DiBiase, 2012). The Open Web has provided a platform for this sharing, enabling educators to reach broad new audiences and universities to open new markets for their intellectual products (Bozkurt et al., 2015).

From one direction, then, Cartography and GIScience instructors have had to adapt to a new Open Web-based toolset for making maps. Additionally, as part of the broader higher education community, many also have adjusted their teaching methods in response to demand for content delivery over electronic networks and in distance education settings. While there are a host of recommendations for online and blended instructional strategies, little empirical research has been done regarding the needs for teaching web mapping in an online or blended environment. The highly technical skills required for web mapping entail special challenges for transitioning to web-based instruction. Part of this research seeks to expose and address these challenges by measuring the outcomes of fully in-person instruction against blended instruction in which the majority of the content is delivered online, with in-person assistance provided by an instructor.

1.4 Challenges to GIScience Education

Despite the changes described above, college and university GIScience programs that offer a course on web mapping in some form remain in the minority, and even fewer offer the subject using online or blended instruction. Little in the way of curriculum development resources for teaching web mapping has been published as of this writing (Ricker and Thatcher, 2017). One known set of challenges to developing curriculum relates to the nature of GIScience students, especially those taking Cartography and GIS courses through Geography departments. These students represent a diverse group of learners, often with little background in computer science (Muller and Kidd, 2014). While today's students are raised with supercomputers in their pockets, their ubiquitous technology use does not necessarily entail an understanding of underlying computer science concepts or an interest in pursuing careers that involve coding (Forrest, 2015; Molnar, 2015).

However, the general nature of students entering Cartography and GIScience programs does not tell the whole story of why colleges and universities have been slow to adopt web mapping into their curricula. More research is needed to elucidate the challenges that face instructors who have attempted to include web mapping in their courses, with the lessons applied to recommendations for teaching the subject that can lower the barriers to its adoption.

1.5 Research Questions and Dissertation Outline

Three problems have been outlined in general terms in Sections 1.2-1.4 above: the need for instructional design around new open web technologies in Cartography courses, the increasing demand for online learning options for Cartography and GIScience, and the lack of clarity around the challenges faced by web mapping instructors. The research described in this dissertation seeks to address these problems by answering the following research questions:

RQ1. What are the major barriers to teaching open web mapping, and what instructional practices can overcome those barriers?

RQ2. What skill-based learning outcomes for open web mapping are achievable in a one-semester upper-level undergraduate Geography course?

RQ3. How does student achievement of the identified learning outcomes for web mapping compare between fully in-person and modular, blended instruction?

The first question is aimed at revealing the problems and successes experienced by instructors of web mapping courses at various institutions as critical context for understanding how to design curriculum that meets the needs of a broad range of GIScience students. The second question centers on delineating learning outcomes for open web mapping. *Learning outcomes* are defined as the measurable cognitive processes or tasks that students should be able to accomplish on their own after completing the course (Spady, 1994). The third question tests fully in-person instruction against a blended approach that modularizes the skills instruction into content modules delivered online through a learning management system, in order to determine whether this is a viable approach to transitioning these skills to the online education setting. Answers to the three questions will be synthesized to develop a set of recommendations for teaching web mapping that can be implemented in a variety of instructional environments.

The remainder of the dissertation proceeds through five additional chapters. Chapter 2 elucidates the technical and pedagogical foundations of the dissertation in greater detail. Section 2.1 defines a web map, reviews the history of web maps, explains their technical underpinnings, and categorizes them based on the system design interests of cartographers and GIScientists. This section is largely based on the author's entry in the GIScience & Technology Body of Knowledge, *Web Mapping* (Sack, 2017). Section 2.2 reviews how GIScience educators have approached teaching pedagogy to date, specifically covering the contributions of constructivist teaching philosophy and their application to GIScience education. Section 2.3 then contrasts teaching web mapping with instruction in other GIScience topics and reviews research conducted on web mapping instruction to date. Section 2.4 looks at the growth of GIScience distance education programs and research on teaching strategies employed by online GIScience courses.

Chapter 3 addresses the first research question through an interview study conducted with web mapping instructors in a variety of higher education settings across North America. Section 3.1 motivates the study with a review of what is known about the state of web mapping education and the barriers to its greater adoption in cartography and GIScience programs. Section 3.2 describes the methods and coding scheme used in the study, while Section 3.3 gives the pertinent results of the qualitative analysis conducted on interview transcripts. Section 3.4 discusses the most common practices used and challenges experienced by web mapping instructors. Finally, Section 3.5 provides a set of recommendations for instructors seeking to incorporate web mapping into their curricula. The content of this chapter has been published in modified form in *Cartographic Perspectives* (Sack, 2018).

Chapter 4 addresses the second research question directly through the description of a curriculum design and evaluation process undertaken for the lab portion of an Interactive Cartography and Geovisualization course (Geography 575) at the University of Wisconsin—Madison. The course is part of a broader Cartography and GIScience program that includes undergraduate, postbaccalaureate, graduate, and professional degree levels (Roth, 2016). Geography 575 is considered an advanced Cartography course, open to upper-level undergraduate, Master's, and Doctoral students. It requires intermediate-level Cartography and Geocomputing courses as prerequisites, building on the respective cartographic representation and scripting skillsets from each with the addition of user interface/user interaction design and coding on the Open Web Platform (see Sections 1.2 and 2.2).

Section 4.1 situates the Interactive Cartography and Geovisualization course within the Cartography and GIScience program curricula offered at UW–Madison. Section 4.2 describes the process used to determine what open web technologies to use for teaching web mapping in the wake of the transition from proprietary browser plug-ins to the Open Web Platform. Section 4.3 describes the development of a new laboratory curriculum reflecting the different set of skills needed to utilize the Open Web Platform instead of the proprietary technologies that had been used previously. Section 4.4 covers the methods used to evaluate the new lab curriculum, while Section 4.5 gives the results of the evaluation study. Based on these results, Section 4.6 identifies a set of threshold concepts—key ideas needed to make a web map—and a set of learning outcomes that are generalizable to any web mapping course. The content of this chapter has been published in modified form in the *Journal of Geography in Higher Education* (Sack and Roth, 2017).

Chapter 5 extends the research described in Chapter 4 to a comparison of in-person and blended curricula to address the second research question. Section 5.1 discusses the differences between in-person and online delivery methods, including various combinations that have been included under the banner of blended methods. Section 5.2 describes adjustments that were made to the curriculum based on results from the Chapter 4 evaluation and how the revised curriculum was modularized for online delivery. Section 5.3 describes methods used to evaluate the revised curriculum, focusing on adjustments made to the process described in Section 4.3. Section 5.4 presents the results of two evaluations conducted on the blended lab curriculum. Section 5.5 compares the student outcomes of the two curriculum versions, answering the third research question.

Chapter 6 concludes the dissertation by reviewing how the three research questions were answered and applying these answers to the design of an example web mapping course. Section 6.1 addresses each research question in order. Section 6.2 describes the larger contributions of the dissertation to the fields of Cartography, GIScience Education, and Online Education. Section 6.3 synthesizes the findings of the three studies into curriculum for a community college web mapping course. Finally, Section 6.4 explores future directions for research into web mapping education.

II. Background Review

Abstract

This chapter reviews the published literature regarding web mapping, web cartography, GIScience education, teaching web mapping specifically, and online and blended instructional delivery models. Section 2.1 defines web mapping and describes its relatively brief but nonetheless multi-stage history. Section 2.2 reviews the cartographic design principles that have been suggested and empirically derived for web maps. Section 2.3 turns to writings on pedagogy for GIScience education, focusing on constructivist teaching methods that have been suggested as appropriate for GIScience courses. Section 2.4 applies this pedagogy to teaching web mapping specifically, elucidating the sparse literature on the topic published to date. Finally, Section 2.5 reviews recommendations for best instructional design and teaching practices for online distance education courses and blended courses that leverage a mix of in-person and online instruction. The chapter introduces many technical and disciplinary terms that are necessary to understand the research described by this dissertation, placing those terms within their historical and theoretical contexts. Key terms are *italicized* in the text below and collected alphabetically with their definitions in the Glossary at the end of the dissertation.

2.1 Web Mapping Definitions and History

A *web map* is a map that is published and accessed via the internet, usually as part of a web page (Sack, 2017). Web maps fall into one of two categories (Figure 2.1). *Static web maps* are map images rendered in the browser that do not change given user input (Roth, 2013). These include map images that the user can increase or decrease in scale via zoom functionality in the browser without changing the image itself. *Dynamic web maps* are web maps that change appearance as they are viewed. There are two sub-types of dynamic maps: animated and interactive web maps. *Animated web maps* change frequently and automatically, using time to represent one or more data attributes (Harrower, 2004). Animated maps often include simple playback controls, but generally are not considered interactive unless the user can do something other than play, stop, or skip ahead and back in the frame sequence. By contrast, *interactive web maps* change in response to user input, enabling a conversation between the user and the map that is mediated by the user's device and software (Roth, 2012). Because of the ubiquity of interactive maps on the internet today, many people casually think of "web maps" as synonymous with interactive web maps. The term *web cartography* refers to the visual design of both static and dynamic web maps, whereas *web mapping* refers to the process of designing and developing an interactive web map specifically (Ballatore et al., 2011; Battersby et al., 2014; Muehlenhaus, 2014). This dissertation focuses on web mapping skills and the most effective ways to teach them, addressing the pressing industry demands identified in Chapter 1.

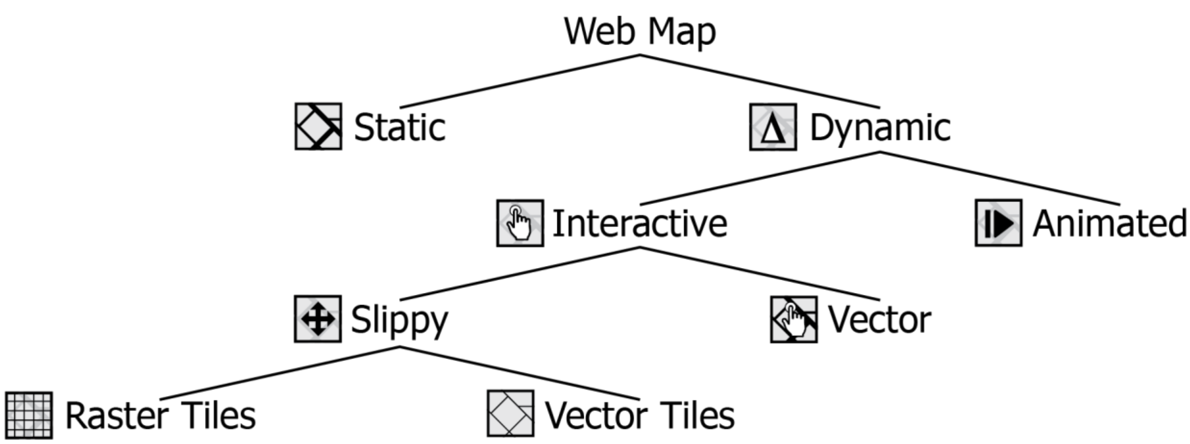


Figure 2.1: Web map taxonomy tree.

Web maps are almost as old as the internet itself, dating to the 1993 introduction of the Mosaic web browser, which allowed scanned map images to be transmitted to the user (Peterson, 2008). The first interactive web map was the Map Viewer website, developed in 1994. It allowed the user to request a unique map centered on a specific point (Peterson, 2014). In 1996, MapQuest introduced an online road map that enabled the user to zoom in and out and pan side-to-side, with each interaction request loading a new web page with the new map view. This was the dominant form of interactive reference map on the web for about a decade, until Google introduced the slippy map in 2005. *Slippy maps* allow for uninterrupted panning and zooming by cutting up the data or map image into 256-by-256-pixel tiles and sending tiles to the user's browser as the browser requests them without reloading the page (Peterson, 2012). Because of Google's use of Open Web standards and image overlays atop their base map tiles, their innovation also allowed 'map hackers' to create custom maps using data from other sources, known as *map mashups* (Gibson and Erle, 2006; Crampton, 2010). Google responded by openly publishing its *application programming interface* (API), a set of instructions that programs use to communicate with one another (Muehlenhaus, 2014). While OpenLayers was soon developed as a noncommercial alternative, Google Maps API remained the most used mashup generator until Google began limiting its unpaid usage in 2011. At that point, open source alternatives rapidly gained in popularity (Roth et al., 2014). Slippy maps remain among the most popular kinds of interactive web maps.

It is important to note that until recently, slippy maps were rarely developed by cartographers, but rather were the domain of web developers with backgrounds in computer programming. In the early 2000s, cartographers who sought smooth animation and interaction while retaining detailed control over map design instead turned to rich internet applications, or proprietary, third-party binary applications embedded in the browser as an extension (often called an applet). The most common of these for mapping was Flash, developed by Macromedia in 1997 and later acquired by Adobe (Peterson, 2008). Some maps were developed using Flash's competitors, Microsoft's Silverlight and Oracle's Java Virtual Machine. By downloading and compiling as a single executable package, programs built on these platforms could overcome the bandwidth limitations of 56K modems to provide a smooth user experience with high-quality rendering of custom graphics (Roth et al., 2014). They were also relatively immune to cross-browser compatibility issues that plagued websites during the browser wars of the 1990s and 2000s (Buckler, 2016).

Multiple trends converged in the late 2000s to force a shift away from third-party applications like Flash and to sole reliance on open web standards for interactive web maps. *Open Web standards* are programming languages including HTML, CSS, JavaScript, and XML, as well as data-handling processes such as AJAX (Asynchronous JavaScript and XML), that are defined and maintained by the World Wide Web Consortium (W3C) based on input from a wide variety of stakeholders (W3C, 2015). These standards conform to the concept of *free and open source software (FOSS),* or programs and technologies that give users the freedom to run, copy, distribute, study, change, and improve them without notifying or paying royalties to prior distributors (Free Software Foundation, 2016). FOSS products may be monetized and distributed commercially, but the program source code must remain accessible and the software must be licensed for modification and redistribution to be considered free and open source (Gaff and Ploussios, 2012). FOSS follows an ethos of personal liberty and voluntary collaboration independent of the interests of business or state actors (Stallman, 2015).

Open Web standards became ubiquitous in the mid-2000s with the adoption of standardized JavaScript, SVG, and HTML Canvas by all major browsers, ending the browser wars (Buckler, 2016). The Open Web saw vastly improved support for graphics rendering and interaction with the releases of the 5^{th} edition of JavaScript in 2009 and HTML5 in 2014 (W3C, 2014; McCormick, 2015). During the same time period, broadband internet became accessible to the majority of the population in the developed world, expanding the market for interactive web applications such as slippy maps that require uninterrupted connectivity to a server to function (Peterson, 2012). Given the growing influence of FOSS on the internet, the web development community sought to move away from proprietary, third-party technologies altogether in favor of sole reliance on Open Web standards (Berners-Lee, 2014). Apple's announcement in 2010 that it would no longer support Flash Player on its mobile iOS platform precipitated a stampede away from Flash by mobile web application developers (Jobs, 2010). With the explosive growth of mobile device usage, browser vendors sought to provide a consistent user experience across desktop and mobile platforms, leading them to phase out browser plug-ins and eventually discontinue support for them entirely (Smith, 2015). In 2015, Adobe itself announced its deprecation of Flash due to these trends (Adobe Corporate Communications, 2015).

As applets went away, cartographers both turned to slippy maps and began seeking alternatives that could overcome the design limitations of raster image tilesets (Bostock et al., 2011). The *SVG (Scalable Vector Graphics)* standard provided a way to draw and style maps that could still be dynamic and interactive directly in the browser. Although its adoption by some browsers took several years and its rendering speed was initially poor, SVG is now supported by all major browsers as of this writing (Chrome, Firefox, Safari, Opera, Internet Explorer, and Edge), and improved bandwidths and graphics processors have made its user experience competitive with that of the slippy map (Peterson, 2008). The newer HTML *Canvas* element supports faster rendering of vector linework (Bostock and Davies, 2013; Lienert et al., 2012). Interaction and 3D scenes can be added to Canvas using *WebGL,* a JavaScript API that uses the computer's graphics processor to further accelerate rendering (Caballero, 2011).

Like all web content, web maps follow a *client-server architecture* model (Donohue, 2014; Figure 2.2). The *server* is a piece of software installed on a computer or group of computers that sends information stored on its machine to a remote device, or *client*. Each server has its own *static Internet Protocol (IP) address*, a string of numbers by which the client locates it. The client—a browser or app stored on the user's device—makes requests for the information stored on the server's machine by making a call to the server's IP address (usually routed through a domain name server (DNS) address, e.g., 'www.example.com'). The server returns to the client the HTML documents, CSS stylesheets, JavaScript program instructions, and other necessary data and images stored in the website's directory. The client then compiles these data and instructions into the web page viewed by the user. The word 'server' has become shorthand for a high-power computer, often one that sits in a data center with hundreds of other similar machines (Peterson, 2014). However, server software can be located on any computer, even a home PC or laptop.

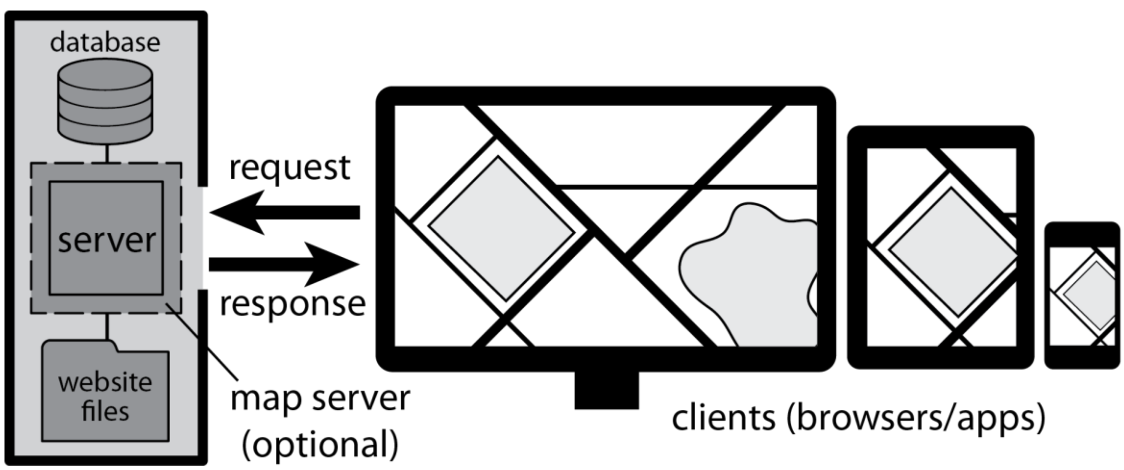


Figure 2.2: An illustration of client-server architecture for web maps.

On the client side, the browser renders pages and executes programs sent to it using the *Document Object Model (DOM),* which organizes all the elements, attributes, styles, data, and script objects and procedures needed to view the web page (Bostock et al., 2011). In addition to browsers, web maps are increasingly being viewed on mobile devices using apps developed to tap into a particular internet service. Google Maps and Apple Maps are the best-known map apps, but there are many others, and cartographers with basic coding skills can build their own for distribution through OS makers' app stores (Muehlenhaus, 2014).

There are now numerous specialized tools and technologies designed for creating interactive web maps. Increasingly user-friendly, cloud-based web applications have brought web mapping within the reach of technical novices (Muehlenhaus, 2014). Services provided by commercial vendors such as Mapbox, Carto, Google, and Esri allow users to add data to preexisting base maps, style the data, add interactive components, and host the final product for direct sharing or embedding in a web page. While these services are proprietary and fee-based, they generally provide a free option with restricted functionality, strong documentation, and APIs that integrate with open web technologies, increasing their accessibility over commercial desktop mapping software. These APIs are particularly useful for more advanced cartographers who wish to customize the representation and user interface of a web map (Peterson, 2014). There are also an increasing number of noncommercial, open source code libraries that have no specific parent service (Roth et al., 2014). Both proprietary APIs and open code libraries provide methods that simplify the use of the DOM to create web maps. The web mapping tools used for this dissertation's research are further elaborated in Chapter 3.

2.2 Principles of Web Map Design

Web maps perform a range of high-level functions that include data exploration, hypothesis confirmation, synthesis of findings, and presentation of knowledge, with the number of possible map views moving from high to low across the sequence (DiBiase, 1990). Based on these purposes, the Cartography^3 (Cartography Cube) framework (Figure 2.3) defines a "'space' of map use" using three axes: private to public, revealing unknowns to presenting knowns, and high to low human-map interaction (MacEachren, 1994: 6). In this schematic, web maps for exploration and analysis reveal previously unknown insights, generate unique views to suit the user's private interest, and thus have a high level of interaction, while those for synthesis and presentation present known patterns of phenomena, are widely distributed to the public in a very limited set of views, and greatly restrict interaction to focus the user's attention on the mapped information.

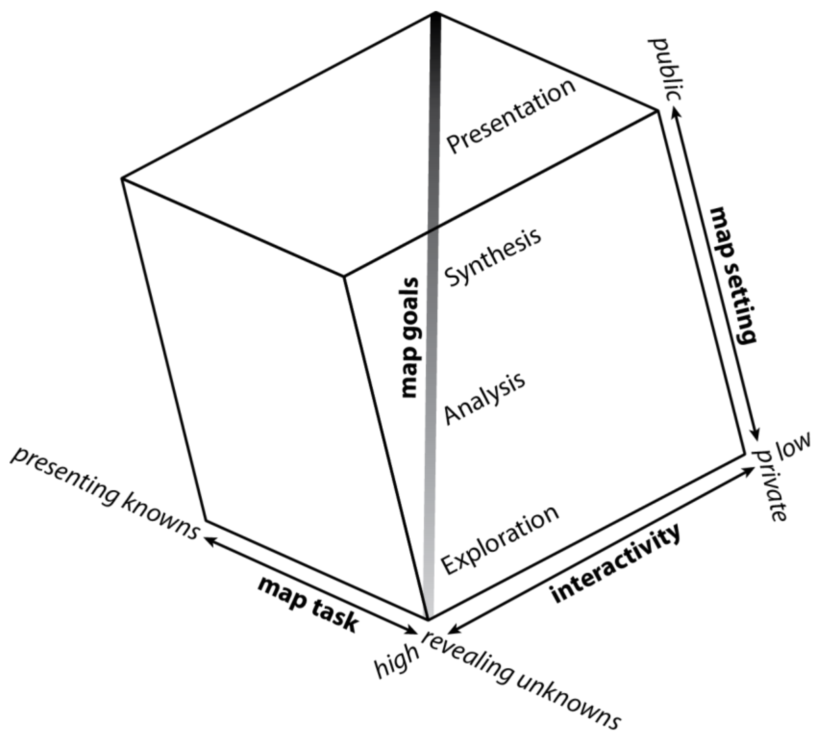


Figure 2.3: The Cartography[3] framework defines a space of map use based on the tasks the map supports, its setting, and its level of interactivity. Based on MacEachren (1994) and MacEachren and Kraak (1997).

Web maps can also be characterized by their *visual hierarchy,* or the relative visual dominance of objects on the map, with more important objects being more visually dominant (Dent, 1990). In general, since more exploratory maps aim to allow users greater freedom to draw their own conclusions, they often have a flatter visual hierarchy between data layers, with no layer particularly dominant. By contrast, thematic or story maps that aim to present specific findings or messages should have a strong visual hierarchy, accentuating the data that is most important while visually downplaying contextual information. To maximize its clarity and usability, a web map should avoid *overdesign;* that is, it should only include the information, map elements, and interactions necessary to accomplish the map's purpose (Tolochko, 2016).

Using these frameworks, web maps can be placed along continua from highly exploratory to highly thematic in purpose, high to low interactivity, and flatter to stronger visual hierarchy (Figure 2.4). A *web GIS* is a highly interactive, highly flexible web mapping application that allows the user to load their own datasets, perform spatial analysis, and create custom data visualizations (Fu, 2015). A *web geovisualization* is likewise highly exploratory and interactive, but employs limited, pre-selected datasets (MacEachren, 1994; MacEachren and Kraak, 2001). A *reference web map* (referred to by Muehlenhaus (2014: 64) as a "general-interest web map") typically contains many datasets visualized in such a way as to enable wayfinding, location-based services, feature search, and general landscape reading, but does not have the advanced interaction capabilities of more exploratory web maps. *Thematic web maps* present a small number of curated datasets with a specific intended message to the user.

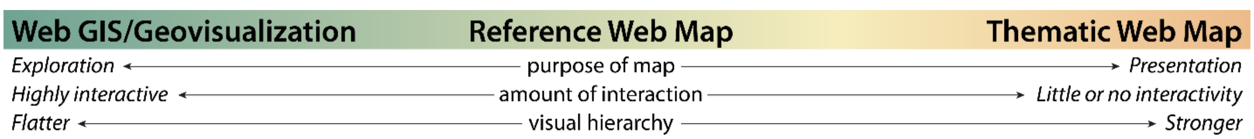


Figure 2.4: The Web Map Continuum.

The basic ingredients of a web map are *data*, *representation* of the data using style information, and, if a dynamic map, *animation* or *interaction* (Bostock et al., 2011; Donohue, 2014). Figure 2.5 expands on Figure 2.2 to show where each ingredient sits within the client-server architecture discussed in Section 2.1.

Data is hosted on the server's machine. How it is stored depends largely on the data model, the major distinction being between *vector* (object ontology) and *raster* (field ontology) models. Vector objects represent discrete points, lines, and polygons, whereas rasters are composed of a continuous grid of cell or pixel values (Longley et al., 2015). Vector data that may change dynamically—for example, as new information is added by users in a crowdsourced web map—are often stored in a database with a geospatial extension (Lienert et al., 2012).

Vector data may also be stored in a file on the server, or converted from a database to a file format for transmission. Vector data used in web maps typically does not include projection information, as most web mapping APIs project the data in the client application, and pre-projected data can generate incomprehensible results (Battersby et al., 2014; Bostock and Davies, 2013). Because they do not lend themselves well to database storage, raster map images (including raster tiles) are almost always stored as individual files in a simple directory structure. In web mapping, data design usually accounts for the greatest amount of time and effort, and can be subdivided into the processes of data acquisition, assessment, analysis, processing, quality verification, storage, and maintenance (Tolochko, 2016).

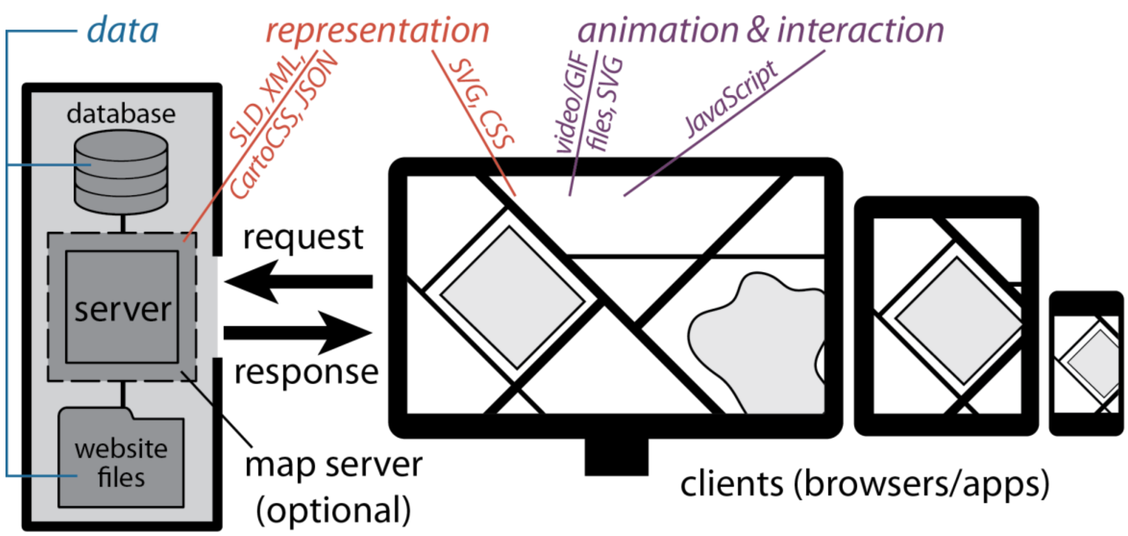


Figure 2.5: A revised version of Figure 2.2 showing where data, representation, and animation and interaction occur within the server-client architecture of the web.

The representation aspect of a web map, also called the *map composition,* is how the mapped information is seen and interpreted by the user (Donohue, 2014; Muehlenhaus, 2014). It involves the application of style information to the data to render a visible map image. This rendering may take place on either the server or client side. In the case of raster map images, the style information is included in the image provided by the server. The map images may either be preexisting files or rendered by a specialized *map server* using a stylesheet in one of several formats that include Styled Layer Descriptor (SLD), Mapnik XML, CartoCSS, and specialized JavaScript Object Notation (JSON) (Ballatore et al., 2011; Lienert et al., 2012). Map servers can be set up to render images on-the-fly as requested by the client, or render each image once and store it in a server-side cache (Peterson, 2012). Slippy map raster tilesets are typically created using the latter approach. Map servers publish data and images as geospatial web services using standards maintained by the Open Geospatial Consortium (OGC) to ensure interoperability between clients (Table 2.1). These standards allow clients to query the map server's data and operations to produce a custom result, and they immediately update with changes to the underlying data, allowing for real-time cartographic representation of changing geographic phenomena (Cerba and Cepicky, 2012).

Table 2.1: OGC geospatial web services.

OCG Standard	Description
Web Map Service (link is external)	Publishes static, whole raster map images rendered by the map server using an SLD stylesheet.
Web Map Tile Service (link is external)	Publishes raster image tilesets rendered by the map server for use in tiled web maps.
Web Feature Service (link is external)	Publishes vector data in GML format.
Web Coverage Service (link is external)	Publishes coverages, or geospatial information representing space/time varying phenomena.
Web Processing Service (link is external)	Specifies rules for client requests and inputs for geoprocessing on a server and for server responses.

For vector web maps, the style information is transmitted separately from the data and rendered by the client. In the case of SVG graphics, the style information may be included within the SVG file, or applied by separate instructions written in CSS or JavaScript, or a combination (Muehlenhaus, 2014). An SVG image may also be built from scratch on the client using separate data sources and instructions in the JavaScript code, as are all maps that are drawn using Canvas (Bostock and Davies, 2013; Lienert et al., 2012). Newer slippy maps may use vector tiles, which consist of many small chunks of data, each covering the geographic area of a single tile, and one separate set of style instructions that is used to render each tile in the browser (Turner, 2015).

The final component of a dynamic web map is the map's *animation* or *interactions,* which are executed on the client side. There are two basic types of animated web maps (Muehlenhaus, 2014). *Stop-frame animation* is provided through a video or GIF image file, which is composed of many individual images (or frames) that appear in rapid succession. Time-series data typically uses this kind of animation. *Tweening,* or a smooth transition between two map states, is created by program instructions in SVG or JavaScript code and is useful for showing continuous spatial-temporal data. *Interaction* relies on the user rather than the system to change the display (Roth, 2012). With the demise of rich internet applications, almost all web map interactions are implemented using JavaScript (Peterson, 2014). Google's invention of slippy maps pioneered creating smooth map interaction through the use of *AJAX (Asynchronous JavasScript and XML),* a set of procedures written into the JavaScript code that allows a client to send requests to a server and receive data in response to user input without having to reload the whole web page (Peterson, 2012).

Interaction is considered part of the *user experience* (UX) of a web map, and is enabled by the *user interface* (UI), the set of elements on the page that the user can see and/or manipulate (Roth, 2017). *UI design* concerns the iterative decisions made regarding the *map layout*, or the visual arrangement of UI elements on and around a web map, leading to the map's successful implementation (Muehlenhaus, 2014). *UX design* is the iterative set of decisions regarding user interactions leading to the user experiencing successful outcomes. Integrated UI/UX design is an increasingly important topic and skill in web mapping and software development generally (Haklay, 2010).

The web map user experience can be broken down using Norman's (1988) Stages of Action model (Roth, 2012; Figure 2.6). To interact, the user first forms a goal based on their motivation for using the map. They then form an intention, or a specific objective meant to further the goal, such as identifying a feature or comparing two features. If possible, the user then translates their intention into action using the UI elements provided for specific interactions, such as zoom buttons or a search box. Users discover what parts of the interface to use through *affordances,* the embedded clues that reveal how to interact with it (e.g., + and – symbols, a magnifying glass icon, shading to indicate an active button, etc.). The user executes the action using an input device such as a mouse, fingers, or keyboard keys, then perceives the system state based on *feedback,* or the signals that the interface gives to the user showing what result occurred. This feedback helps the user make sense of the new state of the map and evaluate whether they have achieved their goal. The cycle then begins again and continues so long as the user maintains an interest or goal to achieve in interacting with the map.

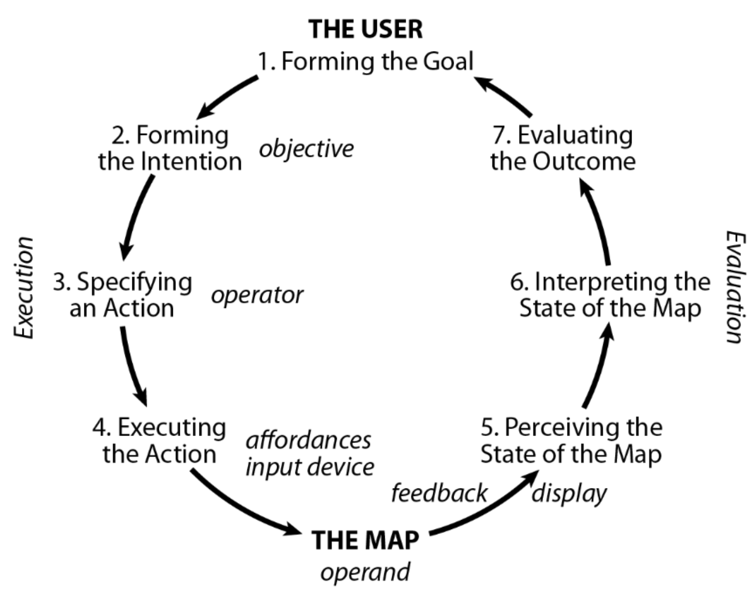


Figure 2.6: The Stages of Action model proposed by Norman (1988) as applied to cartographic interaction by Roth (2012).

Numerous taxonomies have been proposed to categorize map interactions. Roth (2012) found that most taxonomies exist at the "forming the intention" (objective), "specifying an action" (operator), and "map" (operand) stages of action. Operator taxonomies are most useful for categorizing the specific interactions available to users in a web map. Based on results of a card-sorting study with interactive map experts, Roth (2013; 2017) proposes a taxonomy of twelve interaction operators: *zoom, pan, overlay, retrieve, search, filter, sequence, resymbolize, reexpress, reproject, arrange,* and *calculate* (Table 2.2). This taxonomy does not preclude the addition of new operators as web maps evolve; for example, some apps enable map rotation, which does not fit neatly into one of the above categories.

Table 2.2: Common map interaction operators identified by Roth (2013).

Operator	**Definition**
zoom	change the map scale and/or level of detail
pan	change the geographic center of a map
overlay	add or remove feature layers or changed a mapped attribute
retrieve	view specific details about a map feature or features of interest
search	identify a particular location or feature of interest
filter	identify features meeting user-defined conditions
sequence	move between an ordered set of related map views
resymbolize	change the design parameters of map symbols without changing the map type
reexpress	change the type of thematic visualization used on the map
reproject	change the map projection
arrange	manipulate the layout of user interface elements
calculate	derive new information about map features of interest

In addition to this nomenclature, each map interaction has a certain degree of flexibility and of freedom enabled by the interface (Shneiderman, 2010). *Flexibility* refers to the number of interface components that can be used to implement the same interaction; for example, in many slippy maps, the user may zoom the map to a new scale in multiple ways: using zoom buttons, a mouse wheel, or a pinch gesture on a touchscreen. *Freedom* refers to the degree of precision or finesse with which the user can implement a certain interaction; for example, raster tile slippy maps only allow zooming to certain predefined levels, whereas vector tile slippy maps allow zooming to a virtually unlimited number of scales, thus providing greater freedom for the zoom operator. The interface freedom and the total number of interaction operators available in a web map, or the *interface scope,* together determine the *interface complexity,* or the number of unique map views a user can create (Roth, 2017). As per DiBiase (1990), the more exploratory the map, the greater its recommended interface complexity.

A final consideration for web map design is the map's accessibility to a variety of users. *Accessibility* for the web means that people who have disabilities, are older, or live in rural areas or developing countries can use websites and web applications (W3C, 2016). The World Wide Web Consortium (W3C) publishes Web Content Accessibility Guidelines, to which web maps should adhere. The W3C guidelines include methods to facilitate the use of screen readers and keyboard input by the visually impaired, as well as considerations for colorblind users already familiar to cartographers. Interactive web maps may be hampered by slower connection speeds, spotty connectivity, and older, non-conforming browsers, all of which are common in rural areas and the developing world (Tolochko, 2016). Regardless of location, device screen sizes and resolutions (number of pixels in each dimension) vary widely and affect how web maps appear to the user, while even modern browsers and operating systems have slight differences between them in how they display web pages, requiring the web map to be tested on multiple devices and platforms before its release. A well-designed web map will adjust its layout dynamically to fit the available display space of any device, following the strategies of responsive web design (Marcotte, 2011; Roth et al., 2018).

The creation of a usable and cartographically sound interactive web map requires the integration of complex code at the data, representation, and interaction levels. Teaching the skillsets involved is no simple task. The next two section will explore work that has been done to date on instructional design for GIScience education generally and for web mapping specifically.

2.3 Approaches to GIScience Education

Much of the discussion above appears in modified form as an entry in the recently revised Geographic Information Science and Technology Body of Knowledge (GIS&T BoK), hosted by the University Consortium for Geographic Information Science (UCGIS; online at http://gistbok.ucgis.org/). The BoK is the preeminent practice manual meant to guide teaching internationally across the subfields of GIScience, including web mapping. The original Body of Knowledge (BoK 1.0) was published as a static set of over 1600 learning objectives spread across 10 general knowledge areas intended to cover the breadth of GIS and cartography education (DeMers, 2009). The purpose of the Body of Knowledge is to lay out a set of core competencies that should be taught by all undergraduate degree and adult certificate programs in GIS (DiBiase et al., 2006). A revision process to create the second edition has been ongoing since 2012, and new chapters began being published in 2016 (UCGIS, 2017). The second edition (BoK 2.0) is published exclusively online and features a wiki-style interface, which chapter authors can use to update the content dynamically as current practices in GIScience education shift.

The Body of Knowledge is organized into ten knowledge areas: *Foundational Concepts, Knowledge Economy, Computing Platforms, Programming and Development, Data Capture, Data Management, Analytics and Modeling, Cartography and Visualization, Domain Applications,* and *GIS & Society.* Within each knowledge area are several topics viewed as necessary for students to understand, and each topic includes several learning objectives. Learning objectives identify specific tasks that students should be able to accomplish after completing a lesson or instructional unit. Taken as a whole, the Body of Knowledge's learning objectives constitute the range of competencies a student should acquire upon completion of a GIScience program in higher education.

Each topic entry identifies objectives at most or all levels of Bloom's Taxonomy of Cognition, which orders cognitive tasks according to their complexity (Bloom, 1956). The revised taxonomy by Anderson and Krathwohl (2001) categorizes cognitive processes from lower to higher order thinking skills as *remembering, understanding, applying, analyzing, evaluating, and creating.* Learning objectives from the Web Mapping entry in the BoK are listed with their cognitive levels in Table 2.3 (Sack, 2017).

Table 2.3: Learning objectives from the GIScience & Technology Body of Knowledge Web Mapping entry.

Learning Objective	**Cognitive Level**
Identify examples of static, animated, and interactive web maps.	Remember
Explain client-server network architecture.	Understand
Explain how a tiled map mashup is created.	Understand
Use a geospatial web service in a map or GIS project.	Apply
Identify the sources of data, representation, and animation or interaction in an example web map and the roles played by each.	Analyze
Critique the usability of existing web maps, including visual design choices, user interface, and interaction affordances and feedbacks.	Evaluate
Determine a web map's intended purpose and assess its use of visual hierarchy and interaction based on that purpose.	Evaluate
Design, construct, and publish an interactive web map.	Create
Format the styling, text, layout, image resolution, and file type of a static map so that it can be included in a well-designed web page.	Create
Publish a web map service or web map tile service.	Create

Pedagogy is the set of instructional design principles guiding curriculum structure and learning activities based on what will most effectively accomplish the learning objectives identified for a GIScience course*.* The authors published in Unwin (2011) address the question of how best to teach GIScience in higher education by applying constructivist pedagogies. *Constructivism* is an educational philosophy that emphasizes the need for direct experience in authentic learning environments, the conceptualization of new information as cognitive schemata, and the importance of metacognition (i.e., reflecting on one's learning process) in the construction of new knowledge (Neisser, 1976; Fouberg, 2013). Major influences on constructivist thought included Swiss child psychologist Jean Piaget and Soviet psychologist Lev Vygotsky. Piaget believed that people organize knowledge of the world into cognitive schemata—sets of actions, mental operations, concepts, or theories—and that children go through stages of cognitive development involving changes to their cognitive schemata (Meece, 2002). While Piaget saw development and learning as individual processes, Vygotsky emphasized the relationship between the learner and society, viewing learning as a social process.

Vygotsky's concept of the *zone of proximal development* describes the cognitive gap between a student's current understanding or abilities and their potential given instructor and peer assistance (Vygotsky, 1978). Figure 2.7 represents the zone of proximal development as a gradient centered between two axes: the level of challenge of a concept to the learner, and the learner's level of competence with the given concept. Concepts that are below and to the right of the zone—not challenging enough given the learner's level of competence—risk inducing boredom and consequent loss of motivation. Concepts that are above and to the left of the zone—too challenging given the learner's competence level—risk causing anxiety in the learner, which can also lead to a loss of motivation. Based on Vygotsky's model, the ideal curriculum introduces concepts that are within the upper-left portion of the zone of proximal development, challenging the learner but not to the extreme of causing anxiety.

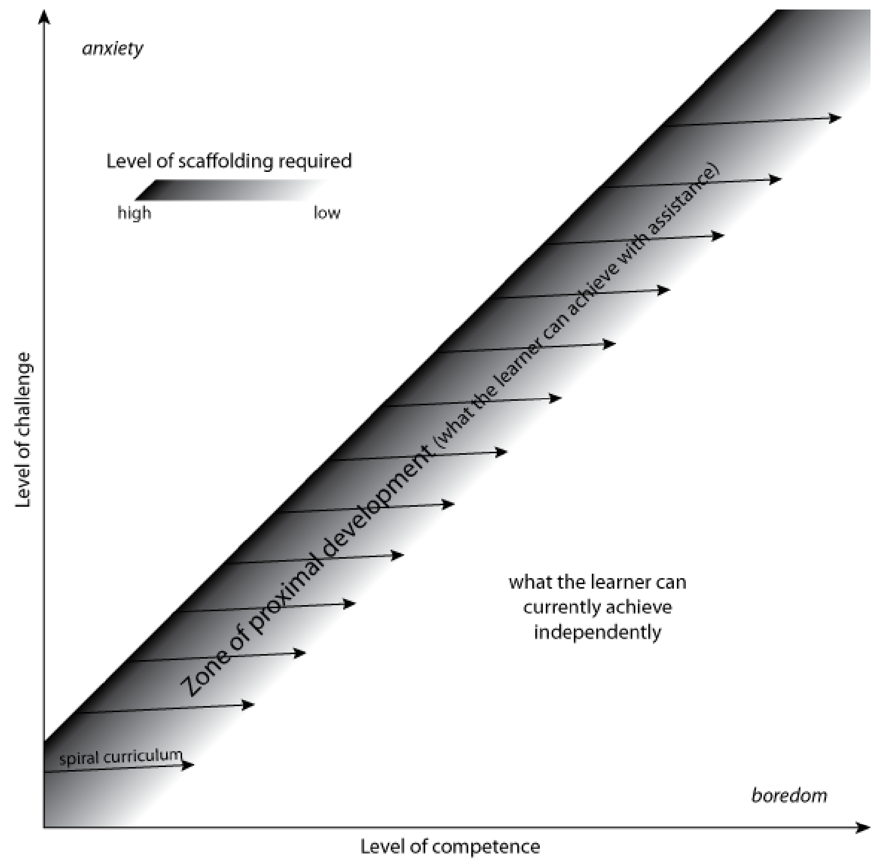


Figure 2.7: The zone of proximal development and its relationship to spiral curriculum and scaffolding

Learners can be supported through *scaffolding,* in which direct instruction and heavy learner support is provided when a new concept is introduced, then support is gradually withdrawn as the learner becomes able to work on the concept more independently (Palincsar, 1986). Initially, the instructor may demonstrate partial solutions to the problem and break the task down into highly structured and simplified steps that students can complete given their competence level. Over time, lessons on the subject are progressively more generalized to require less structure and assistance. Throughout the process, the instructor provides encouragement and feedback to the student and helps control student frustration (Wood et al., 1976). Students are evaluated frequently to calibrate the level of difficulty of a given task to their current abilities.

As the learner grapples with the concept, her level of competence moves to the right within the zone of proximal development until reaching the point of mastery, at which time a new, more challenging concept should be introduced that builds upon earlier concepts (Figure 2.7). This supports the construction of a *spiral curriculum,* one in which topics are carefully sequenced to build upon prior concepts at each new level of challenge (Bruner, 1977; Foote, 2011). In a traditional textbook-style course sequence, each learning unit introduces a new topic and expects students to master that topic before moving onto the next topic, which may bear little relation to the previous ones. By contrast, in a spirally sequenced course, each learning unit adds new challenges built on concepts introduced in previous ones, reviewing the same topics multiple times at a higher level of complexity each time (Figure 2.8). Students must engage with a wider set of concepts in each unit, but revisit those concepts repeatedly throughout the course until they are understood on a deep level and their relationships to one another become clear.

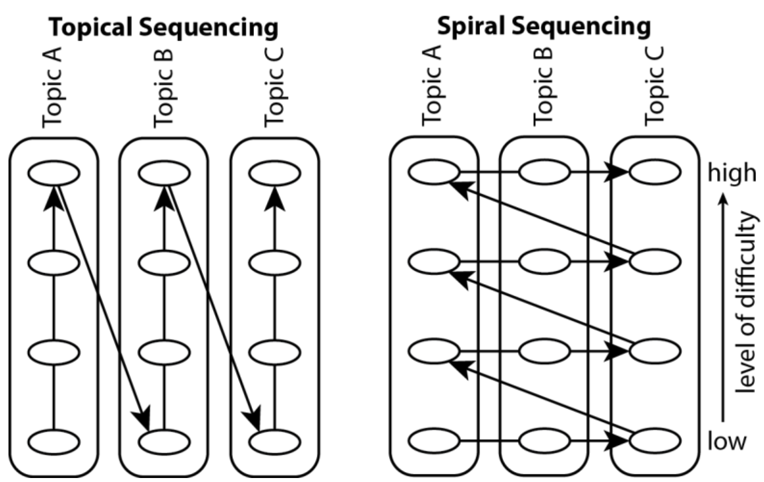


Figure 2.8: Topical versus spiral sequencing of concepts, based on Reigeluth (2007)

Two functions of a spiral curriculum are to calibrate the scope of concepts introduced during a course to the needs of the learner and subject matter, and to sequence those concepts in a way that supports mastery (Foote, 2011). *Scope* describes the depth and breadth at which course concepts are introduced. In another sense, scope can be seen as the range of concepts introduced during the course period. A course may cover a few concepts deeply, or a broad range of concepts with less emphasis on each concepts. Students may find a broad scope early on in the course overwhelming; hence the emphasis of spiral curriculum on introducing new concepts in a *sequence* that builds on prior understanding. Careful sequencing of course material can lead to more rapid mastery of concepts, theories, and skills by helping students logically organize the material in their cognitive schemata (Reigeluth, 2007).

Through scaffolding and spiral sequencing, students can be led to replace misconceptions and incorrect ways of thinking with *threshold concepts* (Bampton, 2011). Threshold concepts are not simply concepts that are difficult for students to master; rather, they are concepts that transform students' way of understanding, interpreting, or viewing a subject matter and must be internalized before the learner can progress any more toward mastery (Meyer and Land, 2003). Instead of being easily forgotten, they are difficult to unlearn. Threshold concepts may be understood suddenly, in the form of an *'aha!'* moment, or they may take time to absorb. The former process is easier to observe as evidence of a threshold concept than the latter.

Two other key constructivist concepts are *active learning,* wherein students are directly engaged in hypothesis testing, problem-solving activities, and *metacognition,* or reflection by students on their own learning (Schultz, 2011). In contrast to the passive learning approaches of lecture and notetaking, active learning includes kinesthetic activities—such as discussion, writing, and peer collaboration—that stimulate deeper thought processes and promote student engagement (Chickering and Gamson, 1987). Applying concepts to solving real-world problems supports authentic learning, showing students how their new knowledge is useful (Prager, 2011). Metacognition promotes problem-solving skills by encouraging students to understand not just the concepts themselves, but the strategies they used to master those concepts (Vos and de Graaff, 2004).

Under constructivism, the process of curriculum development as an ongoing, iterative process of creation, implementation, evaluation, and refinement (Painho and Curvelo, 2011). Given the complexity and conceptually challenging nature of web mapping, a semester-long web mapping course requires the application of high-powered teaching techniques that do a better-than-baseline job of moving students toward mastery. As the next section explains, web mapping is filled with challenging technical skills and stumbling blocks that are necessary to quickly move students beyond in order for the course to generate positive learning outcomes.

2.4 Why Learning Web Mapping is Unique

Interactive and open source web technologies have profoundly changed the production and distribution of geographic information. Apple Maps alone served more than 5 billion map requests per week in 2015, and thematic map producers in both the public and private sector—from The New York Times to the U.S. Geological Survey—are putting their maps online as interactive web applications (Jesdanun, 2015). While the growth of open, online, and interactive platforms present exciting prospects for map design and use, GIScience educators have struggled to keep pace with the rapid technological evolution taking place in the digital age (Roth et al., 2014). In particular, the shift to the Open Web Platform discussed in Section 2.1 increased the range of technical competencies students need to learn in order to make web maps, forcing educators to rethink the scope and sequence of topics introduced in lab curriculum among other changes to pedagogy and instruction (Donohue, 2014; Peterson, 2014).

Mapping on the Open Web Platform requires the integration of many disparate software tools and technologies to develop a usable custom product, as described in Section 2.1. Figure 2.9 shows a selection of the tools used for web mapping on the Open Web Platform; one or more of the tools from each of the first five layers must be combined to produce an interactive web map, while the sixth layer shows proprietary platforms that provide integrated and simplified mapping interfaces but do not support custom cartography. While by no means exhaustive, the diversity of represented technologies demonstrates the complexity of mapping on the Open Web.

Figure 2.9: A selection of commonly used Open Web Platform web mapping technologies.

The dizzying and ever-changing array of development environments, data formats, graphics specifications, scripting languages and libraries, and publishing platforms entails a steeper learning curve than all-in-one commercial packages like Adobe Flash (Donohue, 2014). Learners often have few early successes to reinforce learning and build confidence (Roth et al. 2014). This is particularly true for students and professionals only trained in desktop GIS and graphic design software, as lack of prior coding experience acts as a major barrier to getting started with the programming required by the Open Web Platform. The Open Web also challenges the cognitive connection between cartographic design concepts and map development in a way that Flash did not. Whereas Flash allowed the import of graphic elements designed with other desktop software in the Adobe suite, the Open Web performs cartographic representation through written code and stylesheets that cannot be rendered separately from the program execution (Peterson, 2014). This separation of design and execution makes it more challenging to relate the principles of cartography to the technical skills of web mapping.

However, Open Web Platform mapping technologies hold several advantages for GIScience educators. One obvious advantage is the low monetary cost compared to commercial software, which often can be unaffordable for students, especially once they graduate and are no longer covered by their school's enterprise software licenses. Additionally, open technologies hold pedagogical benefits. Teaching FOSS software may better facilitate higher-level learning objectives and group collaboration, as students can learn from other developers and each other through the sharing, reviewing, and manipulating of existing code (a process that also more closely mimics real-world development than prefabricated lab assignments). FOSS technologies often have robust developer communities dedicated to maintaining and improving the software, and train students to conform to open standards set by the developer community (Gaff and Ploussios, 2012; Open Source Initiative, 2006).

One way to organize the concepts necessary for students to understand Open Web Platform mapping is to put them in the context of a complete workflow. The *Web Mapping*

Workflow describes the process of constructing an interactive web map from start to finish and the core competencies necessary to do so (Figure 2.10; Donohue, 2014). The first stage of the workflow is setting up the *development environment*, the set of coding tools, debugging tools, and development server software necessary to make the map. The second workflow stage—finding, formatting, and loading the necessary datasets—is often the most time-consuming (Tolochko, 2016). The developer then builds the cartographic representation through web page element and stylesheets, and adds interactivity with JavaScript and code libraries; these two stages may be combined and iterative. Once a working prototype has been developed, it is deployed to a server for beta testing and eventual release.

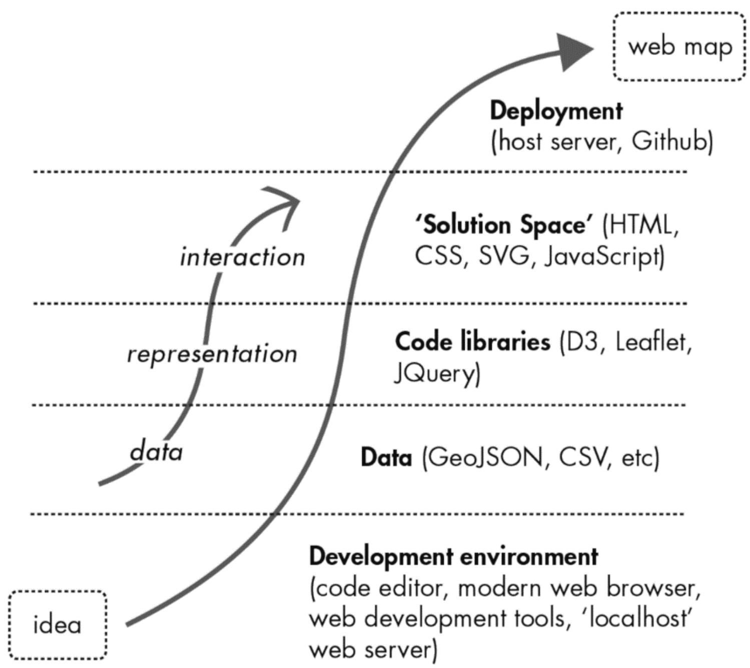


Figure 2.10: The Web Mapping Workflow from Donohue (2014)

Moving through the web mapping workflow requires a key set of cognitive tasks known as *computational thinking*. Wing (2011: para. 1) defines computational thinking as "the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent." Computational thinking components include the ability to visualize potential solutions for processing digital information and the ability to decompose large computational tasks into a logical series of specific steps (Raja, 2014).

Table 2.4: Dimensions and components of computational thinking from Brennan and Resnick (2012).

Dimension/component	**Definition**
Concepts	**The cognitive structures programmers employ as they program**
[Program] Sequence	A series of individual steps or instructions that can be executed by a computer to perform a task
Loop	A mechanism for running the same sequence of instructions multiple times
Event	An action that causes something else to happen
Parallelism	Sequences of instructions that are executed at the same time
Conditional	A statement that executes one of multiple outcomes based on a certain condition
Operator	Something used to support mathematical, logical, and string expressions, enabling the programmer to manipulate numbers and strings
Data	A structure for storing, retrieving, and updating values
Practices	**The habits programmers develop as they program**
Being incremental and iterative	Engaging in repeated cycles of imagining and building toward a final solution; developing a little, trying it out, then developing further based on experiences and new ideas
Testing and debugging	Strategies for anticipating and dealing with problems developed through trial and error, transfer from other activities, or support from knowledgeable others
Reusing and remixing	Reading and building on the work of others, enabling the creation of more complex solutions than could have been designed by the programmer on their own
Abstracting and modularizing	Building something large by putting together collections of smaller parts
Perspectives	**The views programmers form of the world around them and their place in it**
Expressing	The view that computation is something the programmer can use for creative design and self-expression
Connecting	The view that the programmer's creative practice benefits from access to others through face-to-face interactions or online networks
Questioning	The view that the programmer can interrogate the technologies that surround them and use those technologies to make sense of the world

Brennan and Resnick (2012) further describe three dimensions of computational thinking: 1) computer science concepts such as sequences, loops, events, parallelism, conditionals, operators, and data; 2) problem-solving practices such as being incremental and iterating, testing and debugging, reusing and remixing, and abstracting and modularizing; and

3) perspectives the developer forms of how computational concepts relate to the world around them, such as personal expression, connections with others, and feeling empowered to ask questions about and with technology (Román-González et al. 2017; Table 2.4).

While computational thinking is a critical prerequisite for web mapping, there is little GIScience literature to date addressing how to build computational thinking skills. There are a very small number of publications that describe experiences with teaching web mapping. One (Stefanakis 2013: 4) implemented a broad-scope introduction to a variety of Open Web mapping technologies while "avoid[ing] spending time in teaching… fundamentals in infomatics, such as programming techniques." Robinson (2015) describes the structure, content, and delivery of the "Maps and the Geospatial Revolution" MOOC conducted by The Pennsylvania State University, highlighting instructional methods that are particular to the MOOC context but giving little information about the scope and sequence of course topics. Neither paper explores the application of Open Web coding concepts to cartography. Calls for Cartography and GIScience to embrace and incorporate web mapping technologies into curricula remain largely unanswered by the literature (Hermansen, 2010).

2.5 GIScience and Online Learning

There is now a sizeable literature on the recommended differences in pedagogy between in-person and online Geography and GIScience courses. Specific teaching recommendations from several pertinent studies are summarized in Table 2.5. One common theme in online education literature is the importance attached to constructivism as a theoretical underpinning of online pedagogy (Bozkurt et al., 2015; Crawford-Ferre and Wiest, 2012; Mundkur and Ellickson, 2012). Bozkurt et al. (2015) analyzed 861 research articles on distance education published between 2009 and 2013 and found that constructivism ranked third in the most used theoretical perspectives (behind 'community of inquiry' and 'collaborative learning'), and that Lev Vygostky's *Mind in Society,* a foundational book for constructivism first published in the U.S. in 1978, was the second-most cited reference work across all articles.

Table 2.5: Recommendations regarding instructional design for online distance education found in the literature

Recommendation	Authors
Develop teaching strategies for the online environment rather than transferring traditional pedagogy to online courses.	Baran et al., 2011; Bose, 2014
Engage in critical reflection about past experiences, assumptions, and beliefs toward learning and teaching to transform and improve online instruction.	Baran et al., 2011
Support student collaboration through discussion boards, online journals, and virtual meetings.	Mundkur and Ellickson, 2012
Support individual learning through private journals.	Mundkur and Ellickson, 2012
Provide multiple methods of content exploration and transmission, including both synchronous and asynchronous learning activities, as well as formal and informal two-way communication.	Crawford-Ferre and Wiest, 2012
Do not rely on synchronous activities (e.g., virtual meetings) when students may not have access to high-speed Internet or major time zone differences.	Luo et al., 2014; Mundkur and Ellickson, 2012
Orient both instructors and students to the instructional technology used.	Crawford-Ferre and Wiest, 2012; Goett and Foote, 2000
Teach students self-regulation skills, independent study skills, and online research skills including critical evaluation of source material quality.	Bose, 2014; Goett and Foote, 2000
Be aware of time zone differences and language barriers that may challenge international students.	Bose, 2014; Crawford-Ferre and Wiest, 2012
Provide technical support to both instructors and students.	Crawford-Ferre and Wiest, 2012
To support international students, provide clear, specific course expectations, give greater context for assignments, and rely heavily on audio/visual aids.	Crawford-Ferre and Wiest, 2012
Consider the needs of students with disabilities and comply with accessibility guidelines published by the World Wide Web Consortium, in Section 508 of the Rehabilitation Act, and in the Higher Education Opportunity Act.	Case and Davidson, 2011
Participate in online discussions to maintain a "virtual presence in the space of the course" and greatly increase student engagement.	Bose, 2014 (35); Robinson et al., 2015
Group activities are very hard or impossible given little opportunity for trust-building.	Bose, 2014
Older adults may be more successful and benefit more from online courses than younger adults.	DiBiase and Kidwai, 2010; Luo et al., 2014

Online instruction can be used in three different overall course formats: web facilitated, blended, and fully online (Allen et al., 2016). A *web facilitated* course uses web-based technology in what is essentially a face-to-face course, such as posting a syllabus and assignments to a *learning management system (LMS),* a software platform designed for the delivery of instructional content. A *blended* (or hybrid) course delivers between 30 and 80 percent of its content online, typically uses online discussions, and may have a reduced number of face-to-face meetings. In a *fully online* course, most or all content is delivered online, typically asynchronously (i.e., students independently schedule their completion of the coursework) and without face-to-face meetings. The curriculum evaluated for Chapter 5 of this dissertation is considered blended because most of the instructional content is delivered online, but face-to-face lab meetings are still held to assist students in working through the material.

Prior research has posited several benefits of blended learning compared to both traditional in-person instruction and online distance delivery. The socializing aspects of in-person class meetings can be combined with the ability of online discussion forums to facilitate better student articulation of ideas and reflections without the social barriers of face-to-face discussion, creating a community of inquiry (Garrison and Kanuka, 2004). Students who have grown up with interactive and web-based technologies may feel more at home ingesting content and interacting in an online format than through in-person lecture and discussion, while direct in-person instruction remains important as a means of supporting students in assuming increasing levels of responsibility for their learning as they move up the scaffold (Garrison and Vaughan, 2008). The use of multimedia in blended courses can appeal to multiple learning styles. The blended format can support various instructional pedagogies but lends itself to active learning, student collaboration outside of regular class periods, and increased interaction with the course material regardless of discipline (Glazer, 2012). Program administrators typically rate the learning outcomes of blended instruction as superior to both fully online and traditional face-to-face instruction (Allen et al., 2016). The increasing use of blended learning has been identified as an important near-term trend in higher education, with blended courses highly regarded by students and instructors due to the increased communication and interaction over fully in-person instruction (Johnson et al., 2015).

A curriculum designed for both blended and fully online settings must include instructional units formatted for the web. Segments of online instructional content are generally termed *modules;* modules may include general information, tutorials, activities, videos, and assessments (Adams Becker et al., 2017; Bose, 2014; Breetske, 2007). Based on the positive reports of blended learning cited above, one would expect learning outcomes for a blended course with online modules to be similar or higher in quality to a more traditional in-person curriculum. However, much of the literature on blended learning emphasizes increased engagement with abstract concepts as a core benefit. Further research can elucidate the benefits and pitfalls of blended learning for a highly technical and applied skillset such as web mapping.

III. Current State of Web Mapping Education: An Interview Study with Educators

Abstract

This chapter describes an interview study conducted with instructors of web mapping courses at colleges and universities across the United States and Canada, intended to answer the research question, *What are the major barriers to teaching open web mapping, and what instructional practices can overcome those barriers?* Section 3.1 introduces the lack of web mapping courses in most collegiate GIScience programs and proposes the need for research into the practices of those that exist and have been successful to stimulate the field. Section 3.2 discusses the methods used for the study, including the recruitment procedures, interview protocol, and coding scheme. Section 3.3 gives the results of the study, organized by the major themes discussed by study participants that relate to the research question. Section 3.4 summarizes the barriers to teaching experienced by web mapping instructors and the common practices those instructors use in their courses, answering Research Question 1. Finally, Section 3.5 makes several recommendations for instructors and programs that wish to add web mapping to their list of course offerings.

3.1 Motivation

Despite the growing importance of web mapping, a minority of collegiate GIScience programs in the U.S. currently offer a web mapping course. Research into technology adoption in higher education suggests a range of possible factors for the relative dearth of web mapping courses, including the complexity of web technologies, resistance to change, lack of motivation, lack of institutional support, lack of faculty time and resources, and/or negative experiences in prior attempts (Moser, 2007; Abrahams, 2010). The New Media Consortium categorizes these challenges as "managing knowledge obsolescence" and calls for the establishment of "processes…for both technology and pedagogy discovery so higher education professionals can filter, interpret, organize, and retrieve information in an efficient and insightful manner" (Adams Becker et al., 2017: 23).

Nonetheless, there have been some promising developments regarding web mapping education. With its recent updates, the Geographic Information Science and Technology Body of Knowledge now includes several topics and associated learning objectives relevant for web mapping curriculum, including *Geovisualization* (Çöltekin et al., 2018), *Mobile Maps and Responsive Design* (Ricker and Roth, 2018), *UX/UI Design* (Roth, 2017), and *Usability Engineering and Evaluation* (Ooms and Skarlatidou, 2018), in addition to the *Web Mapping* entry summarized in Chapter 2 (Sack, 2017). A recent textbook on Web Cartography (Muehlenhaus, 2014) and another on "Mapping in the Cloud" (Peterson, 2014) provide useful instructional resources. While still in the minority, an increasing number of GIScience programs are integrating web mapping into their existing courses or creating new courses focused specifically on web mapping (Hermansen 2010). A critical mass of courses that would tip the scales toward universal adoption of web mapping in Cartography and GIS curricula may not be far off (Abrahams 2010). The research reported in this chapter aims to hasten its arrival by increasing the awareness of the teaching strategies used by early adopters as well as the challenges they have faced in the process of learning to teach web mapping. It addresses the research question, *What are the major barriers to teaching open web mapping, and what instructional practices can overcome those barriers?*

3.2 Interview and Qualitative Analysis Methods

The research study reported here consisted of one-hour interviews conducted between August, 2016 and April, 2017. Institutions from the United States and Anglophone Canada were included in the study due to their accessibility and similarity of academic contexts, given that most North American universities tend to house GIScience programs within a Geography department (though this is not universally the case). Participants were considered to meet study criteria if they had been an instructor of record for one or more courses that taught students how to create an interactive web map. Potential participants were identified first by emailing faculty at institutions listed in the North American Cartographic Information Society's (NACIS's) University Labs directory (http://nacis.org/community/university-labs/). Additional potential participants were added based on recommendations of those who responded to the first round of emails. Finally, a general recruitment email was sent to the American Association of Geographers (AAG) Cartography Specialty Group listserv in advance of the 2017 AAG Annual Meeting.

Through this process, 92 potential participants were identified, although many of those did not meet study criteria, declined to participate, or did not respond to requests for an interview. Ultimately, 22 interviews were recorded two of which were subsequently discarded because the participants did not meet the study criteria, resulting in a total sample size of 20. Of these, 7 interviews were conducted in person at the 2016 NACIS and 2017 AAG meetings, and the remaining 13 were conducted by phone or video conference. The interviews were audio-recorded and transcribed in full for qualitative data analysis.

A semi-structured format was used for the interviews, providing consistency in participant answers while allowing for more natural conversation and follow-up questions on themes that were of particular interest to participants (Bozkurt et al. 2015). The interview protocol included questions on: 1) the participant's background and training, 2) the scope and sequence of topics covered by their web mapping course or courses, 3) the tools and technologies they relied on for teaching web mapping, 4) their attitudes toward proprietary and free and open source (FOSS) software, 5) their observations on industry trends in web mapping technology and practice, 6) the extent to which they taught the class in person, online, or using a blended approach, 7) their use and creation of open educational resources (OERs), 8) their preferred teaching pedagogy, 9) successes and challenges they had experienced with teaching web mapping, and 10) any techniques they identified as 'best practices' for teaching web mapping. The full interview protocol is included as Appendix 1.

Interview transcripts initially were analyzed using 26 codes organized into six categories: 1) the overall course context, 2) technology used in the course, 3) resources used in teaching the course, 4) the course setting (i.e., whether in-person, online, or blended), 5) the curriculum content, and 6) teaching philosophy and experiences (Table 3.1; Miles et al., 2014). Within each code, similar statements were tallied by frequency and extensiveness. The qualitative analysis revealed that certain codes were more relevant to the research question than others. The codes that are starred in Table 3.1 were determined to be the most relevant, and their results are presented in Section 3.3 and discussed in Section 3.4.

To assess the reliability of the analysis, an independent secondary coder was used to generate a Cohen's Kappa score. The two coders undertook three training rounds of transcript coding, compared results after each round, and adjusted the coding scheme to produce greater agreement. Finally, 25% of the transcripts (5 out of 20) were coded from scratch by both coders and compared to produce the Cohen's Kappa score. The three training transcripts were re-coded by the primary coder for final analysis, and the remaining 12 transcripts were coded solely by the primary coder. Final results produced a Cohen's Kappa score of 0.50, or moderate agreement (Hallgren, 2012). This analysis included statements for which a code was generated by one of the two raters with no corresponding code given by the second coder. Excluding such one-sided codes produced a Cohen's Kappa score of 0.78, or substantial agreement. Only the coding results of the primary coder are reported in Section 3.3.

Table 3.1: Interview coding scheme. Results for starred codes (*) are presented in Section 3.3 and further discussed in Section 3.4.

Code	Description
Category: Course Context	
BACKGROUND	a statement about the instructor's education, training, or prior teaching experience
NAME	The name of a course
PROGRAM	a statement about the program context in which the course or courses is taught (e.g., degree type, prerequisite courses, etc.)
STUDENTS	a statement about the characteristics of a student or students in the course
SUPPORT	a statement about the extent or nature of support for the instructor from their program or institution.
VISION*	a statement about the social or academic role envisioned for the course to fulfill.
Category: Technology	
TOOL*	a specific piece of software, hardware, vendor, or general category of those
OPEN	a statement about the nature of open data or open source software
PROPRIETARY	a statement about the nature of proprietary software or data
MOTIVATION*	a statement about why the subject does or does not prefer to use particular software
TREND	a statement about a trend of development in web mapping software over time
Category: Resources	
OER	a statement about open educational resources used or created by the instructor
TEXT	a purchased textbook or other commercially licensed resource
Category: Setting	
INPERSON	a statement about teaching in-person
ONLINE	a statement about teaching online
BLENDED	a statement about using a mix of in-person and online instruction
Category: Curriculum	
ORGANIZATION	a statement about the organizational structure of the course
SCOPE*	a general statement about the range of topics covered in the course
TOPIC*	a specific topic covered in the course curriculum
SEQUENCE	a statement about how topics are ordered or why they are in a certain order
OBJECTIVE	a statement regarding a desired function of the course
OUTCOME	an ability or result demonstrated by students who took the course
Category: Teaching	
PEDAGOGY*	a statement about the instructor's teaching philosophy or techniques
EXPERIENCE	a statement about the instructor's overall experience in teaching web mapping or related topics
CHALLENGE*	a statement identifying a challenge the instructor faced in teaching web mapping
DEVELOPMENT	a statement regarding course development and/or revisions

Based on the results of the coding analysis, seven codes out of the original 26 were judged to contain the most salient collections of statements pertaining to the goals of the study outlined above: VISION, SCOPE, TOPIC, TOOL, MOTIVATION, PEDAGOGY, and CHALLENGE. For these codes, the Kappa score with null values was 0.45, and without null values was 0.71. Similar statements tagged with each of these codes were grouped together into themes, which were then tallied according to frequency, or the number of times a statement belonging to the theme was made, and extensiveness, or the number of transcripts containing the theme. Each instructor generated multiple (sometimes many) themes for each code, so these themes should not be considered mutually exclusive. Themes that recurred in multiple transcripts are reported in Section 3.3.

3.3 Qualitative Analysis Results

3.3.1 Vision

The VISION code was applied to statements about the big-picture social or academic role that the instructor imagined the course to fulfill. VISION themes discussed by multiple instructors are reported in Table 3.2.

Table 3.2: VISION themes expressed by two or more interview participants.

Extent	Frequency	Theme
12	22	Prepare students for future jobs
5	6	Course fits the needs of the department/program
4	7	Teach geographic thinking
3	3	Improve general geospatial literacy
3	5	Produce students who make better maps
3	6	Course fills a niche that few other courses currently address
2	6	Skills fit regional job market
2	3	Provide add-on skills for non-GIS majors
2	2	Elective course in GIS major/minor
2	2	Course links geography to data analytics
2	2	Course focus fits the dominant trend of GIS toward web-based applications
2	2	Expose students to web mapping at a basic level
2	2	Expose students to a variety of mapping tools they can use in future work

Twelve web mapping instructors saw their major purpose in teaching the class as imparting skills that could be useful to students in future employment. These instructors made statements such as, “we're teaching students based on what the industry demand is,” and “[if]I want to prepare students for jobs and being productive GIS people, they need to be prepared to engage online resources.” Related themes mentioned by two instructors each included meeting regional job market demand (e.g., “we always have to be cognizant of who we're training and what the specifically local or regional job market looks like”), providing add-on skills for non-majors (e.g., “I wanted to create a class that… geographers could take and use”), following trends in the GIS industry (e.g., “I think this is… one of the directions that geoscience is moving in more broadly”), and exposing students to tools they could use in their future work (e.g., “I want them to know about all kinds of tools out there, so they can be well equipped for whatever job position they happen to be going into”). Five instructors taught the course because it fit the particular needs of their department or program, while three saw the course as filling an open niche. Four instructors saw web mapping as a useful way to teach students how to think geographically, and three each wanted to improve geospatial literacy and produce students who make better maps.

3.3.2 Scope

The SCOPE code, based on the concept of scope as defined in Section 2.4, was applied to statements about the overall range and depth of topics covered in the course (Foote, 2011). SCOPE themes discussed by multiple instructors are reported in Table 3.3. Themes in Table 3.3 are color-coded based on their disciplinary categories, described below.

Table 3.3: SCOPE themes expressed by two or more interview participants.

Extent	Frequency	Theme
9	17	Web mapping is integrated into broader course curriculum
8	17	Broad exposure to a variety of web mapping tools
7	21	Heavy emphasis on technical concepts over design concepts
6	15	Geographic thinking/big concepts
4	8	Critical theory
4	9	No programming
3	4	Balance of technical and design concepts
3	3	Heavy emphasis on design concepts
3	8	Not much programming
3	3	JavaScript coding
3	4	Basic web mapping introduction
3	6	Geospatial data
3	3	Server-side GIS/mapping
3	4	Application of tools to solve real-world problems
2	2	Web GIS
2	3	Cartographic design principles
2	4	Introductory/basic level material
2	4	Web map design principles
2	5	Python-based
2	4	User experience/user interaction design
2	2	Basic introduction to cartography
2	4	Not highly technical
2	4	History of mapping/GIS
2	3	Open source technologies
2	2	Acquiring and using GIS data

Nine participants stated that they integrated web mapping within a broader disciplinary category such as critical geography, cartography, and/or GIS. In Tables 3.3 and 3.4, themes are color-coded based on which category they generally fit, with orange representing technical web mapping concepts, yellow representing geography and critical theory, blue representing cartographic design, and green representing geospatial data and analysis. In Table 3.3, white (non-highlighted) rows represent more general SCOPE themes not specific to a disciplinary category.

Eight participants said they exposed students to a wide variety of web mapping tools throughout the course (see Section 3.3.4 for more specific discussion of tools). Seven stated that they maintained a heavy emphasis on teaching technical concepts, in keeping with the vision of web mapping as a career skill. Three instructors mentioned a focus on JavaScript coding, while four said they included no programming at all in their courses, and three specified that they taught very little or only a very basic level of it. Three participants discussed covering geospatial data, and three discussed teaching server-side mapping and GIS.

Three participants reported that they sought to balance technical and design concepts, and another three said they heavily emphasized design. Six instructors reported using web mapping as a platform for encouraging students to think critically or explore a "big idea," and four said they used it to explore critical geographic theory. For example, one participant believed that web mapping is "not just about learning how data is stored and distributed on the internet, [but] instead thinking… on a more holistic social and political level about what kinds of social and political forces these platforms and these technologies are situated in." No participants reported attempting to cover all three areas—technical concepts, design, and critical theory—in a single web mapping course.

3.3.3 Topic

The TOPIC code was more granular still than either VISION or SCOPE, examining specific topics that were covered during units of the course curriculum. Participants covered a wide variety of topics in their curricula, with almost every course seemingly unique. While this lack of cohesion could be judged negatively, one participant saw it as a positive, stating, "I would hate there to be a standard curriculum, so that everybody gets a very generic view of what cartography is… The more variety, the better." The topics listed in Table 3.4 are color-coded following the disciplinary categories discussed in the previous section (3.3.2), with technical topics directly related to web mapping highlighted in orange, cartographic design topics in blue, geospatial information processing and analysis topics in green, and geographic and critical theory topics in yellow.

Table 3.4: TOPIC themes expressed by two or more interview participants.

Extent	Frequency	Theme	Extent	Frequency	Theme
7	16	Introduction to web technologies/code languages	3	4	Interaction design
7	12	Geospatial web services	3	4	Data visualization
7	10	Volunteered/crowdsourced geographic information	3	3	Web feature services
5	13	Web cartography	3	3	JavaScript
5	10	Accessing data	3	3	Social media geodata
5	7	Cartographic design principles	2	6	Internet basics
5	7	Data processing	2	6	Open data
5	6	Map projections and coordinate systems	2	5	History of the internet
5	5	Color	2	4	Territory/tenureship/boundaries
4	6	HTML	2	4	Client-server architecture
4	6	Map critique	2	4	Mobile GIS
4	6	Using Story Maps	2	4	Real time GIS
4	6	Interface design	2	4	On-premises web mapping/web GIS
4	6	Publishing geospatial web services	2	4	Network analysis
4	4	Consuming geospatial web services	2	4	Cloud-based web mapping/GIS applications
3	9	Animation	2	3	Persuasive/propaganda mapping
3	8	Scale	2	3	Using ArcGIS Web AppBuilder
3	8	Cloud GIS	2	3	Interoperability
3	8	Spatial analysis	2	3	Data classification
3	7	APIs	2	2	Nature/philosophy of open source software
3	6	Symbolization	2	2	Construction of a web page
3	5	Web GIS	2	2	Review of web mapping tools
3	5	Web map architecture	2	2	Vector and raster data models
3	5	Vector tiles	2	2	Map layout/elements
3	4	GPS data collection using mobile devices	2	2	Developing mobile apps
3	4	Multiscale map symbolization/generalization	2	2	Thematic web map types

Despite the lack of emphasis on coding in the SCOPE themes, the most frequently mentioned topic was a basic introduction to web languages and technologies, discussed by seven participants. Geospatial web services were also mentioned by seven participants, while four participants spoke more specifically about teaching how to produce such services, and four discussed using these services. Crowdsourced or volunteered geographic information (VGI) was mentioned by seven participants. While “map critique” was mentioned by four participants, other “big idea” topics were mentioned by three or fewer participants; these included scale, general spatial analysis, the history of the internet, territory and boundaries, persuasive mapping, and the philosophy of Free and Open Source (FOSS) technologies.

3.3.4 Tool

The TOOL code captured the most granular set of themes, as it was applied to statements regarding a specific piece of software, hardware, general category of technology, or vendor name. Participants discussed teaching with almost 200 unique tools. Table 3.5 presents the 68 tools that were mentioned by at least two participants. The table is color-coded by tool type, including full-stack commercial mapping platforms, graphic user interface-based applications (which may be included within a commercial platform), Open Web Platform or OGC technologies, and JavaScript code libraries or APIs (see Sections 2.1 and 2.2). The first two types of tools—the greens—are primarily accessed through a graphic user interface (GUI) with a mouse or finger used as the main input device. The latter two—the browns—are primarily code-based, i.e., accessed and manipulated through text input. Tools listed in Table 3.5 that are part of a full-stack commercial mapping platform are also labeled with the logo of the platform to which they belong.

Esri, the most popular GIS software vendor, remains dominant for teaching web mapping. Sixteen participants made use of ArcGIS Online to teach web mapping skills, and thirteen mentioned Esri tools in general. Among Esri’s GUI tools, ten discussed using ArcMap desktop software, nine each used Story Maps and ArcGIS Server, seven used the Collector mobile app, and five used Esri's Web AppBuilder. Among code-based tools that integrate with ArcGIS Online, five participants each used Esri's JavaScript API and web application templates, and three used Esri feature services.

Table 3.5: TOOL themes expressed by two or more interview participants, color-coded by type, with vendor icons denoting full-stack commercial platform tools.

Extent	Frequency	Theme	Extent	Frequency	Theme
16	93	ArcGIS Online	3	14	Google Earth
13	52	Esri tools	3	11	Amazon AWS
12	27	HTML	3	6	Instagram API
11	61	JavaScript	3	6	OpenStreetMap
10	36	Mapbox	3	5	Google Fusion Tables
10	32	ArcGIS Desktop/ArcMap	3	5	Geospatial web services
9	36	ArcGIS Server	3	4	Google My Maps
9	22	Esri Story Maps	3	4	Esri feature service
8	14	Google Maps	3	4	jQuery
7	45	CARTO	3	3	Tableau
7	43	Leaflet	2	9	OGC web services
7	16	Preexisting web map applications	2	8	Google tools
7	13	ArcGIS Collector	2	8	GeoServer
7	7	Mobile device	2	8	OpenLayers
6	15	QGIS	2	7	Python
6	11	CSS	2	6	HTTP
6	9	Web browser	2	5	URL
5	20	Google Maps API	2	5	Vector tiles
5	19	ArcGIS API for JavaScript	2	5	Twitter API
5	16	Mapbox Studio	2	4	Learning management systems (LMS)
5	12	GeoJSON	2	4	Canvas LMS
5	9	Web AppBuilder for ArcGIS	2	4	Git
5	8	ArcGIS Web Application Templates	2	4	Survey123 for ArcGIS
5	7	Web map service (WMS)	2	3	Shapefile
5	7	Web feature service (WFS)	2	3	TopoJSON
5	5	KML	2	3	CARTO Builder
4	22	GitHub	2	3	Notepad++
4	14	Adobe Illustrator	2	3	Microsoft Windows
4	13	Esri	2	2	Course website
4	11	APIs	2	2	YouTube
4	6	GPS	2	2	Google
4	6	Excel	2	2	ColorBrewer
4	4	XML	2	2	Web server
3	15	TileMill	2	2	CARTO API

Mapbox was another commercial platform used by participants. Ten participants mentioned Mapbox (the name of both the vendor and its platform), while five discussed using Mapbox Studio, its primary GUI application, and three continued to use TileMill, Mapbox Studio's now-deprecated predecessor application. Eight participants discussed teaching with Google Maps, which could refer to either Google's mapping platform as a whole or its front-end GUI application. Related GUI applications included Google Earth, Google Fusion Tables, and Google My Maps, each mentioned by three participants. Five participants mentioned using the Google Maps API—Google's code-based interface—and five mentioned the KML data format, which is now an OGC standard but mostly integrates with Google tools. CARTO was used by seven participants, with its Builder application and code-based API each mentioned by two.

In terms of mapping tools that are not connected to full-stack platforms, six participants used QGIS, an open source desktop GIS platform, and two used GeoServer, an open source map server package. For independent code-based tools, seven participants mentioned teaching with Leaflet, a widely-used free and open source web mapping code library, while two mentioned using OpenLayers, a more fully-featured FOSS web mapping library.

Several participants also covered the underlying web languages in their courses. Twelve participants mentioned teaching HTML, eleven taught JavaScript, and six taught CSS. Of web standard geospatial data formats, GeoJSON was used by five participants, and KML and XML were each used by four. Open geospatial web services were also used, with WMS and WFS each mentioned by five and the broader category of OGC services to which those belong discussed by two. Two participants mentioned teaching students about vector tiles, which are used by Mapbox and Google for their tile services (and can now be produced by Esri's ArcGIS Pro and ArcServer). Although no participants mentioned raster map tiles, TileMill, discussed above, produces them.

While components of the web development environment were less frequently discussed, six participants mentioned web browsers and four discussed using GitHub, a version control platform popular among application developers. Additionally, three participants mentioned using Amazon AWS for cloud hosting services, and two mentioned web servers generally.

3.3.5 Motivation

The MOTIVATION code was applied to statements regarding why the participant chose to use a particular tool in their course. The 42 themes in Table 3.6 provide context for the prevalence or absence of tools listed in Table 3.5.

Two somewhat conflicting motivations occupy the top two positions, mentioned by nine participants each. These themes demonstrate the tension between providing students with experience in widely industry-used Esri products and exposing students to a wide variety of alternative tools they may encounter in the future. Some participants emphasized that Esri's products are "the most widely-used... professional web GIS technology," and students would be disservice by not teaching Esri products because "an incredibly high percentage of job applications… want you to have skills in ArcGIS." However, participants—including some who taught with Esri products—also believed that variety was important. In the words of one participant, "I don't want to be like one of those Esri shops where that's all you do, [because] that's not the nature of this field anymore." Another emphasized that "students need to understand that there's a lot of software out there that can do these same functions that is freely available," and that knowing about open source alternatives gave students "the ability to have some other tools in [their] toolbox, so to speak, when [they] encounter GIS problems."

Table 3.6: MOTIVATION themes expressed by two or more interview participants.

Extent	Frequency	Theme
9	21	Industry standard tool that students are likely to encounter in future jobs
9	19	Expose students to a variety of web mapping tools that may be useful in their future work
8	16	Ease of use
6	16	Tools integrate into full stack that addresses all course needs
6	8	Tool is easier to use/teach than alternatives
5	8	Free/no cost
5	8	Tool is popular/common
5	7	Instructor is familiar/comfortable with tool
4	10	Accessible to students
4	6	Knowledge of tool is desirable to potential employers
4	6	Tool is covered by an institution-wide site license at no additional cost
4	5	Interface is highly usable
4	5	Tool enables students to gain transferrable skills
4	5	Tool fits with instructor's ethical/ideological orientation toward open source
4	4	No time in course for exploring alternative tools
4	4	Tool enables students to collect data in the field
3	6	Tool is powerful
3	5	Instructor likes the tool
3	5	Students are already familiar with the tool or its ecosystem
3	5	Students want to learn tool
3	4	Tools work well for particular course needs
3	4	Department/program tradition or inertia
3	4	Tool excites students
3	4	Tools provides valuable job skills
3	3	Lack of instructor time to explore possible alternative tools
2	5	The tool demonstrates a particular topic well
2	4	Tool makes accessing data easier for students
2	4	Tools do not require programming skills
2	3	Aesthetics of the tool
2	3	Matches instructor's skill level/expertise
2	3	Tool provides an important web mapping component or concept
2	3	Tool provides a platform that students can use to access another tool
2	3	Tool enables students to easily create and learn about custom map tiles
2	2	Students can examine the inner structure of the tool
2	2	Instructor saw a demonstration using the tool
2	2	Tool is fun/amusing
2	2	Prior relationship with software vendor
2	2	Web mapping is more accessible/approachable than desktop mapping software
2	2	Tool provides opportunity for remote collaboration
2	2	Tool provides useful features for learning coding
2	2	Tool enables data visualization
2	2	Prior/alternative tool was deprecated

A motivation mentioned by eight participants was using tools that are easy for both students and the instructor to figure out, while six felt that a tool they chose was easier than the alternative tools they could have used. For one instructor, "[If] I can show them [the tool] and give them a pretty straightforward set of instructions and they can do it, that keeps my sanity… I don't have to help 30 students debug something." Six participants liked using tools that could integrate into a full stack of GIS technologies; this usually applied to Esri products.

Five participants expressed the need to use tools that were free, although this did not necessarily also mean open source. For example, one instructor used Google products because, "I wanted it to be a tool that [students] could take to any future job and use it for free." The cost advantage of using a tool already covered by Esri's institutional site license was mentioned by four as a reason for choosing their tools. Five participants each discussed tool popularity and their own familiarity with the tool. Three participants stated that the tool was chosen because it fit the needs of the existing course, and two each said that it demonstrated a particular topic or provided an important web mapping component or concept.

3.3.6 Pedagogy

The PEDAGOGY code was applied to statements about teaching philosophy, techniques, or methods used by the participant. This included but was not limited to statements using the name of a formal pedagogical model (e.g., Bloom's Taxonomy, active learning, etc.). Table 3.7 lists 50 pedagogical themes expressed by two or more participants.

Table 3.7: PEDAGOGY themes expressed by two or more interview participants.

Extent	Frequency	Theme
13	20	Final projects
11	21	Hands-on/active learning
8	9	Students modify templates
6	7	Students complete tutorials and exercises independently
5	7	Field data collection with mobile app
5	5	Online discussion boards
4	8	Video tutorial/demonstration included in lesson
4	7	Peer assistance encouraged
4	4	Simple/straightforward activities
4	4	Later exercises build on earlier topics
4	4	Students find their own data for assignments
3	8	Peer critique
3	7	Projects are open-ended
3	6	Content should be fun
3	6	Students must figure out a solution through independent research
3	5	Instructor uses software/service problems as a learning experience
3	3	Additional readings are assigned
3	3	Deconstructing existing web maps
3	3	Instructor directs students to online tutorials and resources
3	3	Students receive open-ended assistance during lab periods
3	3	Regular weekly or semi-weekly lab assignments
3	3	Lectures are kept brief
2	6	Topics are carefully sequenced
2	5	Guest speakers are invited
2	4	Lecture material is posted online
2	4	Open-book/repeatable online quizzes
2	4	Students engage in group discussion
2	4	Students are encouraged to explore
2	4	Curriculum addresses multiple learning styles
2	3	Do not use lengthy lab assignment instructions
2	3	Balance between theory and practice
2	3	Students choose which tools to use to complete an assignment
2	3	Web maps are included as examples in lecture
2	3	Traditional weekly lab periods
2	3	Each assignment has learning goals/objectives
2	3	Instructor teaches how to copy and paste code
2	3	Instructor assists students remotely using email, phone, and/or videoconferencing
2	2	Course gives students resources to pursue additional skills on their own
2	2	Activities require multiple pieces of software to complete a task
2	2	Activities demonstrate the utility of GIS
2	2	Course includes traditional lectures
2	2	Course balances lecture and lab activities
2	2	Instructor demonstrates code examples
2	2	Instructor focuses on design principles
2	2	Students critique existing web maps
2	2	Students engage in critical thinking and reflection
2	2	Students make a web map from beginning to final product
2	2	Students choose a topic of interest for their final projects
2	2	Bloom's Taxonomy
2	2	Instructor uses sandboxes to teach coding

Although no participants used the word "constructivism," several discussed teaching techniques tied to constructivist theory. Thirteen instructors required their students to complete a final project at the end of the course to apply the skills and concepts they had learned throughout the semester. Eleven participants discussed making the learning active and hands-on. For example, according to one participant, "almost everything I do is active learning to some degree," while another believed that, "if we want to learn something, we learn best by doing it." It was common practice to assign students existing web map templates to customize, with eight participants employing this strategy, while six said they had student work independently on assignments during lab periods. Four participants thought it generally important that students find their own data, thus keeping assignments relevant to their interests—or, as one participant put it, "to try to make [students'] maps a reflection of a lived reality."

Several instructors mentioned blended teaching techniques that leveraged online learning management systems. These included hosting online discussion boards (5/20), creating video tutorials (4/20), posting lecture material online (2/20), and hosting open-book and repeatable online quizzes (2/20). Four participants encouraged peer assistance, while three integrated peer critique into project assignments. Some also emphasized simplicity or enjoyment, including straightforward activities (4/20), keeping lectures brief (3/20), and keeping the course content fun (3/20). Finally, some participants tied their pedagogical practices to specific tools. Five participants reported using a mobile app such as Collector for ArcGIS to have students collect location-tagged data outdoors on personal devices and transfer that data using desktop or cloud-based GIS software. One participant liked using a data collection app because it "gives [students] a first-hand experience in… thinking about a question that you would answer using spatial data and [then] collecting [that] data."

3.3.7 Challenge

The CHALLENGE code was applied to statements regarding what was difficult about the course. Since modern web mapping technologies are both relatively new and technically complex, some challenges are to be expected. Highlighting the challenging areas may indicate where strategies should be developed for workarounds or improvements to instructional technique. Table 3.8 lists 31 themes, differentiated and color-coded by whether each theme was primarily a challenge to the instructor, to the students, or to the course as a whole.

Table 3.8: CHALLENGE themes expressed by two or more interview participants.

Extent	Frequency	Theme
12	31	Instructor: Keeping up with technology changes
10	20	Students: Coding/JavaScript
10	18	Instructor: Time to update curriculum
6	12	Instructor: Instructional time required to teach coding
5	9	Instructor: Teaching computer science skills to students with little background
5	8	Instructor: Server setup and maintenance
5	8	Instructor: Time to build or maintain own technical skills
4	7	Instructor: Lack of expertise in web mapping skills
4	5	Instructor: Institutional software support
4	4	Instructor: Low student motivation
3	5	Students: Lack of prerequisite skills
3	4	Instructor: Limited time in course to teach web mapping tools
3	3	Course: External web service outages
3	2	Students: Software use and problem solving
2	6	Course: Limited bandwidth
2	5	Instructor: Providing clear instructions to students
2	5	Students: Finding required data
2	5	Instructor: Course revisions required by software changes
2	4	Instructor: Solving student problems
2	3	Students: Completing tasks independently
2	3	Instructor: Balancing theory and skills
2	3	Instructor: Student use of incompatible browsers or operating systems
2	3	Instructor: Time requirements of teaching online
2	2	Instructor: Time required by students who struggle
2	2	Instructor: Time constraints of program
2	2	Students: Git/GitHub
2	2	Students: Disruptions from software changes
2	2	Course: Web service data or usage limits
2	2	Students: Understanding cloud data storage
2	2	Instructor: Choosing which tools to teach
2	2	Instructor: Assessment and grading

Twelve instructors found keeping up with the rapid pace of change in web mapping technologies difficult. Several discussed the implications of rapid change, including finding the necessary time to update their course curriculum (10/20) and finding the time to maintain and build their own technology skillsets (5/20). Two participants specifically described the pace of technology change as "exhausting," while at least one experienced developing "entire lessons and modules that were out of date before we could even teach the term, and so they had to be scrapped." One participant complained that, "there's no teaching for the teachers of these [web mapping tools]."

For students, according to ten participants, the most difficult aspect of web mapping to learn was coding, particularly in JavaScript. Six reported that it was difficult to find adequate time to teach coding, while five discussed the difficulty of imbuing JavaScript skills on a set of students who do not necessarily enter the course with adequate background in general computing, and three saw this as a cause of students' difficulties learning the material. Four participants discussed their own lack of expertise in web mapping as an impediment to teaching it. For instance, one participant described multiple teaching failures, and shared that, "I had to teach myself how to do this, and teaching yourself how to do it works well for making maps as a consultant, but doesn't work very well for teaching."

Other challenges were more technical in nature, such as the difficulty of setting up and maintaining an in-house web server for the course (5/20), a lack of institutional support for required software (4/20), and outages in web services that were relied upon to teach the course (3/20). The latter theme particularly came up in interviews conducted after a worldwide outage of Amazon Web Services, which powers thousands of major websites and services including the ArcGIS Online platform.

3.4 Discussion: Common Practices and Challenges

Interviews revealed a great deal of variety in instructional practices but relatively consistent visions for the role of web mapping courses. A majority of instructors (12/20) saw web mapping as a career skill, and teaching it as necessary to prepare students for the current GIS job market. At least four also viewed teaching these skills as enabling inquiry into broader critical and geographic questions. However, even those who stressed critical theory saw the technical skills of web mapping as benefitting students' future careers. To quote one instructor, "By focusing on web mapping… we can really hone in on what makes geography and geographic thinking special, and the skills that my students have that their competitors don't in the job market."

In terms of the overall scope of participants' courses, four general threads emerged: *standalone web mapping* (emphasizing technical interactive web map coding skills), *critical web mapping* (emphasizing the broader theoretical implications of web mapping technologies), *web GIS* (emphasizing geospatial data collection, processing, analysis, and presentation and spatial thinking generally), and *web cartography* (emphasizing map design for the web). As Figure 3.1 shows, while there was certainly overlap in teaching practices between these disciplinary categories, instructors who emphasized GIS and cartography concepts tended to rely on full-stack commercial mapping and GIS platforms, whereas those who focused on web mapping or critical approaches to web mapping technologies placed a heavy emphasis on coding skills.

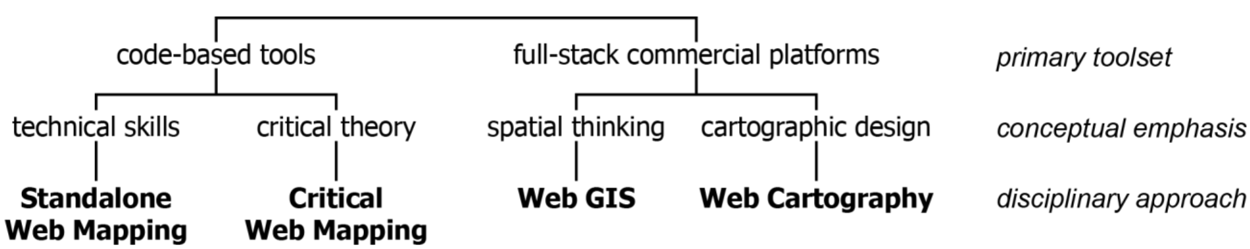


Figure 3.1: Four disciplinary approaches to web mapping curriculum, as defined by the amount of emphasis on custom coding versus the use of full-stack platforms and the conceptual scope of the course.

The most commonly taught topics were related to *enabling web technologies, data,* and *cartographic design.* Several instructors had students work with geospatial web services—producing them (4/20) and/or consuming them (4/20)—to demonstrate how data layers and maps can be shared in real time across networks. However, managing the necessary server software, whether in-house or in the cloud, was described as a challenge by a quarter of participants (5/20). Data—downloaded from traditional sources, collected in the field, or crowdsourced—and cartographic design are core components of web maps and were likewise key course topics.

In the market for teaching tools, Esri continues to exert dominance. While some interview questions were intended to prompt participants to reflect on the difference between open source and proprietary software and why they would use one over the other, most participants' answers showed this to be an amorphous divide. By far the more relevant division was between Esri and non-Esri software. Participants highlighted the need for tools to be free, easy to use and teach, and relevant to students' future jobs. Four instructors mentioned an ideological preference for open source tools, but none exclusively relied on them for teaching web mapping. For most instructors, the type of software license was of minor or no concern.

Esri's ArcGIS Online platform provides a full suite of scalable tools and applications that cover virtually every component of web mapping architecture. At its basic, free tier, the software includes a hosted web mapping service with an accessible graphic interface. For client-side development, Esri provides an open-source JavaScript API, easy-to-modify application templates, and open access to many of its web services for non-profit use. Subscriptions to more advanced spatial analysis, data collection, and hosting capabilities are fully covered by the vendor's educational site licensing and thus entail no cost to instructors. Esri's desktop software, often used in introductory GIS courses, is increasingly integrative with its online platform. Instructors stressed convenience, good documentation, vertically integrated applications, and absence of any additional cost as reasons for sticking with Esri, in addition to the vendor's continued dominance in the industry.

Nonetheless, there are downsides to Esri software that led some participants to consider other options. ArcGIS Server was frequently highlighted as difficult to set up and maintain. While templates are available for beginners to modify, the Esri JavaScript API is more complex than some open source mapping APIs. Two participants rejected Esri software out of ideological adherence to free and open source (but used competing proprietary platforms). The most frequently stated reason for using non-Esri software was simply to expose students to a wide variety of web mapping tools that they might encounter in the workplace. "I don't want [students] to know about just one thing, or one set of tools," opined one participant. "I want them to know about all kinds of tools out there, so they can be well equipped for whatever job position they happen to be going into."

In terms of pedagogy, hands-on active learning was seen as critical to student success. These are basic principles of constructivism, the philosophy that the role of the instructor is to assist students in building their own knowledge structures around the material (Foote 2011). They seem obvious in the case of a technical skillset such as web mapping, which operates at the uppermost, "create" level of Bloom's cognitive taxonomy, requiring students to synthesize concepts to produce an application (Anderson and Krathwohl 2001). Most instructors provided an exercise in creation via an end-of-term final project. What was not as expected was the prevalence of certain teaching techniques enabled by online learning management tools, such as hosting ongoing discussion boards and posting written and video tutorials for students to review. At least two participants commented on finding unanticipated benefits to posting lecture material and demonstrations online, such as increased comprehension among students for whom English is a second language, and the ability of all students to review the material and uptake concepts they may have missed during the class session.

The two greatest challenges in teaching web mapping were, unsurprisingly, teaching students how to code and keeping up with rapid technology changes in the industry. Most participants' courses are offered by a Geography department or closely related discipline, so few students come to them with advanced computer science skills or programming experience (one instructor was an exception, teaching at an institution that required all incoming students to take a computer science course). There seemed to be consensus among participants that a single semester is simply not enough time to turn beginners into coders at a higher than cursory level. Nonetheless, while four participants avoided teaching any code, others considered basic knowledge of HTML, CSS, and JavaScript important, even in courses with a broader scope such as Web GIS. Several participants (8/20) assigned students to modify existing templates as an approachable way to learn some basic web development concepts.

With the available teaching tools changing quickly, many participants struggled to find time to update course materials (10/20) as well as their own tool awareness and skillsets (5/20) given other teaching and research commitments. Most software vendors and open-source projects continue to support older versions after a new release, but those instructors who chose to use more innovative or cutting-edge products sometimes found themselves faced with acute disruption when a vendor chose to discontinue development of the selected tool. Further, the growing importance of interconnected cloud services may have promoted a false sense of security, as even the most trusted e-services were proven vulnerable to technical failure during the time period when interviews were conducted. At least three participants had their courses disrupted on February 28, 2017, when Amazon Web Services—which hosts ArcGIS Online—suffered a major outage caused by human error, knocking those tools offline (Del Rey, 2017).

Taken together, these factors require web mapping instructors to be nimble and adaptable to change, while maintaining technology blog subscriptions and attending technical conferences are increasingly required.

3.5 Conclusion

The interview study reported here was designed to answer RQ1, *What are the major barriers to teaching open web mapping, and what instructional practices can overcome those barriers?* This section will address this question based on the findings of the study.

The newness of the field, the rapid pace of technology change, and the lack of prior experience with computer programming among the current generation of students pose major challenges for web mapping courses. Instructors often lack the time to develop new curriculum, adapt existing curriculum to technology changes, and hone their own technical skillsets adequately amidst other academic and institutional responsibilities (although this could be seen as a problem in any academic field that teaches with cutting-edge technology). Additionally, the full range of technical skills—particularly coding skills—that are required to make a custom interactive web map is difficult for students to learn within the time constraints of a single-semester course. Finally, institutions may lack the capacity to support the specific technology needs of web mapping courses, or they may have policies in place for network security or system updates that do not align with the needs of web mapping instructors.

This research shows that despite these barriers, GIScience instructors have been able to integrate web mapping into their curricula using one of four disciplinary approaches: teaching web mapping as a standalone subject, teaching mapping technologies to promote critical theory concepts, including web mapping within a broader web GIS course, and emphasizing cartographic design as it applies to web maps. Courses that can attract upper-level students possessing at least some computer science skills, or that can be spread out over multiple semesters, may be well served by adopting one of the first two approach and going into greater depth with coding skills and code-based mapping tools. Courses with few or no computer science prerequisites can mostly rely on cloud-based, full-stack commercial mapping platforms to demonstrate the integration of spatial thinking and/or cartographic design concepts with web technologies. In these latter courses, small coding projects using prefabricated templates may be a useful introduction to the languages and developer skillsets of the Open Web. Regardless of the chosen approach, the use of well-organized, hands-on, active learning exercises appears to be essential to the success of a web mapping course.

While this interview study characterized the instructional strategies used by web mapping instructors to overcome teaching challenges, its ability to fully answer the research question is limited in two ways. It can only speak to those instructional practices that are used by participants. Second, due to the qualitative nature of an interview study, it did not test specific instructional practices in a controlled setting to determine which produced the best learning outcomes vis-à-vis web mapping concepts and skills. Chapter 4 further addresses the need to gauge learning outcomes for web mapping courses by applying a more in-depth evaluation of a single course structured around the constructivist practices of scaffolding and spiral curriculum.

IV. Lab Curriculum Development and First-Round Evaluation

Abstract

Chapter 2 examined the complexities of web maps, general approaches to teaching GIScience, and the specific challenges of teaching open source web mapping in higher education. Chapter 3 then surveyed how these challenges are being experienced by instructors of web mapping courses and the teaching strategies used by those instructors. This chapter focuses on the learning outcomes of a case study web mapping curriculum that utilizes specific constructivist pedagogical strategies. It is intended to answer Research Question 2, *What skill-based learning outcomes for open web mapping are achievable in a one-semester upper-level undergraduate Geography course?*

Section 4.1 situates the Interactive Cartography and Geovisualization course (Geography 575) within the broader Cartography and GIScience program curricula offered at UW–Madison. Section 4.2 describes the need for a new curriculum for the lab portion of Geography 575 to follow the shift in web mapping technology from Adobe Flash to the Open Web Platform (outlined in Section 2.1). Section 4.3 explains the curriculum organization, including the spiral topic sequence and the ways in which scaffolding was applied. Section 4.4 discusses the qualitative methods used to evaluate the curriculum, while Section 4.5 summarizes the results of the curriculum evaluation. Finally, Section 4.6 explores the threshold concepts and barriers to learning identified by the evaluation. It also develops a framework of learning outcomes experienced by students in the course, thus answering RQ2.

4.1 Web Mapping at UW–Madison

The University of Wisconsin–Madison is home to a long-running academic Cartography program that dates to 1937 and became highly influential in the field under the leadership of

Arthur Robinson and his successors (Roth, 2016). Cartography courses are housed within the school's Geography Department, where they are combined with GIScience courses to form one of four subfields alongside Human, People-Environment, and Physical Geography programs. The department offers two undergraduate majors: a major in Geography with a specialization in one of the subfields (Human, People-Environment, or Physical), and a major in Cartography & GIS. Additionally, the department offers a post-baccalaureate GIS Certificate, a traditional Master's Degree in Cartography & GIS, a non-thesis online professional Master's in GIS & Web Map Programming, and a Ph.D. in Geography focused on one of the four subfields. The online professional Master's program began in 2016 in response to increasing demand for training industry professionals in new web mapping technologies.

Web mapping as defined in Section 2.1 is regularly taught in two courses within the Cartography & GIS program sequence (Figure 4.1). Both are upper-level undergraduate courses that may also be taken by graduate and Certificate students. Geography 576, *Spatial Web & Mobile Programming,* introduces a range of client-side web mapping technologies along with programming map-based mobile apps. Geography 575, *Interactive Cartography and Geovizualization,* focuses on web cartography and UI/UX design for web maps (see Section 2.2). Both courses require students to have taken at least one prior computer programming course. Most students in those courses fulfill this requirement with Geography 378, *Introduction to Geocomputing,* which focuses on python and command-line scripting for GIS tasks. Several other courses within the department use web maps in some form. For example, Geography 170, *Our Digital Globe,* is a freshman-level survey course that includes an introduction to web maps along with other geospatial technologies. Geography 574, *Spatial Database Design and Programming,* covers geospatial databases, which are necessary server-side technologies for many web map applications. Graduate seminars in Cartography and GIS (Geography 970) may explore specialty topics related to web maps.

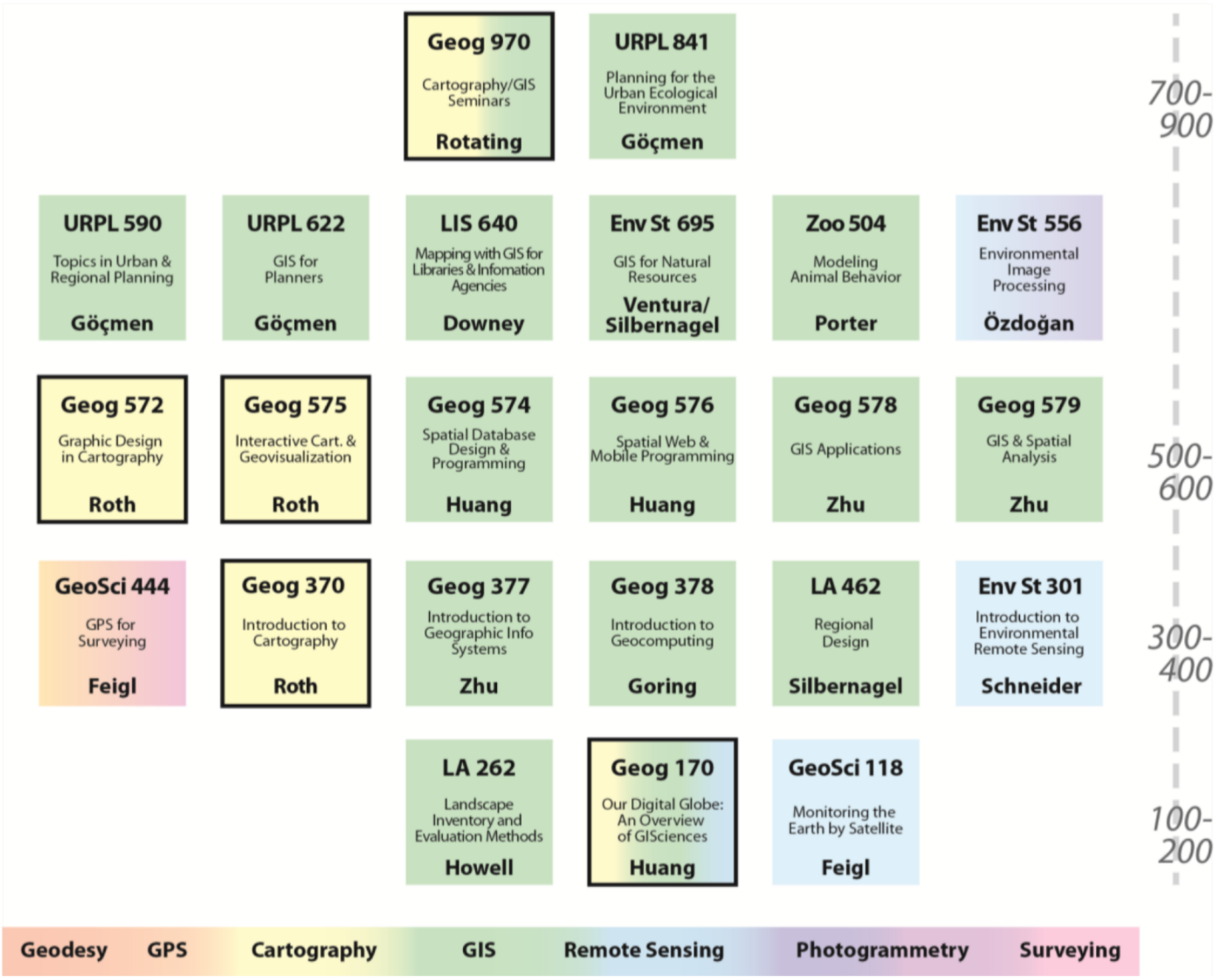


Figure 4.1: The GIScience curriculum at UW–Madison as of the 2015-16 school year, from Roth (2016).

The study presented in this chapter focuses on Geography 575, *Interactive Cartography and Geovisualization.* While web mapping is taught in both 575 and 576, the former course has a much longer history within the program and thus afforded the opportunity to build on a preexisting curriculum and examine the specific impact of Open Web Platform technologies on that curriculum. The course is divided into lecture and laboratory components, each allocated two credit hours. Lectures walk through tenets of UI/UX design and cartographic interaction as discussed in Section 2.2 (Roth, 2013; Roth, 2016). The lab sections serve as the technical complement to lectures, introducing the skills needed to implement lecture concepts and create a successful interactive web map. Labs are instructed by a teaching assistant (typically a graduate student) under the supervision of the course instructor, who delivers the lecture component. Student learning is assessed through two major exams and 8-10 quizzes covering the lecture material, along with two multi-week lab assignments and a final project. Because the lab component of the course covers competencies for mapping on the Open Web Platform, the lab portion of the course curriculum is the focus of this study.

4.2 Transitioning to the Open Web Platform

The Geography 575 lab curriculum was redesigned after a transition away from the use of Adobe Flash, to teaching the Open Web standards of HTML, CSS, JavaScript, SVG, the DOM, and AJAX, collectively known as the Open Web Platform (see Section 2.1). Constructivist pedagogies of scaffolding and spiraled curriculum, introduced in Section 2.3, were applied in the curriculum redesign to account for the increased range of competencies required for modern web mapping. The purpose of the research reported in this chapter was to assess the learning outcomes of the redesigned lab curriculum.

In 2012, as the developer community moved away from Flash, a study was conducted to determine what combination of then-emerging web mapping tools could best support the design concepts and examples presented in the course lecture (Donohue, 2014; Roth et al., 2014). The results of that study led to the selection of the Leaflet (http://leafletjs.com/) and D3 (http://d3js.org) JavaScript code libraries as the primary tools for teaching web mapping. While these tools are now widely used, they were nascent in 2012. Both libraries are open source and freely available on GitHub (http://github.com). Both are client-side tools that render maps and other data visualizations in the browser and enable cartographic interaction. Another widely used JavaScript library, jQuery (https://jquery.com/), was introduced along with Leaflet as a helper library to simplify rendering of web page elements and interactions. Several other enabling technologies were required in keeping with the integrated nature of the Open Web mapping approach discussed in Section 2.4. The full list of the technologies used in the course is provided in Table 4.7.

As mentioned in Sections 2.1, 3.3, and 3.4, such code-based tools are not the only solution; there are also commercial web mapping services such as ArcGIS Online, Google Maps, CARTO, and Mapbox that integrate server- and client-side Open Web Platform technologies, providing more contained and graphical interface-driven solutions for beginning web mappers. However, these services were not selected as learning platforms for multiple reasons: they were much less robust at the time the study was conducted than they are today, they remain much less flexible in terms of UI/UX design and map composition than code-driven solutions, and they promote reliance on proprietary, fee-based services rather than teaching students how to leverage open source technologies.

The Flash-based laboratory curriculum consisted of four multi-week assignments reflecting the topics presented in the lecture component of the course: (1) an "Animation Challenge" that introduced students to the Flash authoring environment and the use of ActionScript to create cartographic animation, (2) an "Interaction Challenge" that introduced user interface controls for sequencing through time intervals and retrieving thematic information from the map, (3) a "Coordinated Visualization Challenge" that taught how to coordinate user interactions across data views in a simple geovisualization application, and (4) an open-topic final assignment that required students to collaborate in groups of three, applying what they had learned to the development of an interactive map addressing a real-world problem.

The instructions for the first three laboratory assignments were initially rewritten to make use of Leaflet and D3 instead of Flash. The first two assignments were transitioned to a Leaflet code base, and eventually collapsed to a single, longer exercise (Donohue et al., 2014). The third assignment was recreated using D3 because of that API's utility for creating both maps and non-map data visualizations, and to present students with an opportunity to transfer learned skills to a new technology context (Sack et al., 2014). The final project was only changed as far as the expectation that students would make use of the Open Web Platform with Leaflet and/or D3 instead of Flash. The course was taught with the rewritten lab assignments in Spring of 2013. While the underlying technology changed from Flash to the Open Web Platform, the design considerations of assignments and the sequence of topics introduced in the course remained the same as they had previously.

Some students were very successful in the 2013 course, but quite a few struggled to learn and integrate the new Open Web Platform tools and technologies. Many students had difficulty extending the skills introduced in the lab assignments to generate custom solutions for their independent final projects (Donohue, 2014). The sheer range of concepts that students were expected to master caused frustration, resulting in learned helplessness for some. Learning D3 in particular turned out to be overwhelming for a number of students given its unique code structure. By the end of that semester, it was clear that there was a need for a more formally structured lab curriculum with greater attention to key stumbling blocks for comprehension of this highly technical material.

4.3 An Interactive Web Mapping Curriculum

The lab topic sequence was reorganized and expanded for the 2014 course offering to provide a better scaffolding for learning the required technologies. Constructivist educational theory provided the framework for this redesign. As described in Section 2.3, constructivism emphasizes direct experience in authentic learning environments, conceptualization of new information as cognitive schemata, and metacognition, or the learner's self-reflection of their learning process (Neisser, 1976; Meece, 2002; Fouberg, 2013). A constructivist learning environment for professional web development skills should thus facilitate direct experience with the tools, be relevant to students' interest domains, and allow for student control over the learning experience (Ellis, 2003). This approach to learning fits well with the ethos of free and open source software (FOSS), which emphasizes personal initiative, collaboration, and experimentation in software use and development. Leaflet and D3, the primary web mapping tools selected for the course, are examples of FOSS software.

The new curriculum was subdivided into a series of short lessons across four larger course units: 1) a preliminary tutorial assignment and two weeks of lessons in preparation for the first major lab assignment, 2) lessons and work time pertaining to the Leaflet lab assignment, 3) lessons and work time pertaining to the D3 lab assignment, and 4) the final project work period (Table 4.1). Each unit integrated concepts related to the three web map components of data, representation, and interaction (See Section 2.2; Donohue, 2014; Tolochko, 2016). The concepts introduced during the Leaflet and D3 units built on related concepts covered in prior units, setting up a curriculum spiral (see Section 2.3). The sequence of activities in each unit acted as a scaffold, entailing a heavy early emphasis on what were anecdotally observed to be key areas of difficulty for students in the 2013 course. Thus, there were large blocks of direct instruction and follow-along exercises in the first two weeks of the semester, and smaller lectures at the beginning of each successive unit, with progressively more independent work time following them.

Table 4.1: Outline of 2014 web mapping curriculum topics. The first column shows introductory topics presented prior to the first lab assignment, the second and third columns represent the time frame of each lab assignment, and the fourth column shows the time frame for the final project.

Pre-Lab	**Lab 1 (Leaflet)**	**Lab 2 (D3)**	**Final Project**
Week 0 (no lab meeting) • Online JavaScript Tutorial	**Week 3** • Using Reference Documentation • Online Forums and Examples • Slippy Map Tile Concepts • Leaflet Basic Concepts and Methods	**Week 6** • GitHub Concepts and Web Hosting • SVG Basic Elements and Attributes • D3 API Reference and Examples • D3 Core Selectors and Generator Functions	**Weeks 10-13** • Individualized Assistance
Week 1 • Text Editors • Directory Structure • HTML Basic Elements and Attributes • CSS Basic Style Rules • JavaScript Basic Concepts • jQuery Basic Concepts	**Week 4** • Using Developer Tools for Debugging • Custom UI Elements and Interactions	**Week 7** • Final Project Group Selection • D3 Interactions • D3 Geography	**Week 14** • Final Project Completion
Week 2 • Data Levels and Types • Geographic Coordinates • Data Language Specifications • AJAX (Asynchronous JavaScript and XML)	**Conference Week** (no lab meeting)	**Week 8** • Workshop Final Project Proposals • Individualized Assistance	
	Week 5 (final week for Lab 1) • Individualized Assistance	**Week 9** (final week for Lab 2) • Review: TopoJSON, D3 Projections, Debugging • Individualized Assistance	

Prior to the beginning of the course, students were required to independently complete an online JavaScript tutorial through either Lynda.com or Codecademy.com to introduce basic JavaScript coding concepts. For the early weeks of each unit, the lesson format included a mixture of lecture-style direct instruction, coding demonstrations by the instructor, and weekly active learning exercises that students were required to complete. Students were provided with example code from each lesson and pointed to relevant online tutorials, examples, and documentation to complete the exercises and solve problems. The topics covered during Weeks 1 and 2 were considered foundational web development and data handling skills, necessary for students to apply to their assignments throughout the course. As each unit progressed and students' skills developed, direct instruction diminished and students spent more of the lab period learning and problem-solving the assignment independently with over-the-shoulder assistance, following the scaffolding model. Over the course of the curriculum as a whole, each unit involved progressively less time spent on direct instruction, with the final project weeks dedicated solely to individualized assistance with problem-solving as needed.

The remainder of the chapter focuses on the evaluation conducted to test the efficacy of the reworked curriculum. The goals of the evaluation were to elucidate barriers that remained to student learning and to establish a baseline set of outcomes against which could be measured future iterations of the curriculum. It sought to answer the research question, *What skill-based learning outcomes for open web mapping are achievable in a one-semester upper-level undergraduate Geography course?*

4.4 Evaluation Methods

The revised lab curriculum was implemented and evaluated in the Fall Semester of 2014. Twenty-four students completed the course that semester. To test the efficacy of the curriculum, the evaluation used three instruments that collected a mix of qualitative and quantitative data regarding how students learned the material presented. This allowed for triangulation between results of each component to strengthen the reliability of the study findings (Merriam, 1988). The three instruments used were a lab instructor observation log with weekly entries, student feedback compositions collected at the end of each lab assignment, and an extensive exit survey administered at the end of the course. Additionally, a shorter entrance survey was administered to incoming students at the start of the course, but the results of this instrument had to be disregarded because survey respondents were kept anonymous and some later dropped the course, making the results not directly comparable to the information collected in the exit survey (this issue was resolved for the evaluations described in Chapter 5).

The instructor observation log was used to record the instructor's subjective reflections from each lab period. In a constructivist setting, lessons are a means to an end rather than the end in themselves, and thus rarely are executed exactly as planned (Spady, 1994). A detailed observation log was useful for generating a record of how the curriculum was actually implemented in the classroom, identifying unintended consequences of the curriculum, and obtaining direct evidence of students' mastery of the skills introduced (Lewy, 1977). The logs captured challenges encountered during lesson implementation, successful teaching experiences, and student reactions to the material. Critical incidents recorded in the logs provided insight into students' learning patterns and where they encountered successes or difficulties.

For each of the two major lab assignments, students were given the option to submit for extra credit a minimum half-page composition describing their struggles, 'aha!' moments, and suggestions for improvements to the assignment. Fourteen of these descriptions were received for each of the two lab assignments (the Leaflet and D3 labs). The content of these were subsequently coded following the tenets of qualitative data analysis into four categories: (1) concepts described by students as difficult, (2) other problems encountered in completing the assignment, (3) 'aha!' moments experienced while completing the assignment, and (4) other benefits perceived as coming from the assignment (Miles et al., 2014). Within these categories, similar statements were grouped into themes and the extensiveness of each theme was recorded, following a procedure similar to the coding in Chapter 3.

At the end of the course, students were asked to take an extensive online exit survey. Survey responses helped to confirm and interpret findings generated by the other instruments (Merriam 1988). Twenty-three students completed the exit survey, out of 24 who completed the course. The survey included five different sections each assessing a different dimension of the course: 1) competence with Open Web mapping tools, 2) level of challenge associated with Open Web mapping tools, 3) topic sequencing, 4) usefulness of learning resources, and 5) the experience of completing each lab assignment. The full exit survey is included as Appendix 2.

The first section of the exit survey asked students to self-assess their level of competence with 20 Open Web Platform mapping tools before and after taking the course. Fourteen of these were used as tools in the lab curriculum, and six were not, providing controls against which to measure the lab curriculum's impact on student learning of each of the technologies. Students' ratings were collected using a seven-point Likert scale, from "I have never used this specification" (1) to "expert level knowledge" (7), modeled after Prager and Plewe (2009). The second section of the exit survey asked students to rate how challenging they found each of the tools covered in the curriculum, also using a seven-point Likert scale. The third section asked students to reorder the topics (see Table 4.1) in any sequence that made more sense to them and to suggest topics that needed more reinforcement or were unnecessary and could be removed. The fourth section asked students to rate their reliance on various learning resources presented in lab and the helpfulness of each of those resources. The fifth section narrowed in on each of the two major lab assignments and final project. For each assignment, the survey asked which parts of the assignment students found challenging, what resources they relied on for help, whether they felt they had learned the material, and their overall emotional experience (a 7-point Likert scale from "extremely negative" to "extremely positive").

4.5 Results

4.5.1 Instructor Observation Log

The key insights from the instructor observation log are reported in chronological order below, reflecting the progressive increase in student capabilities throughout the course. This description references a number of technical terms that are described in Section 2.1 and the Glossary. While the findings below are anecdotal, they provide context for the results of the other two evaluation instruments.

During the Week 1 lessons, many students found it challenging to grasp the DOM, the conceptual framework that describes the internal structure of website elements. Understanding the DOM requires the ability to visualize abstract data structures, a key component of computational thinking (Raja, 2014). Students also found it difficult to understand when to use syntax belonging to JavaScript versus the syntax of jQuery and other code libraries. Identifying the correct syntax for a particular code library is necessary in order to apply the methods provided by library. In Week 2, the second week covering foundational concepts, students had difficulty grasping AJAX. Using AJAX requires careful attention to the order of execution of tasks in a script, as some tasks will only execute after data is received from a server (i.e., asynchronously with the rest of the script).

Leaflet was introduced to students in the second unit. The lessons for Weeks 3 and 4 largely relied on follow-along instructor demonstrations using the Leaflet library. Due to slower progress through the lessons than anticipated, debugging using in-browser developer tools was introduced during Week 4 as a homework exercise rather than directly in lab. Week 5, the final week students were given to complete the Leaflet lab assignment, primarily consisted of one-on-one assistance to students, representing a partial removal of the scaffold.

The third unit began with an introduction to the Git version control system and GitHub web service as the first lesson of Week 6. GitHub proved to be challenging to teach and to learn. GitHub requires the use of either a command-line application or a desktop graphical application. As many students felt uneasy with command-line tools, GitHub was introduced through the graphical user interface; however, at the time of the course, this application was difficult to set-up on a network of lab computers, poorly documented by its creators, somewhat buggy, and not very forgiving of beginner mistakes. Additionally, GitHub uses a different workflow than other file-sharing software, in which software developers copy or 'clone' a repository of source files, make changes to the project files on their own computer, 'commit' those changes to the cloned repository, then 'merge' those changes with the remote repository stored online. This workflow was difficult to grasp for students who were used to the drag-and-drop interfaces of Windows and Macintosh operating systems. Additionally, some students experienced difficulty reconciling conflicting file versions when they occurred in the course of collaborative work on the final project.

In contrast to GitHub, the introductory D3 lessons presented in Week 6 were among the most successful teaching experiences reported in the log. The "D3 Core Selectors and Generator Functions" lesson was divided into separate conceptual chunks or "best practices" that were introduced clearly and sequentially and built on data concepts covered in prior lessons. This segmented approach allowed students to focus on manageable steps toward mastering the D3 library and integrating it with other Open Web Platform technologies, skills that were major hurdles in the 2013 offering of the course.

During Weeks 7-9, it became evident that while students' success with learning D3 overall exceeded expectations, students did experience challenges with three specific components of the D3 lab assignment. First, a number of students had difficulty making use of the TopoJSON data format, mostly due to unanticipated crashing of the online file conversion software they were directed to use—a risk of using very new open source tools that are not yet well-established. Second, D3's implementation of geographic projection parameters—covered in Week 7's "D3 Geography" lesson—proved the most conceptually difficult aspect of D3 and required extra review in Week 9. Finally, a number of students experienced difficulty debugging script errors. However, students' confidence in their coding abilities seemed to soar during the D3 unit; as recorded in the instructor log, "the attitude generally seems to be, 'I'm learning and know I'll get beyond this' rather than helplessness or giving up."

4.5.2 Student Feedback Compositions

The 28 student feedback compositions that were turned in with the two lab assignments gave a glimpse into students' perspectives on their learning processes. The results below are ordered by the four categories of statements described in Section 4.3—1) difficult concepts, 2) non-conceptual problems, 3) 'aha!' moments, and 4) other positive experiences (Table 4.2). Tables 4.3-4.6 show the most common themes and their frequencies for each lab assignment. Some of these themes are highlighted in the description below.

Table 4.2: Total number of student feedback themes in each category.

Category	Leaflet	D3	Total
Difficult Concepts	23	31	45
Other Problems	21	19	33
Aha! Moments	12	14	25
Other Positive Experiences	12	17	26

Students reported 45 themes regarding difficult concepts, with 19 of these mentioned in two or more essays (Table 4.3). The most extensive themes were *difficulty extending the Leaflet lab product beyond the example code provided in the assignment tutorial* and *attempting to add a Leaflet plug-in to accomplish a task not included in the provided example code.* The D3 assignment compositions included fewer conceptual difficulties, with the most common *using script to manipulate and combine data objects*—a concept that is related to JavaScript and computational thinking but not to D3 itself. One student summed up the problem of *distinguishing which API certain methods belong to* as, "I didn't fully understand when I was using a Javascript, Jquery or Leaflet function. I understand the difference between these things for the most part, but maintaining proper syntax throughout all of them is extremely difficult, especially when you are also including html and css" (original formatting).

Table 4.3: Extensiveness of difficult concept themes expressed by multiple students.

Difficult Concept Theme	Leaflet	D3	Total
Implementing a custom interaction operator	7	0	7
Using correct code syntax	3	2	5
JSON data manipulation	3	1	4
Debugging	3	1	4
Joining data objects using loops in the script	0	4	4
Implementing a Leaflet plugin	3	0	3
Distinguishing which API certain methods belonged to	2	1	3
Understanding what parts of lab examples to change	2	1	3
Incorporating online examples into custom code	2	1	3
Adding a dynamic legend to the map	2	0	2
Creatively solving problems	2	0	2
Improving the efficiency of the code	2	0	2
Understanding JavaScript	2	0	2
Understanding object-oriented programming	1	1	2
Positioning interface elements with CSS	1	1	2
Dynamically updating interface elements	1	1	2
Coordinating interactions across two data visualizations	0	2	2
Implementing dynamic text wrapping in HTML	0	2	2
Rescaling a bar chart to visualize a new attribute	0	2	2

Students reported 33 themes regarding other problems that were not necessarily conceptual in nature (Table 4.4). By far the most extensive theme was the *presence of unresolved issues with the map display at the time of lab submission*, although the display issues varied widely in their specific nature. Otherwise, the most common problem themes during the Leaflet assignment were *the choice of a dataset that did not fit the assignment scenario*, a feeling of *frustration with the student's own lack of understanding,* and the feeling of *not having enough practice with coding to complete the assignment successfully.*

Table 4.4: Extensiveness of other problem themes expressed by multiple students.

Other Problem Theme	Leaflet	D3	Total
I have unresolved display issues	3	5	8
I had trouble converting data to TopoJSON	0	5	5
I was frustrated with my lack of understanding	3	1	4
The dataset I selected didn't fit the assignment scenario	3	1	4
I needed more practice	3	0	3
I had problems with time management	2	1	3
I was frustrated with my lack of success	2	1	3
The data I wanted wasn't available	1	2	3
There were cross-platform differences in my map's appearance	2	0	2
I was intimidated/experienced a lack of confidence	2	0	2
I lack adequate HTML and/or CSS skills	1	1	2
I needed further instruction on the TopoJSON command line tool	0	2	2
It was easy to get "stuck" on errors	0	2	2

Table 4.5: Extensiveness of 'aha!' moment themes expressed by multiple students.

Aha! Moment Theme	Leaflet	D3	Total
Using the browser console helped to find problems	0	3	3
The order of execution in the script became clearer	2	1	3
I had success using online plugins and examples	3	0	3
Working through errors created 'Aha' moments	0	2	2
Neat code formatting aids understanding/debugging	0	2	2
Asking others for help overcame blockages	0	2	2
I had success using jQuery methods	2	0	2

Students reported 24 themes related to 'aha!' moments (Table 4.5). The most extensive themes included *discovering how to track the order of execution in the script, success using online plug-ins and examples,* and *learning how to test script in the browser console to solve problems in the code.* Four students reported that *working through errors independently, with a peer, or with the instructor* generated 'aha!' moments, showing the importance of active learning and collaboration. One student reflected, "I was getting overwhelmed by the enormity of trying to solve huge problems. I needed to break it down and solve things one at a time, not all at once." The 'aha!' moments in the Leaflet compositions demonstrated more conceptually basic realizations such as how to use a particular library method or perform a scripting task, whereas many of the D3 compositions reflected higher-level creative problem-solving insights.

Table 4.6: Extensiveness of other positive experience themes expressed by multiple students.

Other Positive Experience Theme	**Leaflet**	**D3**	**Total**
The lab assignment tutorial was clear	2	3	5
I am now more comfortable with JavaScript	2	2	4
I learned a lot	1	2	3
Having example code provided helped	2	0	2
Instructor overview of code helped	2	0	2
I understood D3 better than Leaflet	0	2	2
I can make use of techniques I learned in the final project	0	2	2
I was able to debug independently	0	2	2
I like the visual appearance of my D3 map	0	2	2

Students reported 26 themes regarding other positive experiences that were not attached to a specific 'aha!' moment (Table 4.6). The most common theme was an *appreciation for the clarity and helpfulness of the lab assignment tutorials.* Four students (two for each assignment) reported that they felt more confident using JavaScript than they had before starting the assignment. Multiple students remarked that they *liked and understood D3 better than Leaflet,* which seems to run counter to the experience of the 2013 course, and points to the success of the prior lessons at building the scaffold necessary for understanding D3.

4.5.3 Exit Survey

As described in Section 4.3, the exit survey included sections on: 1) understanding of Open Web Platform tools, 2) difficulty of learning the tools, 3) the sequence of lesson topics, 4) learning resources, and 5) lab assignments. To analyze the first section of the exit survey, the means of students' beginning-of-course expertise ratings were compared to their end-of-course ratings for each of 20 Open Web Platform technologies using unpaired t-tests (Table 4.7). Percentage change of mean for each technology is shown in the rightmost column of Table 4.7 and color coded by interpolating between the ColorBrewer Blues color scale extremes using a 0.5-power scale to enhance difference perception (Harrower and Brewer, 2003; Stevens, 1957). D3 was used to create the interpolation. Differences that were not statistically significant at the 95% confidence level are shown in brackets ([]) within uncolored cells.

Students' ratings for all but one of the technologies covered by the curriculum exhibited a significant increase at the 95% confidence level, mostly from low to moderate levels. The one covered technology that students did not significantly gain proficiency with, the CSV data format, most likely was familiar to students from prior GIS courses, as its initial mean expertise was moderate (4.6 out of 7). The lowest mean expertise as a final outcome in this group of tools was for AJAX (2.9), indicating that students still found this a difficult technology to use even after grappling with it extensively. In contrast, for the six technologies that were *not* explicitly covered by the curriculum, there were only slight increases in mean expertise, and all final means fell below 3/7. Only one of these technologies, Mapbox Studio/TileMill (the name changed with a new version released during the course), had an increase that was statistically significant ($p = 0.01$); this may have been due to some students choosing to use this tool independently to complete their final projects.

Table 4.7: Student-reported increase in expertise with Open Web Platform mapping tools. Percentages shown in brackets are not statistically significant at the 95% confidence level. Cell color values are interpolated on a square root scale from 41% to 225%.

Open Web Platform technology	Mean initial expertise	Standard deviation initial expertise	Mean final expertise	Standard deviation final expertise	Mean Expertise percent change
Tools covered in lab curriculum					
HTML	3.26	1.76	4.61	1.27	41.4
CSS	2.87	1.52	4.30	1.33	49.8
JavaScript	2.30	1.36	4.04	1.33	75.7
jQuery	1.83	1.27	3.43	1.24	87.4
AJAX	1.52	0.99	2.87	1.18	88.8
The DOM	1.74	1.21	3.30	1.18	89.7
Git/GitHub	1.74	1.36	3.61	1.64	107.5
Leaflet	1.65	1.27	4.17	1.11	152.7
D3	1.17	0.65	3.78	1.24	223.1
JSON	2.17	1.53	4.48	1.27	106.5
GeoJSON	1.91	1.53	4.43	1.31	131.9
TopoJSON	1.61	1.41	4.39	1.31	172.7
SVG	2.04	1.52	3.91	1.50	91.7
CSV	4.57	1.56	5.17	1.40	[13.1]
Tools **not** covered in lab curriculum					
Google Maps API	1.78	1.00	2.26	1.29	[27.0]
OpenLayers	1.35	0.93	1.87	1.32	[38.5]
ArcGIS Online	1.61	1.27	2.04	1.49	[26.7]
Mapbox Studio/TileMill	1.96	1.33	2.96	1.22	51.0
CartoDB	1.43	0.95	1.91	1.35	[33.6]
KML	2.57	1.50	2.96	1.69	[15.2]

In the second section of the survey—assessing the difficulty of learning each tool—D3 was rated the most difficult tool to learn, with a mean difficulty rating of 5.2 out of 7. It also had the lowest standard deviation (1.17), indicating most students rated it very difficult (Table 4.8). However, these difficulty ratings do not necessarily indicate how positively students viewed their learning experiences with each tool. When asked to rate their overall emotional experience with the lab assignments and final project, the mean positivity rating rose for each successive assignment. The mean positivity rating for the D3 lab assignment was higher than that of the Leaflet assignment (5.2 versus 4.8), and all students rated their experience with D3 a 3 out of 7 or better (Figure 4.2). These results indicate that students found D3 challenging yet rewarding, and echo the improvement of self-confidence recorded in the instructor log.

Table 4.8: Students' mean difficulty ratings for Open Web Platform mapping tools, from most to least difficult.

Technology	Mean difficulty	Standard Deviation
D3	5.22	1.17
JavaScript	4.52	1.83
AJAX	4.43	1.41
jQuery	4.13	1.49
Leaflet	4.13	1.32
The DOM	3.74	1.86
Git/GitHub	3.74	1.54
CSS	3.48	1.68
SVG	3.17	1.44
TopoJSON	2.83	1.44
HTML	2.78	1.57
GeoJSON	2.70	1.49
JSON	2.48	1.44

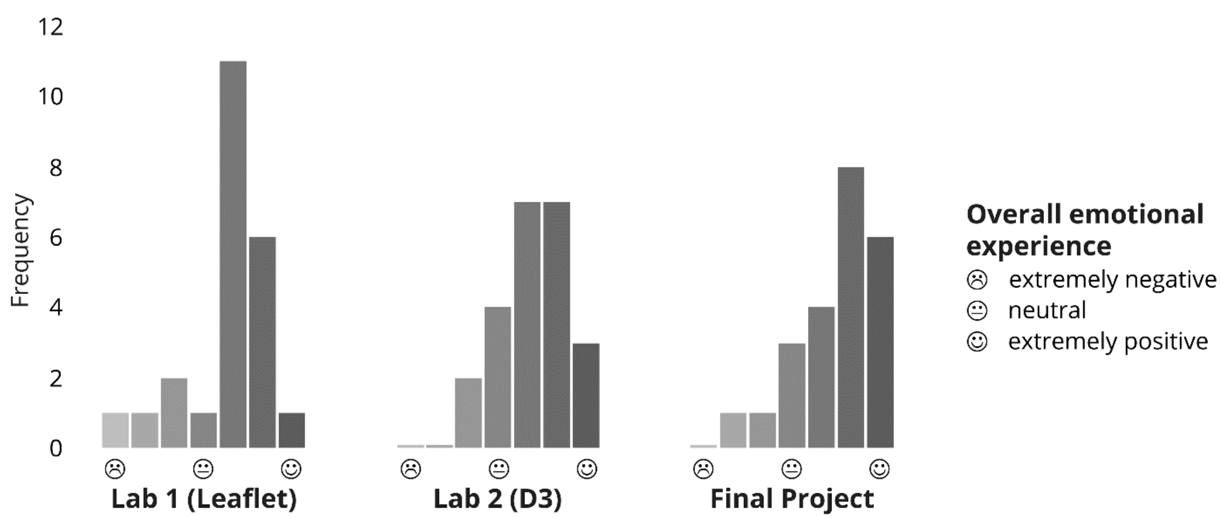


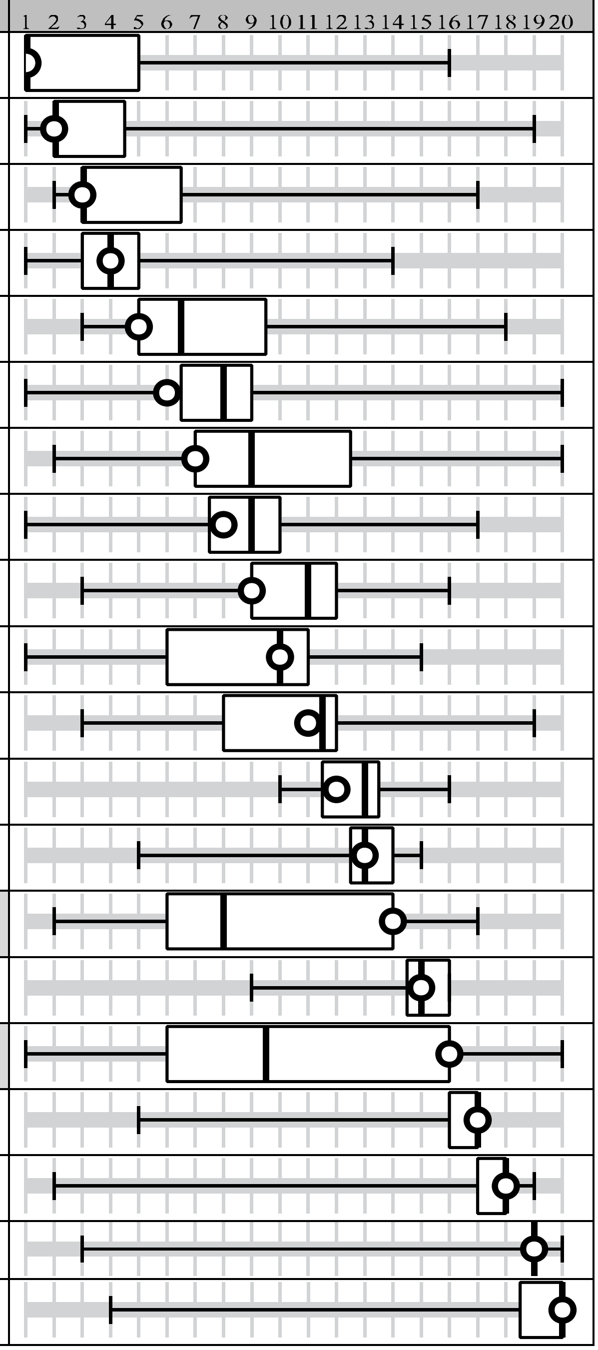
Figure 4.2 Students' overall emotional experiences while completing each lab assignment and the final project.

In the third section of the survey, students were asked to reorder the curriculum topics in a way that made most sense to them, and to comment on which topics needed more reinforcement or were not useful. The median positions indicated that *using in-browser developer tools to debug* and *basics of GitHub* should come earlier in the sequence (Table 4.9).

Six students found the *coordinate systems* lesson unhelpful and five found the *data levels and types* lesson unhelpful, largely due to the redundancy of these topics with prior Cartography and GIS courses.

Table 4.9: Topic sequence with order means from exit survey. Highlighted rows show topics that came significantly later in the sequence than students would have preferred.

Topic	Mean/ St. Dev.	Topic order as taught (circles) vs. student suggestions (boxplots)
1. Local website directory setup	3.7/4.3	
2. Basics of HTML	3.8/4.0	
3. Basics of CSS	4.8/3.4	
4. Basics of JavaScript	4.7/3.0	
5. Basics of jQuery	7.8/4.2	
6. Data levels and types	8.7/4.9	
7. Geographic coordinate systems	10.5/5.1	
8. Data format specifications	8.3/3.5	
9. Asynchronous JavaScript and XML (AJAX)	10.5/2.8	
10. Using online reference documentation	8.7/3.6	
11. Using search engines and online resources	10.4/3.7	
12. Slippy map tile concepts	12.6/1.5	
13. Basics of Leaflet	12.6/2.9	
14. Using in-browser developer tools to debug	9.6/4.7	
15. Leaflet custom UI elements and interactions	14.7/1.7	
16. Basics of GitHub	10.6/6.2	
17. SVG elements and attributes	15.7/3.0	
18. D3 selections	16.7/3.7	
19. D3 generator functions and scales	17.8/3.9	
20. D3 projections and path generators	18.3/4.0	



In the fourth section of the survey—covering learning resources—six students indicated that instructor-led coding demonstrations in which the instructor worked through a coding problem with students observing and following along were not useful or were difficult to follow. This important observation was missed by the instructor logs and student essays. In the fifth section—covering the lab assignments and final project—21 students (91%) agreed that they "knew how to make an effective and well-designed web map using Leaflet" after completing the first lab (mean agreement of 5.2/7). 16 students (69%) agreed with the same statement for D3 (mean agreement of 4.8/7), which is in line with the rating of D3 as the harder of the two tools to learn. Four students said they wanted more time to complete the final project, and five indicated that they had been challenged by various aspects of collaboration with their peers.

4.6 Discussion: Threshold Concepts and Learning Outcomes

The curriculum evaluation discussed above provides insight into the second research question of the dissertation, *What skill-based learning outcomes for open web mapping are achievable in a one-semester upper-level undergraduate Geography course?* To answer this question, it is first necessary to examine the specific threshold concepts that students experienced while learning web mapping (Table 4.10).

As indicated by the instructor log and student feedback submissions, early in the course many students experienced difficulty understanding the *Document Object Model (DOM), basic JavaScript syntax,* and *Asynchronous JavaScript and XML (AJAX).* These concepts are necessary to grasp in order to successfully implement an interactive map on the Open Web Platform. AJAX and JavaScript syntax in particular kept cropping up as challenges in student feedback (see Table 4.1), and were both rated as quite difficult on the exit survey (see Table 4.8), indicating a need to refine and expand the lessons on those topics. Similarly, understanding *the relationship* *between native JavaScript and API methods* was observed to be difficult for some students in the early weeks of the course and continued to be mentioned by student as challenges for both lab assignments. These four threshold concepts represent *computational thinking* skills: visualizing an abstract set of computational objects, becoming accustomed to the strictness of programming syntax, following the non-linear flow of execution in a program, and recognizing where to apply different methods, respectively (see Section 2.4).

Table 4.10: Threshold concepts encountered by students, general to web mapping and specific to certain tools used in lab.

General Web Mapping Concepts
The Document Object Model (DOM)
Basic JavaScript syntax
Asynchronous JavaScript and XML (AJAX)
The relationship between native JavaScript and API methods
Integrating example code into the code base
Debugging
Software-specific Concepts
Implementing a custom interaction operator in Leaflet
D3 geographic projection parameters
GitHub version control workflow

Understanding how APIs relate to one another and to native JavaScript is also required to be able to integrate multiple code libraries and plugins that can be used together to produce a desired solution. Many students also experienced difficulty *integrating code from examples into their code base,* whether those examples were from lab handouts or found online. Such code integration requires understanding what parts of the example code are relevant to the task being attempted and which should be altered or discarded. Given the complex array of open-source technologies that is a key feature of the Open Web Platform discussed in Section 2.1, integration of multiple APIs and examples, or *confluence,* represents an important area of understanding for web mapping.

Certain challenges that students experienced stood out as specific to particular software tools. Many students had difficulty *implementing a custom interaction operator* of their choosing into their Leaflet lab assignment. Doing so required leveraging the Leaflet API documentation to build custom code from scratch and/or integrating one of the many Leaflet plugin libraries available. Likewise, particular parts of the D3 API were difficult to grasp, most notably its implementation of *geographic projection parameters*, and D3 overall was rated very difficult to learn in the exit survey. Finally, many students struggled with the *GitHub version control workflow,* and some dropped the use of GitHub for the final project altogether. These difficulties indicated the need for more robust and better structured lessons, and GitHub in particular needed to be introduced much earlier (see Table 4.9). These three challenges are also indicative of a broader skillset that is required to build custom interactive web maps: *competence* with each of the individual software tools used to create the web map.

Debugging was observed in the instructor log as a key challenge for students throughout the course. Debugging is unquestionably a key part of the development process, as all software contains bugs—that is, behaviors that the developer does not want—and many bugs are created during the software development process (Telles and Hsieh, 2001). The sequencing preferences indicated in the exit survey (Table 4.9) reinforced the need to better integrate the use of browser developer tools—the primary toolset for debugging web maps—into the initial unit. Introducing these tools earlier could have enabled students to better visualize and grasp the difficult concepts associated with the DOM, syntax use, and AJAX as those concepts were introduced, as well as saving debugging headaches encountered during the Leaflet lab. Students indicated in their feedback that once they did learn these tools, they generated numerous 'aha!' moments.

These 'aha!' moments provide evidence of another key cognitive development necessary to successfully build a web map: *confidence* in one's own coding capabilities and ability to learn new thought processes. Anecdotal evidence of students' growth in confidence through the course was provided by the two statements quoted in the results: from the instructor log, "the attitude generally seems to be, 'I'm learning and know I'll get beyond this' rather than helplessness or giving up;" and from the student feedback compositions, "I was getting overwhelmed by the enormity of trying to solve huge problems. I needed to break it down and solve things one at a time, not all at once." These statements were exemplary of other student feedback. Further evidence of the growth in confidence was provided in the exit survey, by the increasing positivity of students' emotional experiences (see Figure 4.2) and by the overwhelming majority of students agreeing that at the end of the course they "knew how to make an effective and well-designed web map" with Leaflet or D3. Perhaps the best evidence for confidence as a leading requirement of web mapping success is an unfortunate negative example: one student (who did not take the exit survey) made statements throughout the course reflecting a feeling of not being good at coding and not being able to understand the process. That student failed to produce working products for either lab assignment or the final project, and was given an alternative writing assignment to complete the course grade.

Table 4.11: Four learning outcomes for a web mapping course using Open Web Platform technologies.

Learning Outcome	**Definition**
Computational Thinking	The ability to follow the flow of execution in computer program code, visualize data objects and processing techniques, and decompose large computing tasks into a series of smaller steps.
Competence	The ability to apply Open Web Platform mapping tools successfully across a range of contexts.
Confluence	The ability to analyze how multiple data, representation, and interaction technologies integrate to produce a final product.
Confidence	The ability to evaluate one's achievements positively and trust one's ability to overcome obstacles and improve at difficult web mapping tasks.

In summary, the threshold concepts and 'aha!' moments experienced by students helped to illuminate the key over-arching cognitive tasks required to make a custom interactive map on the Open Web Platform, regardless of the specific technologies that are used. These tasks, defined in Table 4.11, may be considered desired learning outcomes for any web mapping course, addressing the research question above.

Even before implementing the new curriculum, it was evident that the use of Open Web Platform tools requires *computational thinking,* or the ability to 'think like a computer' (Raja, 2014; see Section 2.4). The curriculum evaluation provided ample evidence of the need for computational thinking and revealed the importance of the latter three outcomes. Developers must build *competence* in applying the tools of the Open Web Platform through using them and getting to know their reference documentation. Interactive web maps often require the use of multiple tools that accomplish different tasks, so students need to analyze how these tools work together to integrate them in an effective *confluence* that results in a unified solution. Finally, the web mapper must have *confidence* that they can learn new tricks to overcome obstacles; such confidence is built in students through learning, encouragement, and eventual success. These processes are non-linear and iterative; each new discovery of a solution builds *computational thinking, competence, confluence,* and *confidence.*

The threshold concepts and learning outcomes identified through the 2014 curriculum evaluation led to revisions in the curriculum for the next iteration of the Interactive Cartography and Geovisualization course, taught in the Spring of 2016. Additionally, the curriculum was modularized for delivery through an online learning management system to meet the instructional needs of the program for blended and fully online courses. With these revisions came the need to further evaluate the curriculum to determine its efficacy in the online delivery format. The evaluation of the blended curriculum is reported in Chapter 5.

V. Blending the Curriculum

Abstract

This chapter presents a major revision of the curriculum described in Chapter 4, conducted to both better address the threshold concepts identified in that chapter and modularize the curriculum for online delivery in distance education and blended instructional settings. It reports on evaluations conducted over two semesters of the blended version of the curriculum. Section 5.1 describes the process of modularization undertaken to enable delivery of the lab curriculum in both in-person and distance settings. Section 5.2 overviews the changes that were made to the topic scope and sequence of the lab curriculum during the process of modularization. Section 5.3 discusses the qualitative methods used to evaluate the modular curriculum, including how methods differed from the 2014 curriculum evaluation. Section 5.4 presents the results of the 2016 and 2017 curriculum evaluations and compares those results to the outcomes discovered in 2014. Finally, Section 5.5 discusses the extent to which the modular curriculum succeeded in producing computational thinking, competence, confluence, and confidence outcomes, particularly in comparison with the 2014 curriculum. It addresses Research Question 3, *How does student achievement of the identified learning outcomes for web mapping compare between fully in-person and modular, blended instruction?*

5.1 The Problem of Online Learning

Section 4.1 discussed changes that are occurring in the way higher education delivers its product to students, including the recent growth of online GIScience programs (NCER, 2014; Allen and Seaman, 2013; Allen et al., 2016; Luo et al., 2014, Robinson et al., 2015). This growth trend spurred the creation of an Online Masters in GIS and Web Map Programming degree program at UW–Madison in 2016. The Interactive Cartography and Geovisualization course (Geography 575) is a required course for the Online Master's program, recommended for completion in the second enrolled semester after completing one cartographic design and one GIS programming course. To enable dual support of both online and in-person versions of the course, the in-person course was shifted to a blended format, with the course lab material delivered as online instructional modules that could also be used in the fully online course.

Dual support of both in-person blended and online formats required significant changes to the structure of the course's lab component. As per recommendations presented in Table 2.5, the curriculum needed to be made mostly asynchronous, allowing students to work through lessons at their own pace without set class meeting times (Crawford-Ferre and Wiest, 2012; Mundkur and Ellickson, 2012). Weekly topics from the 2014 curriculum (see Table 4.1) were transformed into ten instructional modules, each contained within a separate web page and consisting of 2-4 topical lessons each (Singh, 2003). The modules were posted to the Canvas LMS for delivery. The goal of modularization was to create a typical blended learning experience (see Section 2.5) with comparable learning outcomes to the entirely in-person format. Classes continued to meet in person for lecture and lab periods, but students were expected to read each week's module and complete as much of the module on their own as possible leading up to the weekly lab period.

During modularization, the topic scope and sequence were adjusted based on the results of the curriculum evaluation presented in Chapter 4. The final LMS-based lab curriculum consisted of ten modules, each contained within a separate web page, plus the final project. Modules were organized into four units: 1) an initial Workflows and Data unit, 2) a unit focused around the Leaflet lab assignment, 3) a unit for the D3 lab assignment, and 4) a final unit for the Final Project (Table 5.1). Most of the step-by-step directions for each of the major lab assignments described in Section 4.2 were moved into the module pages to break up the otherwise lengthy assignment instructions.

Each module page began with a brief Introduction section with an overview of the module's content and a list of 3-4 module objectives (Figure 5.1). This was followed by 2-4 lessons, each containing multiple sub-sections. Module content was formatted as a mix of step-by-step tutorial and conceptual explanation, with interspersed paragraph text, copiable example code blocks, conceptual figures, screenshots displaying the results of each example block, and numerous hyperlinks to outside resources (Figure 5.2). Various fonts and text colors were used to highlight keywords, code, links, and filenames in different ways. Humor and anecdotes were used to retain a light, personable tone, keeping learning fun in an online environment.

Table 5.1: Interactive Cartography and Geovisualization online lab units and modules

Unit 1: Workflows and Data
Module 1: Setting Up Your Workspace
Module 2: Scripting and Debugging
Module 3: Data and AJAX
Unit 2: Programming with Leaflet
Leaflet lab assignment
Module 4: Using Online Resources
Module 5: Leaflet Interactions
Module 6: The Internal Logic of Leaflet
Unit 3: Designing with D3
D3 lab assignment
Module 7: D3 Foundations
Module 8: Mapping in D3
Module 9: Coordinated Visualizations
Module 10: Coordinated Interactions
Unit 4: Final Project
Final project guidelines

At the end of each lesson, an ungraded "Self-Check" quiz was provided to allow students to test their understanding of the lesson's content; each quiz included one multiple choice and one true/false question (Figure 5.3). Each module also contained a set of activity directions displayed as blockquotes at various points throughout the module and reiterated at the bottom of the page as "Module Deliverables" that students were expected to submit to the instructor (Figure 5.3). Except for the first module, student submissions were made via commits (commented file backups) to each student's GitHub repository for the unit. The Quizzes and Module Deliverables were linked to separate quiz and assignment pages within the learning management system. Appendix 3 contains Module 1 in its entirety to demonstrate the design of a complete lab module.

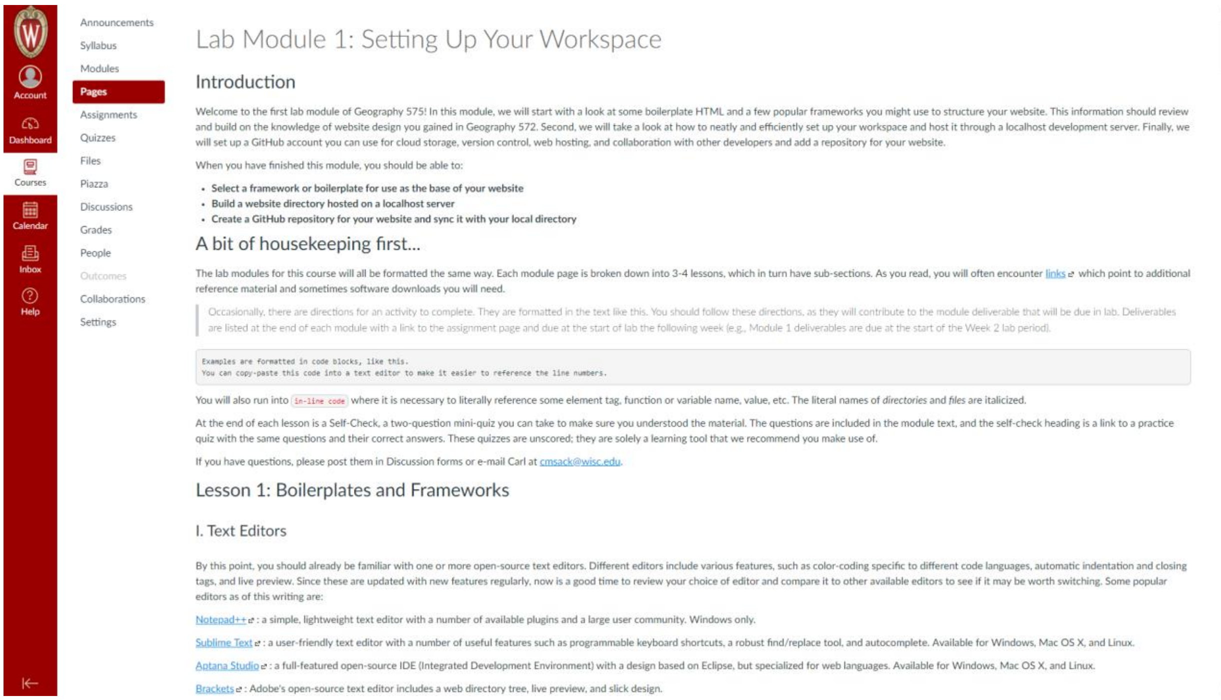


Figure 5.1: A screenshot showing the beginning of the first online lab module within the Interactive Cartography and Geovisualization course's learning management system.

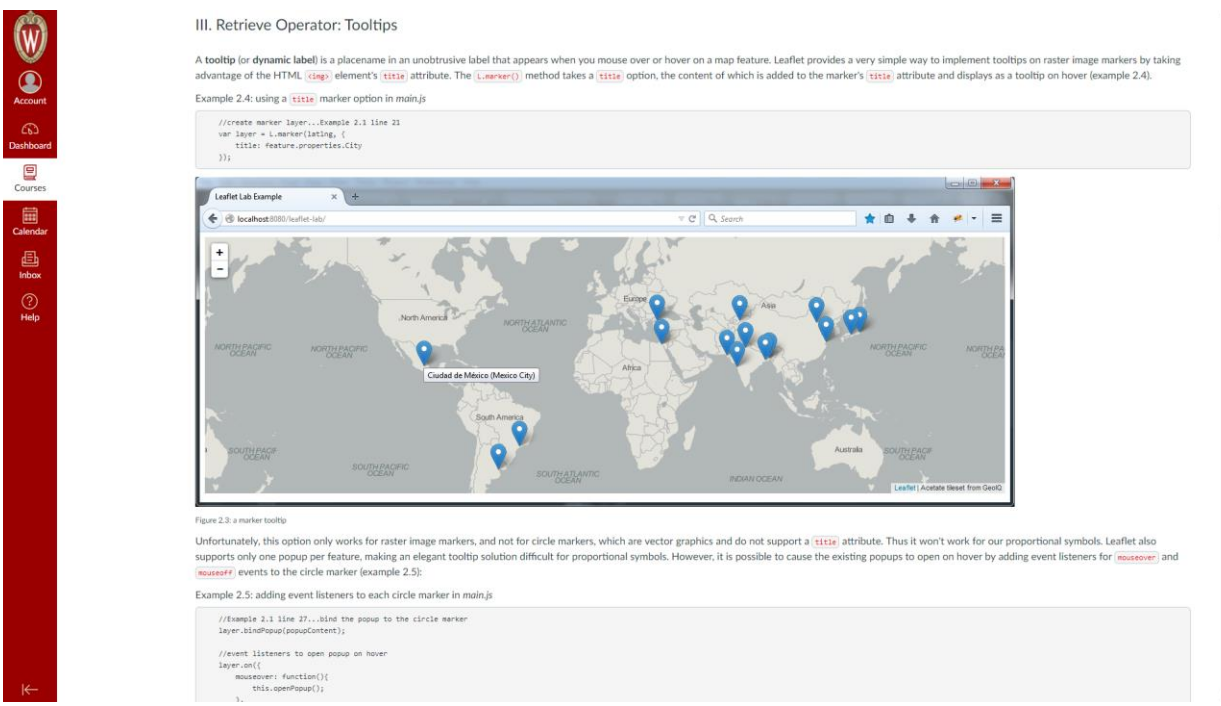


Figure 5.2: A screenshot showing example code blocks and graphics in Module 5 (Leaflet Interactions).

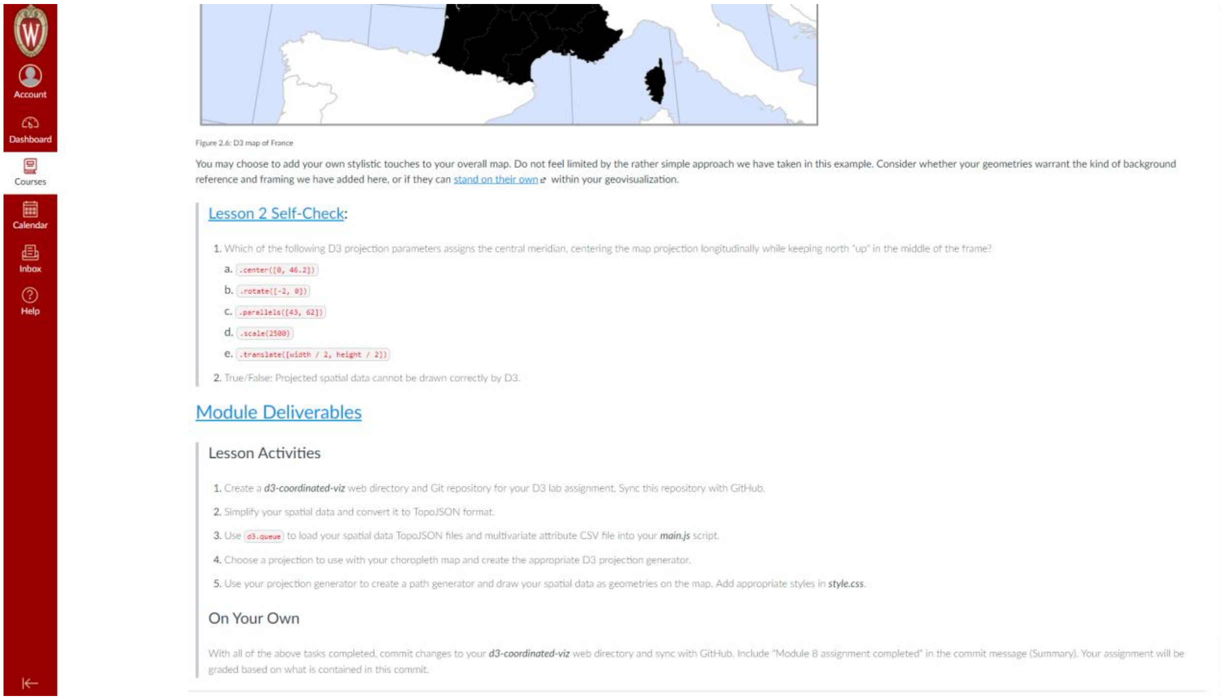


Figure 5.3: A screenshot showing a lesson Self-Check quiz and set of Module Deliverables for Module 8 (Mapping in D3). The Self-Check title is a hyperlink that directs users to an ungraded quiz, allowing students to see the correct answer after they have taken it. The Module Deliverables title is a hyperlink to a separate assignment page with the same list of directions.

Before the course was offered fully online for the first time, one semester of the course (Spring, 2016) was offered in the blended format. At the time of writing, the modular lab curriculum remains in use for both the fully online and in-person blended versions; these have been offered concurrently in Spring semesters since 2017. The fully online version of the course offered in the Spring of 2017 could not be adequately evaluated due to problems that arose with accessing students in the course for research purposes. This chapter presents evaluations conducted during the 2016 and 2017 blended course offerings, addressing RQ3, *How does student achievement of the identified learning outcomes for web mapping compare between fully in-person and modular, blended instruction?*

5.2 Curriculum Scope and Sequence Adjustments

During the process of modularizing the lab curriculum, adjustments were made to the depth of certain topics as well as the topic sequence based on the threshold concepts and learning outcomes identified by the evaluation described in Chapter 4. Table 5.2 shows the revised topic sequence and highlights lessons that were expanded upon and/or moved.

Table 5.2. The revised 2016-17 topic sequence. Topics shown in italics *were significantly expanded to better address threshold concepts identified by the curriculum evaluation. Underlined topics were moved to a different point in the topic sequence due to student feedback.*

Unit 1: Workflows and Data	Unit 2: Programming with Leaflet	Unit 3: Designing with D3	Unit 4: FP
Module 1: Setting Up Your Workspace Boilerplates and Frameworks Web Directory Setup *GitHub Setup (moved from Week 6)* Assigned: JavaScript Online Tutorial	**Module 4: Using Online Resources** *Leaflet Tutorials and API* *Using Online Examples* *Using Help Forums* Finding Tilesets and Data	**Module 7: D3 Foundations** D3 Selections and Blocks Data Scales, Axes, Text	**Final Project**
Module 2: Scripting and Debugging *Exploring the DOM* *JavaScript Basics* *jQuery Basics* *Debugging in the Developer Console (moved from Weeks 4 and 9)*	**Module 5: Leaflet Interactions** Making Leaflet Layers Dynamic Zoom, Pan, and Retrieve Interactions Sequence Interaction Additional Interaction Operators	**Module 8: Mapping in D3** *D3 Helpers: TopoJSON, MapShaper & Queue* *D3 Projections and Path Generators*	
Module 3: Data and AJAX CSV, XML, and JSON formats and their geographic variants *AJAX Concepts and Syntax* *AJAX Callback Functions*	**Module 6: The Internal Logic of Leaflet** Object-oriented JavaScript Extending Leaflet Objects Using SVG Graphics	**Module 9: Coordinated Visualizations** Dynamic Map Styling Drawing a Coordinated Visualization	
		Module 10: Coordinated Interactions Dynamic Attribute Selection Transitions Linking Interactions Between Map and Chart *Deploying Your Geovisualization (moved from Week 6)*	

The *Basics of HTML* and *Basics of CSS* lessons were removed from the curriculum due to their inclusion in another cartographic design course and the availability of remedial online training. Per student feedback in 2014 (see Chapter 4), *Data Levels and Types* and *Geographic Coordinate Systems* were likewise removed as redundant with other courses. To address the sequencing issues highlighted in Table 4.9, the initial GitHub lesson was moved to Module 1 (consistent with the first week of the course), and the lesson on using the browser's developer tools to debug was moved to Module 2. GitHub was also adopted as a platform for submission of student work throughout the course. However, the web hosting component of GitHub was separated from the version control system component and moved to the end of the topic sequence, better matching the web mapping workflow (see Section 2.3).

Several new topics in Table 5.2 represent subdivisions of prior topics meant to break down threshold concepts into more readily learnable chunks. A new "Exploring the DOM" lesson was created to more thoroughly introduce students to the concept of the Document Object Model. The AJAX lesson was separated into two lessons dealing first with AJAX concepts, then with its application. The reference documentation lesson was given more structure and refocused to take advantage of the easy-to-use Leaflet documentation and tutorials, while the lesson on forums and online examples was broken into separate lessons on examples and forums. Finally, three topics that needed extra review during the 2014 course—TopoJSON, D3 projections, and debugging—were reinforced in the revised curriculum, with TopoJSON included in a new lesson on "D3 Helpers" and the latter two topics given their own dedicated lessons. The final project was maintained as a capstone assignment taking up the last few weeks of the course.

In addition to scope and sequencing issues, the new course format provided opportunities to address other pedagogical concerns raised in Chapter 4. Rather than requiring one complete lab assignment submission at the end of a multi-week period as the sole lab submission for the unit, each week's module included smaller checkpoint activities that built toward the final deliverable. These submissions were intended to provide students with more manageable building blocks toward each larger lab assignment, thus building student confidence, and allow instructors more frequent assessment of student progress, better supporting scaffolding (Palincsar, 1986). Also, feedback from students during the 2014 evaluation indicated that coding demonstrations by an instructor were difficult to follow and comprehend. The modules afforded the opportunity to replace instructor-led demonstrations with written code examples that students could copy and paste into their own work to follow the lesson progress, providing a more hands-on experience and better scaffolding toward mastery of JavaScript.

To test the efficacy of these curriculum revisions, curriculum evaluations were conducted during both the 2016 and 2017 blended offerings of the course. The modular curriculum remained substantively identical between the two semesters, although the lab instructor differed between semesters. The evaluation methods for the 2016 and 2017 classes are discussed in the next section.

5.3 Evaluation Methods

Lessons learned during the 2014 curriculum evaluation were applied to the methods used to evaluate the 2016 and 2017 blended offerings of the course. As described in Section 4.3, the 2014 evaluation instruments that ultimately were analyzed consisted of a weekly instructor observation log, student feedback compositions collected along with submission of the Leaflet and D3 lab assignments, and an exit survey. For the 2016 and 2017 evaluations, an entrance survey was added for comparison to the exit survey. The entrance survey questions are included as Appendix 4. Both surveys collected student identifiers, allowing for cross-referencing individuals' results between the two surveys. Both surveys also asked students to rate their expertise prior to the class with a range of web development tools, thus allowing comparison of students' estimations of their prior knowledge before and after the class. The entrance survey also included demographic questions such as gender, race, nationality, and academic level, as well as the number of prior courses taken in programming, web technologies, and GIS. Finally, the entrance survey gave students the option to state why they were taking the course and what they hoped to gain from it.

The 2014 exit survey was used as a model for the 2016 and 2017 exit surveys. However, groups of questions in the latter surveys were more explicitly tailored to gauge success with the overall learning outcomes for web mapping identified in 2014: *computational thinking, competence, confluence,* and *confidence* (see Table 4.11). New sets of questions also were added to the 2016 and 2017 exit surveys to directly assess changes in students' computational thinking and confluence skills. These two outcomes were each subdivided into three tasks: for computational thinking, understanding program order of execution, writing correct syntax, and breaking down problems; and for confluence, identifying methods that are made available by different code libraries, integrating multiple code libraries, and executing the entire web mapping workflow. Each task was assessed using one positively-worded and one negatively-worded question.

As in 2014, competence was gauged using before- and after-course expertise ratings for the set of web mapping tools introduced during the course lab and for a control set of tools not covered by the lab. Students then were asked to rate how challenging they found each tool used in the course, the level of challenge of each lab module, and how fun each module was to complete. Each of the ratings used a five-point Likert scale, replacing the seven-point scales used in the 2014 survey to simplify student responses and make completing the survey less time consuming. To normalize the two scales for direct comparison, a percent change score was calculated for each tool's ratings. As in 2014, students were asked to reorder the topics covered by the lab modules (numbering 31 in the modular curriculum instead of 20), give written feedback on those topics, and rate the effectiveness of different resources. Finally, the survey included questions to gauge student confidence. For each lab unit, students were asked to rate their overall emotional experience, how much time they spent on the modules (in relative terms), what 'aha!' moments and frustrations they experienced, and how difficult they would find it to complete the unit objectives on their own after taking the course. Exit survey questions are included as Appendix 5.

An instructor log was not feasible in 2016 because the researcher was not the lab instructor and could not ask the lab instructor for extra uncompensated labor or control for differences in observational technique. However, rather than collect only two student feedback compositions during the semester, students were asked to volunteer feedback on their frustrations and 'aha!' moments along with each week's module deliverables. Encouraging weekly student feedback added the pedagogical benefit of increased metacognition, or students thinking about their learning to discover their threshold concepts (Fouberg, 2013). Mean student scores on each weekly submission also were tracked and analyzed, providing a more objective assessment of each module's level of difficulty.

5.4 Results

5.4.1 Entrance Survey

The entrance survey revealed several relevant characteristics of each class, as well as important differences between the 2016 and 2017 classes. Twenty-five students took each entrance survey, out of 27 students in each class. Of the 2016 participants, nine (36%) were male and sixteen (64%) were female; in 2017 this ratio was reversed, with seventeen male students (68%) and eight female students (32%) responding to the survey. Of those students who chose to specify a nationality, there were eighteen from the U.S., two from China, one from the Netherlands, and one from Mexico in 2016; in 2017 there were sixteen from the U.S., four from China, two who identified as mixed European, and one "Mexican-American." The vast majority of both classes racially self-identified as White, with the 2016 respondents including two Chinese, two Hispanic, and two no response, and 2017 respondents including four Chinese, one Asian-American, one African-American, one Latinx, and two no response.

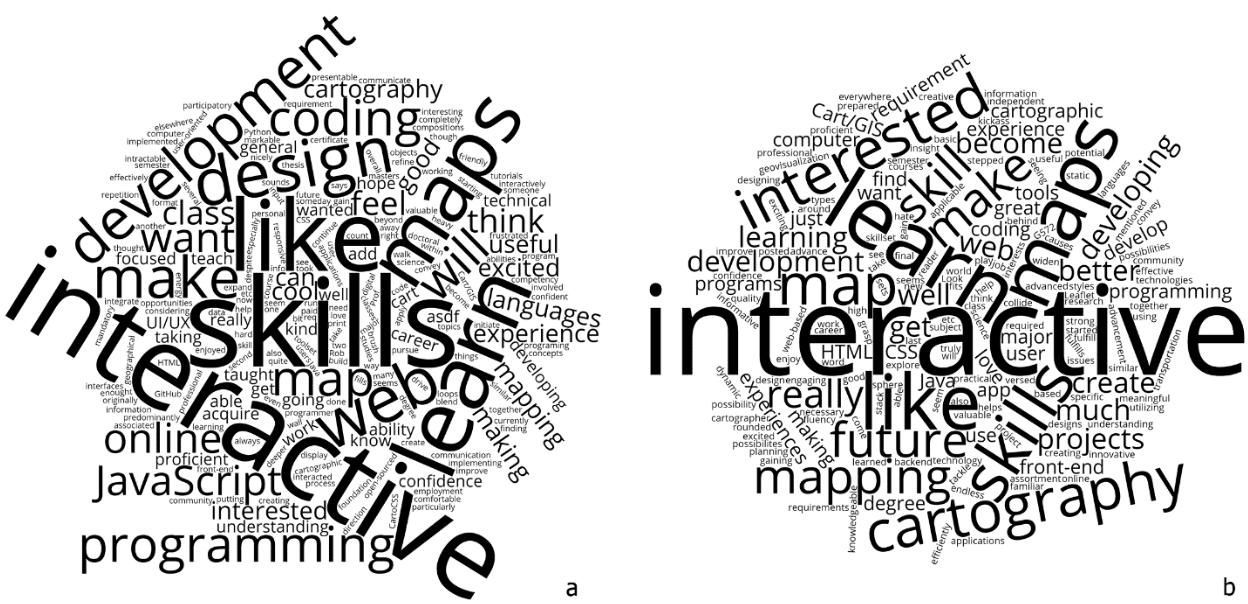


Figure 5.4: Word clouds based on students' stated reasons for taking the Interactive Cartography and Geovisualization course and learning goals in 2016 (a) and 2017 (b).

When asked about their learning goals and reasons for taking the course, thirteen students out of twenty-one who provided a response in 2016 (62%) and ten students out of eighteen who provided a response in 2017 (56%) discussed an interest in creating interactivity on the web and/or interactive maps. The emphasis on gaining technical skills to build interactive maps is demonstrated by the word clouds in Figure 5.4. Several students also mentioned a desire to improve their design skills (4/21 in 2016, 3/18 in 2017), programming skills (5/21 and 2/18), and general web development skills (4/21 and 3/18).

All but three students (88%) in 2016 and every student in 2017 had taken at least one prior programming course; this was to be expected, as an introductory programming course was listed as a prerequisite for Interactive Cartography and Geovisualization. In both classes, most students had *not* yet taken a course involving web coding languages; however, the 2016 course had nine students (36%) who had taken one web programming course and two students (8%) who had already taken three or more such courses, while the 2017 course had ten students (40%) who had taken one prior web programming course and no students with more than one such course behind them.

5.4.2 Exit Survey: Competence

While the entrance survey gauged students' prior experience and reasons for taking the course, the exit survey was a key instrument for determining the impact of the curriculum on the four learning outcomes identified in Section 4.5. As in Chapter 4, the before-and-after Likert scale ratings of student expertise with various web technologies gave an impression of the amount of growth in student *competence* with those technologies.

Because the entrance survey was used to collect data on prior expertise with various web technologies, prior expertise ratings could be compared between the two instruments to gauge the reliability of the analysis. Fourteen students completed the ratings on both surveys in 2016 and sixteen completed both in 2017. Both classes' prior experience ratings were consistent across the two instruments, showing only minor and statistically insignificant variations. The prior expertise ratings from the exit survey were used for the analysis presented in Tables 5.3-5.5, while the prior expertise ratings from the entrance survey were not analyzed further. This allowed for a slightly larger final sample size of sixteen students in 2016 and seventeen students in 2017 who completed the competence portion of the exit survey.

Table 5.3: Comparison of prior expertise with Open Web Platform tools used in the course as reported on 2016 and 2017 exit surveys. Expertise was rated on a 5-point Likert scale from 1 (I have never used this tool) to 5 (I am an expert with this tool). Cell color values for means are interpolated on a square root scale from 1 to 5.

Open Web Platform Tool	2016 mean initial expertise	2016 standard deviation	2017 mean initial expertise	2017 standard deviation
HTML	2.31	1.14	1.88	0.99
CSS	2.06	1.12	1.82	1.01
JavaScript	1.38	0.89	1.18	0.53
jQuery	1.31	0.87	1.12	0.49
AJAX	1.31	0.87	1.00	0.00
The DOM	1.56	0.89	1.24	0.56
GitHub	1.50	0.82	1.59	1.06
Leaflet	1.25	0.68	1.00	0.00
D3	1.19	0.75	1.00	0.00
JSON	1.44	0.89	1.18	0.53
GeoJSON	1.44	0.73	1.29	0.59
TopoJSON	1.19	0.4	1.29	0.69
SVG	1.56	0.81	1.65	0.79
CSV	3.31	1.2	2.69	1.04

Table 5.4: Comparison of final expertise with Open Web Platform tools used in the course as reported on 2016 and 2017 exit surveys.

Open Web Platform Tool	2016 mean final expertise	2016 standard deviation	2017 mean final expertise	2017 standard deviation
HTML	3.19	0.91	3.12	0.93
CSS	3.13	0.96	3.18	0.81
JavaScript	3.13	0.62	2.94	0.56
jQuery	2.81	0.83	2.65	0.7
AJAX	2.75	0.86	2.24	0.56
The DOM	2.94	0.68	2.59	0.71
GitHub	3.25	0.77	3.24	1.09
Leaflet	2.94	0.68	3.18	0.81
D3	2.94	0.77	2.94	0.83
JSON	3.00	0.73	2.76	0.97
GeoJSON	3.13	0.62	3.24	0.83
TopoJSON	2.94	0.68	3.24	0.83
SVG	2.81	0.66	2.59	0.71
CSV	4.00	0.82	4.06	0.9

Table 5.3 shows sample mean prior expertise ratings (on a 5-point Likert scale) and standard deviations from each exit survey for tools taught in the course. Table 5.4 shows the sample mean final expertise ratings and standard deviations from each exit survey. As in Section 4.4, the color codes applied to visualize each table's mean values are perceptually scaled by interpolating between the ColorBrewer Blues color scale extremes using a 0.5-power scale to enhance difference perception (Harrower and Brewer, 2003; Stevens, 1957). Most tools (11/14) show a slightly higher mean initial expertise in the 2016 sample than in 2017. However, none of these differences achieved statistical significance at the 95% confidence level when tested with an unpaired t-test. For the final expertise ratings, there is little difference in means and no apparent pattern among the differences between 2016 and 2017 results (eight tools are rated slightly higher in 2016, five are slightly higher in 2017, and one is identical for both classes). This appears to indicate that while some students in the 2016 class started at a higher average competence level than the 2017 class, both classes ended the course with approximately the same competence outcomes.

This finding is further reinforced by calculating the differences between before and after mean expertise ratings in the exit survey (Table 5.5). Percent changes in expertise are generally comparable across 2014, 2016, and 2017 classes, with no overall statistical significance between years when tested with an unpaired t-test. The web programming building blocks of CSS, JavaScript, jQuery, and AJAX saw greater amounts of change in expertise with each subsequent class, while for most other covered tools, 2016 saw the least improvement in expertise. This was particularly the case with Leaflet and D3. These differences could be attributed to a variety of factors; possibilities include the newness of the modular curriculum in 2016, teaching differences between lab instructors for each semester, and the slightly (though not statistically significant) higher average starting point of the 2016 class demonstrated in

Table 5.2. Consistent across all three classes, there was little change in expertise with the CSV data format and with tools that were not taught as part of the course lab, the exceptions being minor but statistically significant increases for Mapbox Studio/TileMill in 2014 and for OpenLayers in 2017.

Table 5.5: Comparison of normalized percent change of mean expertise from each exit survey. Percentages shown in brackets are not statistically significant at the 95% confidence level. Cell color values are interpolated on a square root scale from 35% to 225%.

Open Web Platform Tool	2014 % change of mean	2016 % change of mean	2017 % change of mean
Tools covered in lab curriculum			
HTML	41.4	37.8	65.6
CSS	49.8	51.5	74.2
JavaScript	75.7	127.3	150.0
jQuery	87.4	114.3	136.8
AJAX	88.8	109.5	123.5
The DOM	89.7	88.0	109.5
GitHub	107.5	116.7	103.7
Leaflet	152.7	135.0	217.6
D3	223.1	147.4	194.1
JSON	106.5	108.7	135.0
GeoJSON	131.9	117.4	150.0
TopoJSON	172.7	147.4	150.0
SVG	91.7	80.0	57.1
CSV	[13.1]	[20.8]	[16.9]
Tools **not** covered in lab curriculum			
Google Maps API	[27.0]	[40.0]	[43.5]
OpenLayers	[38.5]	[31.3]	55.6
ArcGIS Online	[26.7]	[-12.1]	[9.5]
Mapbox Studio/TileMill	51.0	[31.0]	[39.3]
CARTO/CartoDB	[33.6]	[12.3]	[18.2]
KML	[15.2]	[36.0]	[33.3]
CartoCSS	[0.0]	[11.8]	[23.1]

5.4.3 Computational Thinking and Confluence

Responses to exit survey questions regarding computational thinking and confluence skills were analyzed to produce an overall response valence for each of the six skills referenced in Section 5.3 by subtracting agreement with each negative statement from agreement with the corresponding positive statement (Figure 5.5). Student responses to the individual questions are visualized in Figures 5.6 (2016) and 5.7 (2017). The responses generally reflect positive achievement in both computational thinking and confluence. However, the level of achievement in every skill was greater for students in the 2016 class than in 2017. Across all six skills, the difference between the two classes is significant at the 95% confidence level. Once again, there could be a number of factors influencing these outcomes, such as teaching style and initial skill level of each class. Even so, the contrast is striking given the general evenness in competence outcomes between the two classes (Table 5.4).

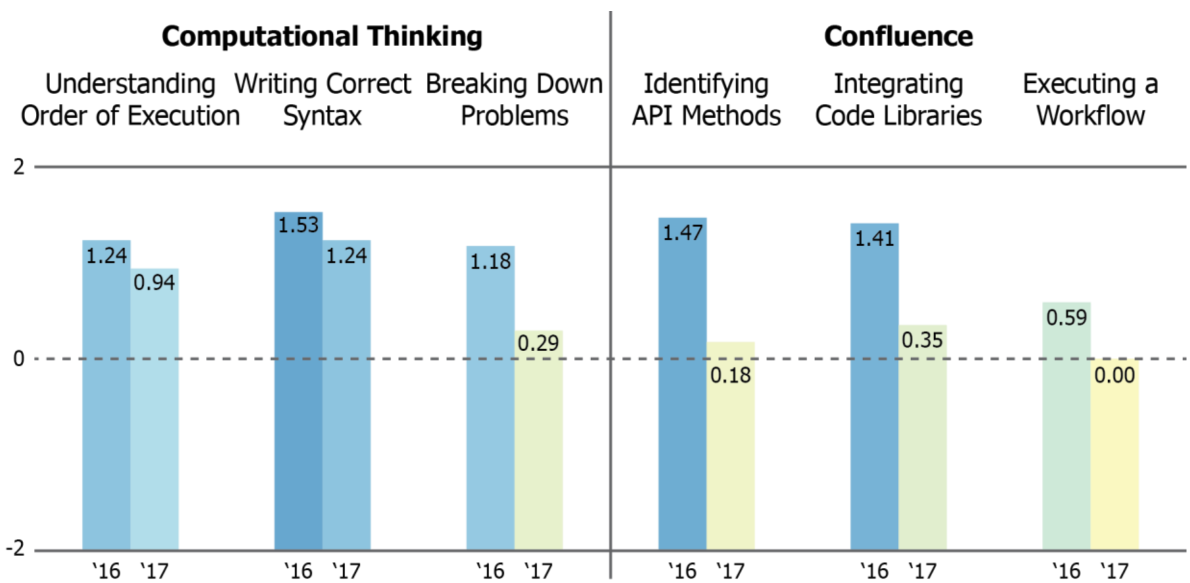


Figure 5.5: Comparison of computational thinking and confluence valences between the 2016 and 2017 classes. Bar colors are correlated to valence values, with bluer values more positive and yellower bars more neutral.

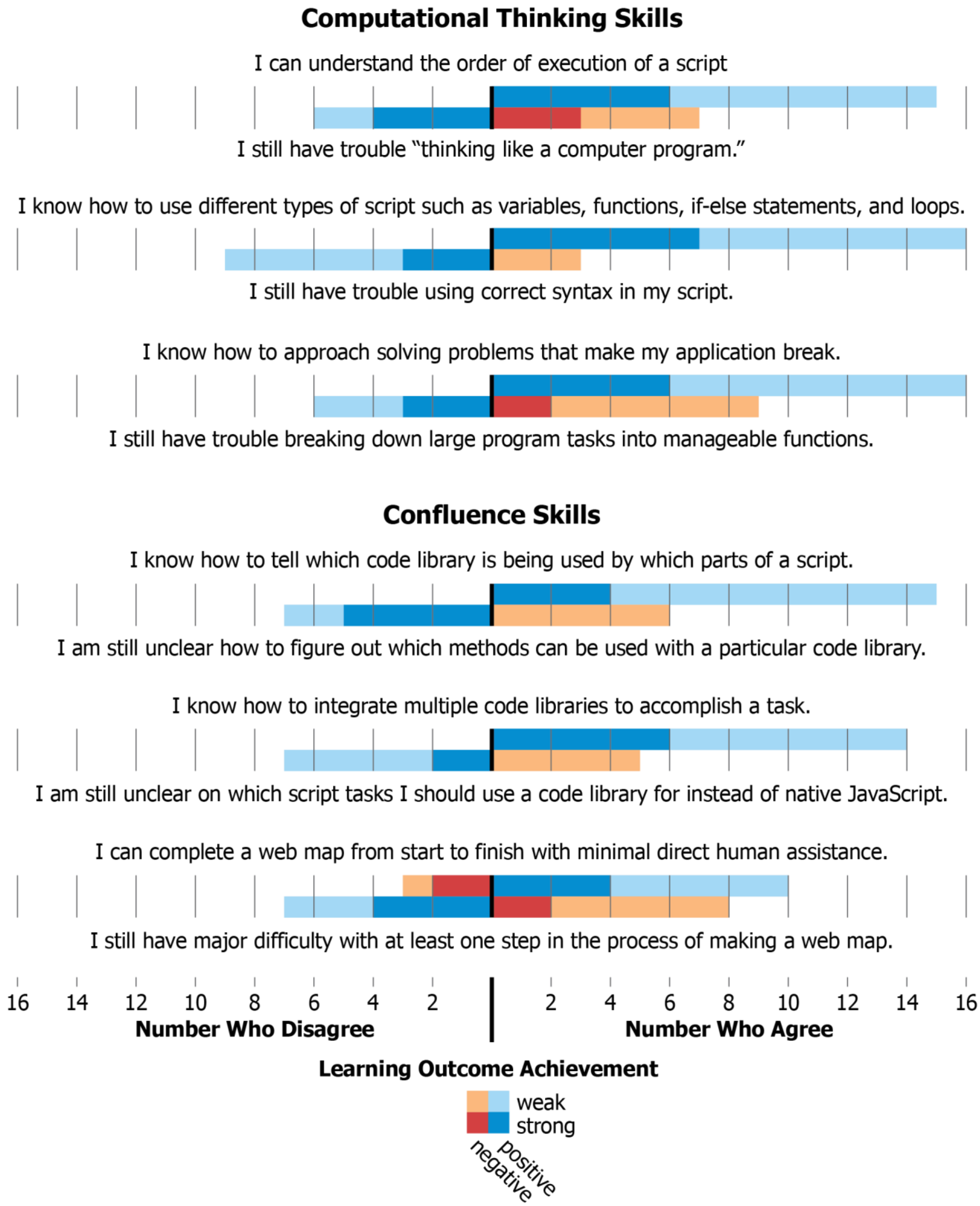


Figure 5.6: 2016 Students' agreement with outcome statements reflecting different computational thinking and confluence tasks. Paired questions reflect the same task phrased positively (top) and negatively (bottom).

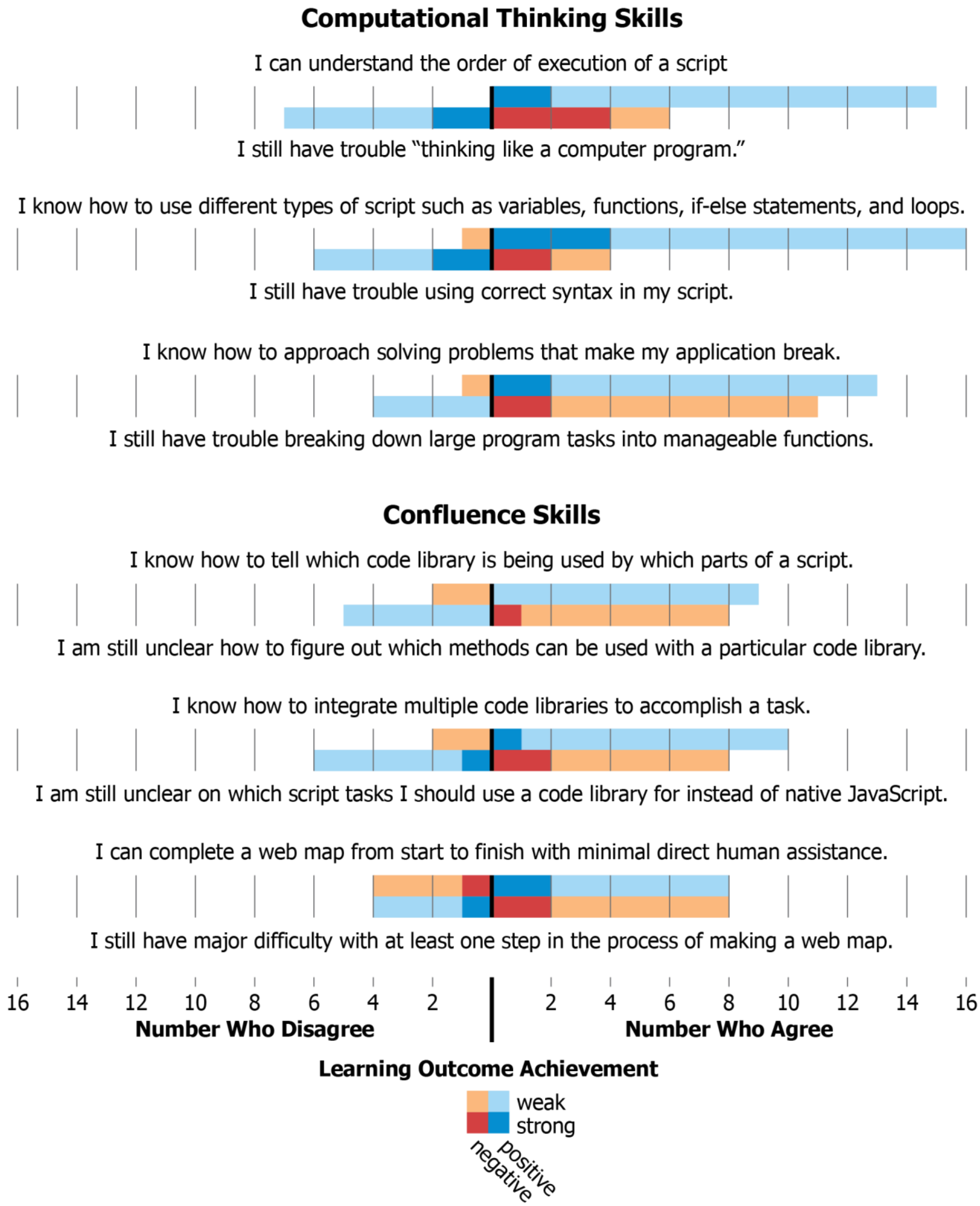


Figure 5.7: 2017 Students' agreement with outcome statements reflecting different computational thinking and confluence tasks. Paired questions reflect the same task phrased positively (top) and negatively (bottom).

Another striking comparison was the difference in patterns of student confidence throughout the course between the three offerings. On the exit surveys for all three classes, students were asked to rate their overall emotional experience with each lab unit, providing one measure of confidence. In 2016 and 2017, students also were asked to rate how difficult they would find it to accomplish each overall unit objective on their own after completing the course. Figure 5.8 compares the overall emotional experience and difficulty ratings for the 2016 and 2017 classes. It uses the five-class ColorBrewer Red-Yellow-Blue color scale to code student Likert scale responses from most negative and hardest (red) to most positive and easiest (blue; Harrower and Brewer, 2003).

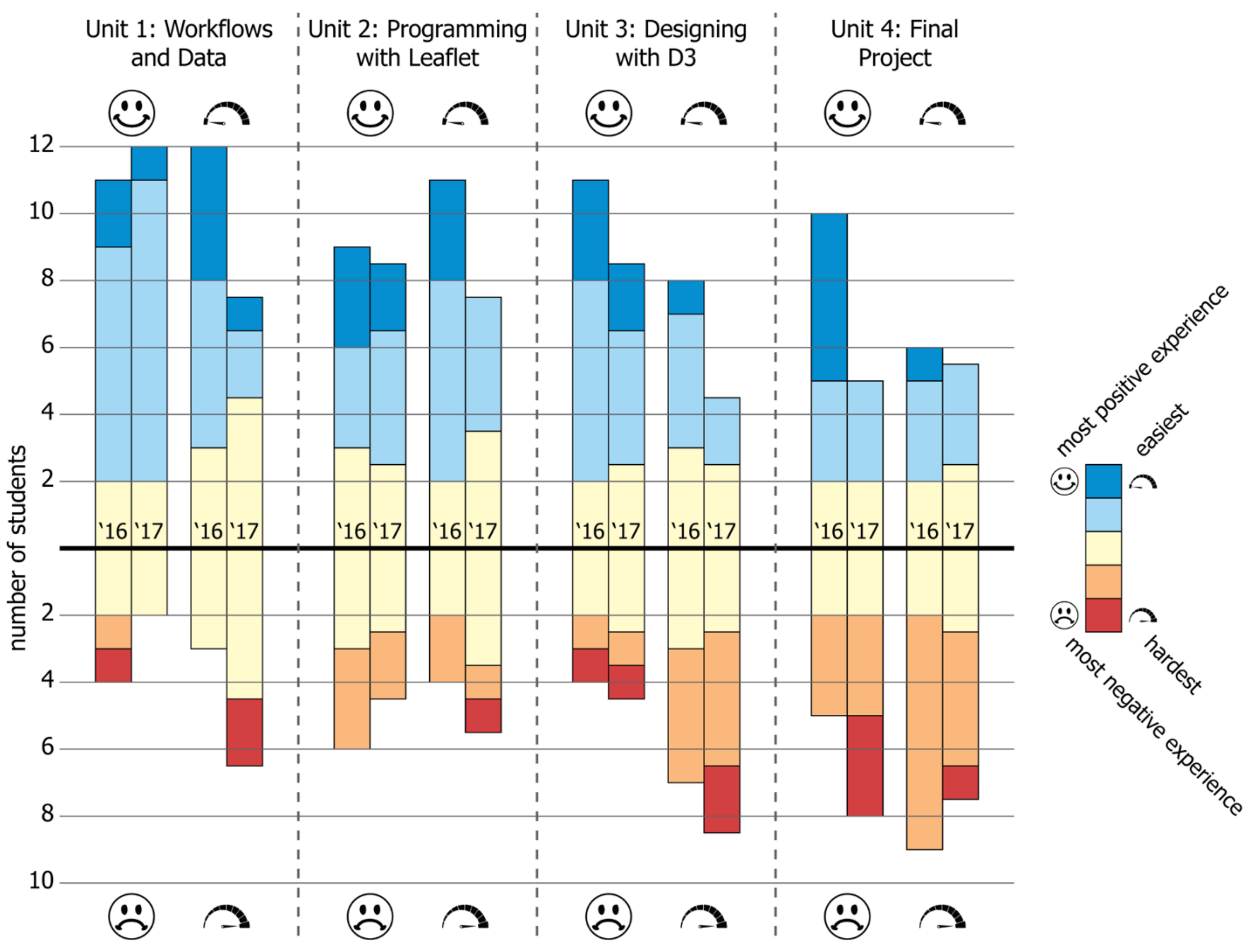


Figure 5.8: Comparisons of two measures of student confidence between 2016 and 2017: students' overall emotional experiences with each unit (left column pairs) and how much difficulty students believed they would have accomplishing each unit objective on their own (right column pairs).

Table 5.6 shows the mean valences between -2 (very negative experience/very challenging) and +2 (very positive experience/very easy) for each unit and class. Color codes for each value are perceptually interpolated along the Red-Yellow-Blue diverging color scale to visually reinforce the divergence between positive and negative mean valences (Harrower and Brewer, 2003; Stevens, 1957).

Table 5.6: Mean emotional experience and challenge valences between -2 (very negative/very challenging) and +2 (very positive/very easy) for each successive curriculum unit, color-coded using a red-yellow-blue scale for visual reinforcement.

	Emotional Experience		Challenge	
Unit	**2016**	**2017**	**2016**	**2017**
1: Workflows & Data	0.53	0.78	0.87	0.00
2: Leaflet	0.40	0.46	0.67	0.08
3: D3	0.60	0.38	0.13	-0.46
4: Final Project	0.67	-0.46	-0.13	-0.23

In 2014, the emotional experience appeared to conclusively grow more positive with each successive unit of the Leaflet and D3 labs and Final Project (an equivalent to the first unit of modular curriculum was not tested in 2014; see Figure 4.1). However, the picture for the 2016 and 2017 classes is more mixed. In 2016, students' experiences seemed to dip with the Leaflet unit (mean valence of 0.4 vs. 0.53), while the D3 and Final Project units had similar valences, with the Final Project slightly higher in mean valence but with slightly fewer students experiencing positive outcomes overall (Figure 5.8 and Table 5.6, left columns). The 2017 class experienced a clear overall decline of emotional experience with each successive unit, most dramatically with the final project, which saw a precipitous drop in morale to a mean valence of -0.46, and twice as many students left with a negative experience as with a positive one.

In terms of difficulty level, the 2016 class rated each unit's objective successively more difficult (Figure 5.8 and Table 5.6, right columns). The 2017 class rated building a map with D3 (Unit 3) the most difficult (mean valence of -0.46) and working with colleagues to complete the entire web mapping workflow (Unit 4) slightly less difficult (-0.23), even though their overall emotional experiences were much more negative with the latter than with the former. They rated building a simple AJAX web application (Unit 1) and building a Leaflet map (Unit 2) the easiest and about on par with one another (mean valences of 0 and 0.08, respectively). Overall, the 2017 class appeared to have lower confidence than the 2016 class, although the difference between mean valences does not quite rise to statistical significance when emotional experience and challenge valences are combined and compared across classes (p=0.056).

5.4.4 Exit Survey: Sequence, Challenges, and Student Feedback

While students were asked to reevaluate the topic order in 2016 and 2017, neither class chose to significantly reorder the lesson topics, indicating no strong preference for any sequencing changes.

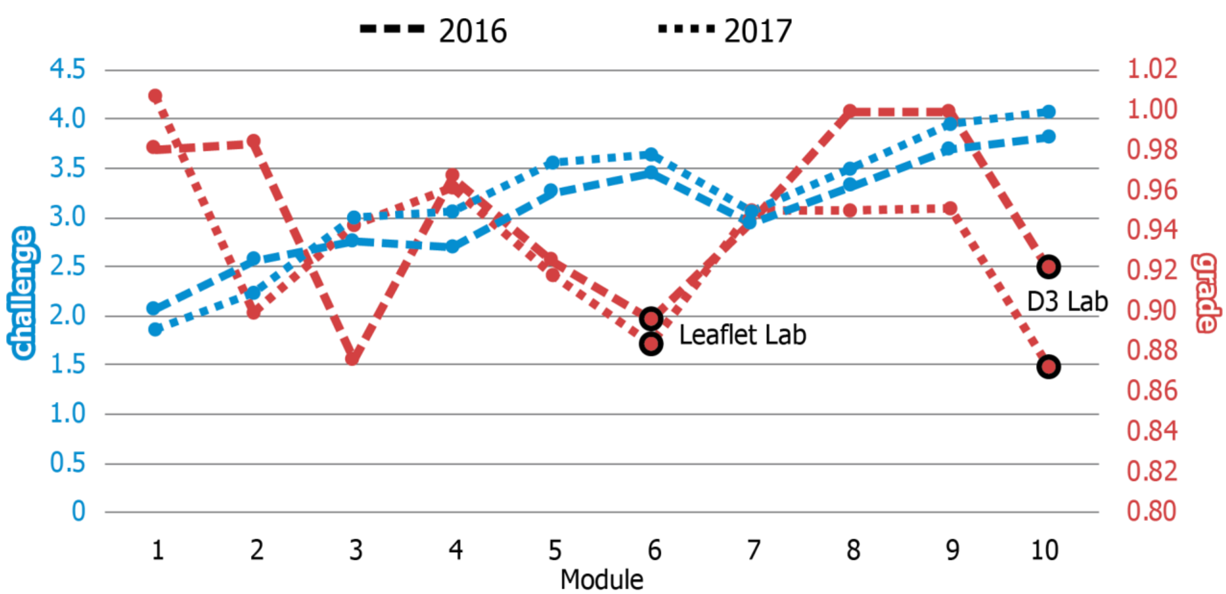


Figure 5.9: Graph showing each module's mean challenge rating (blue) and mean student grade on the module assignment (red) for the 2016 and 2017 classes. The outlined grade dots represent the major lab assignments; smaller dots represent smaller module activities.

Figures 5.9 shows students' mean ratings of each module's level of challenge along with the mean grade earned by students who completed each module assignment as a percentage.

For both classes, the perceived challenge level increased fairly gradually and consistently throughout the course, with the minor exception of Module 7, the beginning module of the D3 unit, which was logically seen as easier than the final module of the Leaflet unit (it functioned as the start of a new turn around the spiral and thus dropped lower on the scaffold). Students' activity grades appeared to have little correlation with the challenge level of each module. However, congruent with the other measures of class progress discussed previously, the 2017 class rated most of the modules as more challenging and achieved lower mean grades for several module assignments than did the 2016 class.

In response to open-ended exit survey questions, individuals expressed a variety of opinions—some contradictory—regarding what could be added, removed, or changed from the lesson material, but few cohesive themes emerged. One exception was a request from several students (four in 2016 and two in 2017) for more material on working with CSS, a request that also was noted in the 2014 exit survey but not integrated into the modular curriculum due to its coverage in a different course. Regarding the course overall, the clearest feedback theme, expressed by two students in 2016 and three in 2017, was that the lectures and labs felt disconnected or could even be taught as two separate courses. A perhaps blunter sub-theme, expressed by two students in each class, was that the coding portion felt more valuable, and any lectures should focus on the coding aspects of the course. This feedback seems to reflect the strong orientation towards employable skills within many students' learning goals expressed in the entrance survey and shown in Figure 5.4.

5.4.5 Weekly Feedback Submissions

Weekly feedback submissions from students provided a rich qualitative dataset chronicling individual students' progress. These student experiences were very diverse. During the first unit (Modules 1-3), almost half of 2016 students (13/27) reported that they struggled with GitHub, but ten said that they found the assigned online JavaScript tutorials (from either Codecademy or Lynda.com) helpful. Many (11/27) also found the debugging practice in the second and third modules useful, and several reported exciting breakthroughs (5/27) or feeling like they were learning the material in general (4/27). In 2017, three students (out of 25) reported difficulty setting up a local server, and six found debugging a challenge. As in 2016, though, several (8/25) found the online tutorials useful, and several (6/25) reported improvements in their understanding of coding concepts during the first three weeks of the course.

During the second unit, accompanied by the Leaflet lab assignment, several 2016 students reported feeling satisfied with their ultimate product (6/27) or finding the lab exercise useful (11). Multiple 2016 students reported difficulties with formatting the map symbol legend (6/27) and working with CSS in general (2/27). Thirteen students in 2017 and five in 2017 struggled to add custom interactions to the map for the open-ended portion of the lab assignment requiring them to seek out a solution beyond the lab module content. This is consistent with students' experiences with the Leaflet Lab in 2014, discussed in Section 4.5.

In the third unit, covering the D3 lab assignment, multiple 2016 students found D3 easy to learn (3/27), understandable (3/27), and enjoyable to use (4/27). Conversely, two 2016 students reported difficulty understanding D3 concepts and syntax. The greatest problems students encountered involved loading their custom datasets into the script, with ten students reporting errors when trying to access their data. This is not surprising, given that the lab assignment involved storing and using data in TopoJSON format, a more complex format with a more involved loading process than the GeoJSON data used for Lab 1. In 2017, seven students reported positive experiences with D3, although two reported difficulties related to that library's transition to a new version with significant API changes (Version 4). Three students reported problems finding appropriate data, and five had issues loading the data once it was ready. Three also reported struggles implementing highlighting and dehighlighting functions, which required students to write some code from scratch rather than copy and paste example code.

To better identify the threshold concepts contained in students' feedback, statements from the weekly feedback submissions were distilled into more general themes related to successes and difficulties that students experienced. Statements were categorized as indicating either breakthroughs and other positive experiences or struggles and other negative experiences, then grouped and tallied by common themes. An overall valence was derived from subtracting the number of negative statements from the number of positive statements for each theme. The results of this analysis are presented in Table 5.7. As in Table 5.6, valence color codes are perceptually interpolated using the 5-class ColorBrewer Red-Yellow-Blue color scale to reinforce the divergence between positive and negative valences (Harrower and Brewer, 2003; Stevens, 1957).

The valence tally gives a good general indication of where students struggled the most and where they achieved the most success and confidence. The greatest frustrations for students were the limited time allotted given the expectations for deliverables (valence of -29), independent writing of custom code (-28), producing and formatting data (-24), and styling web pages and map elements (-23). Finding appropriate data (-17), finding solutions to code problems in online forums (-17), integrating module example code into the student's own code (-16), and using GitHub (-15) were also often seen as challenging.

Table 5.7: Themes related to successes (+) and challenges (-) reported by students in weekly feedback submissions.

Theme	2016 +	2016 -	2017 +	2017 -	Total +	Total -	Overall valence
General understanding and confidence	26	4	20	14	46	18	28
D3 code	19	3	12	4	31	7	24
Online tutorials	17	10	18	2	35	12	23
Debugging	21	12	16	17	37	29	8
Satisfaction with product	7	1	4	2	11	3	8
Getting a working product	4		6	2	10	2	8
Prior coding experience	3		4		7	0	7
Assistance from instructor/peers	2	2	5		7	2	5
Using documentation	3	2	4		7	2	5
Basic web concepts	4				4	0	4
Pseudocoding	2		1		3	0	3
Lecture concepts	1		1		2	0	2
Leaflet code	5	5	4	3	9	8	1
Collaboration	2		2	3	4	3	1
Project management		1			0	1	-1
D3 projections	3	2		3	3	5	-2
Choosing from multiple code options				2	0	2	-2
Commenting code		1		1	0	2	-2
Identifying library methods		1		1	0	2	-2
Formatting SVG elements		3		1	0	4	-4
Sequence slider	2	4		3	2	7	-5
Refactoring code	1	3		3	1	6	-5
Dynamically adjusting a visualization		5			0	5	-5
jQuery	1	6	1	2	2	8	-6
Lack of challenge		5	1	2	1	7	-6
Linking to external files	1	6		1	1	7	-6
Implementing a Leaflet plugin		4		2	0	6	-6
HTML		6			0	6	-6
JavaScript variables, objects, methods	3	7		3	3	10	-7
Dynamic SVG legend	3	9		1	3	10	-7
Linking interactions between elements		3		4	0	7	-7
Loading tiles from a service		5		2	0	7	-7
AJAX	10	20	4	2	14	22	-8
TopoJSON		6	1	4	1	10	-9
Development environment		4	2	8	2	12	-10
Data processing in script		6		4	0	10	-10
Control flow/order of execution	1	13	4	3	5	16	-11
Creating a D3 visualization	3	12		4	3	16	-13
GitHub	7	20	4	6	11	26	-15
Implementing example code	4	8	4	16	8	24	-16
Finding code solutions online	4	20	1	2	5	22	-17
Finding appropriate data	1	11		7	1	18	-17
Styling and CSS	1	10		14	1	24	-23
Data production and formatting	3	11	1	17	4	28	-24
Custom interaction code	5	23	7	17	12	40	-28
Time constraints		12	1	18	1	30	-29

By contrast, students experienced the most success with recognizing overall growth in their coding skills (+28), understanding D3 code (+24), and learning from online tutorials (+23). Students also had overall positive experiences with debugging (+8), the level of satisfaction with their creations (+8), seeing a working product (+8), and leaning on skills learned in prior courses (+7).

5.5 Lab Outcomes and Barriers to Learning

The results of the 2016 and 2017 curriculum evaluations provided a greater understanding of course outcomes and a point of contrast between the fully in-person and blended versions of the Interactive Cartography and Geovisualization course lab. Before discussing the results in detail, it is necessary to reiterate that they were affected by several confounding variables that could not be fully controlled in a classroom setting. These included different lab instructors for each iteration of the course with accompanying differences in expertise and teaching style, differences in levels of prior expertise, external time commitments, and interest levels among student enrollments, different group dynamics among students in the two classes, slightly different semester schedules between the two offerings, and different self-selected samples of students responding to the surveys and weekly feedback prompts. However, the sample sizes for each measurement instrument were appropriate given the nature of the analysis, and the consistency of the teaching materials, scope, sequence, and evaluation techniques used for each iteration lend validity to the qualitative analysis results. The paragraphs below will discuss outcomes for each of the four learning goals (competence, computational thinking, confluence, and confidence), then address challenge points and barriers to learning, and finally make some recommendations for how the course lab might be improved in the future.

In terms of competence, the modularization of the curriculum appears to have had little or slightly positive impact on student growth. As shown by Table 5.5, like in 2014, almost all of the tools covered in lab materials saw significant growth in student expertise, while almost none of the dummy tools (those not covered in lab) saw significant change. The 2017 class appears to have experienced the greatest overall competence growth, despite very slightly lower mean initial expertise ratings in 2017 than in 2016 (see Table 5.3). In other words, it appears that the 2017 class effectively played "catch-up" with the 2016 class in terms of knowing how to use individual tools that were taught in lab.

However, this "catch-up" did not extend to computational thinking or confluence. Examining Figures 5.5 and 5.6, computational thinking—the ability to understand the order of execution of a script, use correct syntax, and break down problems into solvable tasks—saw slightly better outcomes in 2016 than 2017. The difference between semesters was even more noticeable regarding confluence, or the ability to distinguish which methods belong to a particular code library, integrate multiple code libraries to accomplish a task, and integrate tasks to complete the web mapping workflow. In 2016, student responses to confluence questions were comparable in valence to computational thinking responses; whereas in 2017, confluence valences ranged from only slightly positive to totally neutral overall. The lack of negative valences points to at least some confluence growth in both semesters. Nevertheless, the greater divergence between semesters in confluence compared to competence points to a learning gap between the ability to use individual open web tools (competence) and the ability to understand how the tools work together to accomplish a program task (confluence). While the former ability clearly can be taught and appears less dependent on prior knowledge, the latter ability seems to largely come with experience. Computational thinking may be somewhere in the middle, drawing from both instruction and experience.

The most troubling outcome observed in the evaluation was the overall decline in emotional experience with each successive course unit in 2017, a reversal from the 2014 evaluation (Table 5.6). The reasons for this loss of confidence over time are not clear, and may be partially due to factors outside of the course lab curriculum described at the beginning of the section. Open-ended student feedback indicated that data problems, difficulty implementing custom code, and feeling overwhelmed by module assignment due dates may have taken a toll during the latter half of the course. 2016 students remained relatively steady emotionally, but like the 2017 class felt a steady increase in the level of challenge. One could speculate that the modular format overly rigidizes expectations for each unit, making it harder for the lab instructor to adjust downward or go back to reinforce prior concepts to meet students' scaffolding needs. Further research that better controls for external influences on students' experiences with the course material would be needed to test this hypothesis.

Success points described by students, shown in Table 5.7, mostly involved students observing their own growth and achievements. Feelings of accomplishment at squashing a bug, getting a feature to work, or simply understanding a new concept through an 'aha!' moment boost student confidence. Students saw the online tutorials required at the beginning of the course and the Leaflet tutorials required as part of Unit 2 as helpful resources. The approach to D3 in Modules 7 and 8 continues to generate student excitement, with students experiencing overwhelmingly positive associations with D3 even as other coding challenges present themselves.

Students' greatest points of frustration were lack of time, implementing custom code, CSS styling, and working with data. The time crunch is a consistent and difficult problem to address, given the learning objectives of the course. Learning highly technical web development skills in a semester is simply a tall order, regardless of sequencing. However, the modularization of the curriculum may have exacerbated the problem by adding additional content to address threshold concepts identified in Chapter 4 and make up for the lack of ability of an in-person lab instructor to extemporize as needed in the online distance learning environment. The number of topics covered by the curriculum jumped from 20 in 2014 to 34 in the modularized curriculum, a 70% increase. While some of these topics were created by splitting former topics, additional explanatory content was added in several places, as discussed in Section 5.2. This new content did not necessarily introduce new skills, but simply made working through modules to learn the existing skills a time-consuming process.

Customization of code is a skill of the highest order in Bloom's cognitive taxonomy—Create—and requires synthesis of computational thinking, competence, confluence, and confidence (Anderson and Krathwohl, 2001). Repeated practice, strong scaffolding, and adequate instructor support are prerequisites, but additional time to work out problems independently can also help students here. Complaints about the difficulty of styling a map and web page are more specific to particular tools, and indicate the need for additional training with CSS, perhaps through requiring an additional online tutorial on it. Again, more time would be needed to implement such a requirement. Data is always the biggest challenge of any mapping project, and likewise points back to the need to allot more time. Time pressure is the key thread running through these various difficulties.

The greatest challenge for the modular lab curriculum seems to be maintaining student confidence throughout the semester while attempting to accomplish a highly challenging and technical set of learning outcomes. If student confidence can be maintained, the level of challenge can remain high. However, as witnessed with several students in 2013 and with one student in 2014, burnout is a very real danger when confidence dips. To feel confident in their learning abilities, students need to experience success, and success with difficult goals takes time. There are a few options for change that might address the time crunch problem while maintaining the course goals and modular curriculum structure.

In open-ended feedback on the exit survey and weekly submissions, some students suggested separating the lecture and lab as two different courses as one way to relieve time pressure. This strategy is not recommended, as the UI/UX design principles covered in lecture are required for producing web maps that are usable and effective as well as technically sound. Rather, student feedback in this vein may indicate a need for further integration of the lecture and lab components of the course. In particular, assessments could be modified such that the exams and quizzes given in lecture explicitly tie coding skills such as debugging and library integration to UI/UX concepts, while module deliverables and lab assignments could require students to critique their work based on the design tenets introduced in lecture.

Another possibility for extending the course length is to offer the course as five to six credits instead of four. If it continues to be taught over a single semester, this would necessitate students taking fewer other courses, leaving them more time to concentrate on Interactive Cartography and Geovisualization coursework. If split into two 3-credit semesters, each semester could cover two of the lab units and associated lecture content, with more weeks dedicated to work on each unit and accomplish all required assignment tasks at a higher level of quality. However, serious drawbacks would include the requirement for more instructional labor that may be hard to come by and the added load on student schedules that could throw off degree plans, lengthen time to degree completion, and make the course a less attractive option than competing computer science and GIS courses.

Adjusting the expectations for assignment deliverables may prove a more achievable option than lengthening the course. Students are required to find their own datasets for each lab assignment, which is valuable in that it adds to the real-world applicability of the coursework and calibrates assignments to students' personal interests. However, the parameters of each dataset could be made more flexible to speed data collection and processing, or students could be required to create a single dataset for use in both major lab assignments. In addition to saving data processing time, this would deepen students' understanding of the structure of their data and how it can be manipulated using JavaScript code.

Additionally, implementing custom code for a fifth interaction operator in the Leaflet lab was seen by students as among the most difficult tasks of the semester. Such a synthetic task may come too early in the course sequence for students to be adequately prepared for it. It may be better to require only four interaction operators and allow students to customize their interface elements. Likewise, students may feel more successful with the D3 lab if fewer features are required in the final deliverable. With each assignment, more emphasis should be placed on achievable goals leading to student success and less on the richness of interactivity included in the final product. For less experienced students, accomplishment of fewer goals may boost confidence, while for more advanced students, fewer requirements could open up opportunities for greater creativity and self-direction in the design of the final deliverable. Encouraging students to prioritize achievable goals over advanced functionality for the final project could reduce stress and result in better-designed, more web-ready products. In short, this approach reorients the lab assignments from maximizing the demonstration of UI/UX principles to accomplishing fewer goals and doing them creatively and well.

Whether or not any of the options outlined above are implemented, this evaluation study makes clear that students in the Interactive Cartography and Visualization course learn valuable web development and web mapping skills from the course lab. Overall, the modular lab consistently succeeds in the learning goals of computational thinking and competence even with different levels of prior experience and class dynamics among students. Further research is needed to determine the primary factors affecting confluence and confidence. Confidence was the greatest point of departure from the pre-modular lab curriculum, but this change did not result in observable burnout among students in either iteration of the course. Modularization of the scaffolded and spiraled lab curriculum can therefore be considered successful at achieving similar outcomes to the pre-modular curriculum, albeit with continued room for improvement.

VI. Findings and Applications

Abstract

This chapter reviews and synthesizes the research presented in this dissertation and explores future research possibilities. Section 6.1 reviews the findings of the studies presented in Chapters 3, 4, and 5 in answer to the three research questions presented in Section 1.5. It also discusses the limitations of each study. Section 6.2 synthesizes the findings of these three studies to propose a curriculum for an introductory web mapping course at the community college level. This curriculum is geared toward lower-level undergraduates, envisioning how the complex technical skills of web mapping might be introduced to students with little or no prior computer science knowledge. Section 6.3 reviews the contributions of the work presented in this dissertation to the fields of Cartography, GIScience Education, and Online Instructional Design. Finally, Section 6.4 presents future research directions that could build on the findings of this dissertation regarding effective web mapping instruction.

6.1 Research Findings

This dissertation sought to answer three research questions. The findings for each question are summarized below.

RQ1. What are the major barriers to teaching open web mapping, and what instructional practices can overcome those barriers?

The first question was answered by an interview study of educators who teach web mapping in North American universities (Chapter 3). Twenty instructors of web mapping courses were interviewed about their teaching practices for the study. Following the tenets of qualitative data analysis, transcripts were coded using 26 codes across categories consisting of Course Context, Technology, Resources, Setting, Curriculum, and Teaching. The most salient codes were found to be VISION, SCOPE, TOPIC, TOOL, MOTIVATION, PEDAGOGY, and CHALLENGE. Instructor statements receiving these codes were grouped by common theme, and frequency and extensiveness tallies were reported for each theme under each code to derive key takeaways regarding the challenges and successful practices of web mapping instruction.

The results of the interview study analysis indicated that common barriers to teaching web mapping included imparting basic coding skills on inexperienced students, keeping curriculum and instructor skillsets up to date given rapid changes in web mapping technology, technical failures of cloud-hosted services, and receiving adequate support and resources from the employing institutions. Web mapping was taught in one of four ways: as a standalone topic focused on technical web map development skills, using web mapping skills as a gateway to critical geographic theory, using web mapping platforms to promote spatial thinking and GIS, and focusing on cartographic design for web maps. Most instructors covered the basics of HTML, CSS, and JavaScript, but a subset of instructors eschewed teaching code altogether. Instructors generally chose teaching tools that were free, easy to use and teach, and presumed to be relevant to students' future jobs. Some instructors stayed entirely within the Esri technology suite because of its workplace relevance, integration, and license availability, while others sought to expose students to a wide variety of different web mapping technologies they might encounter in the future. Virtually all instructors relied on the constructivist principle of active learning and assigned at least one student-directed project assignment.

RQ2. What skill-based learning outcomes for open web mapping are achievable in a one-semester upper-level undergraduate Geography course?

The second research question was addressed by an evaluation of lab curriculum for Interactive Cartography and Geovisualization using Open Web Platform technologies (Chapter 4). The lab curriculum introduced interactive web mapping over the course of twenty lesson topics in three units, with a fourth unit for a collaborative, student-directed final project. The topic sequence was designed as a spiral, with web mapping skills introduced in each new unit building on skills used in the prior unit(s). Additionally, the curriculum was scaffolded to rely heavily on instructor support early in the semester, and gradually introduce more independent work as students' coding skills improved.

The lab curriculum was evaluated using three instruments: an instructor log, student feedback compositions submitted with each major lab assignment, and an extensive exit survey. The instructor log collected anecdotal observations of student progress. Student feedback compositions were coded to determine what difficult concepts, other problems, 'aha!' moments, and other positive outcomes students experienced with each lab assignment, with common themes in each category tallied by frequency. The exit survey used both Likert scale and open-ended response questions to assess students' growth in expertise with web mapping technologies used in the course, the level of challenge of each technology, the efficacy of the topic scope and sequence, the usefulness of different learning resources, and the overall emotional experience students associated with each major lab assignment and the final project.

The curriculum evaluation revealed that the most important general learning outcomes of an undergraduate-level web mapping course are computational thinking, competence, confluence, and confidence. *Computational thinking* involves the ability to step through the order of execution in code, understand code syntax and control mechanisms, and break down large program tasks into manageable chunks. Qualitative evidence of student progress in computational thinking was provided by the instructor observation log and student feedback compositions.

Competence involves understanding how to make use of specific tools within the web mapping ecosystem. Evidence of growth in competence was primarily provided by student assessments of their expertise with web mapping tools before and after the course, measured in the exit survey. Mean expertise ratings demonstrated statistically significant growth from low to moderate levels for each tool introduced in the lab curriculum except one that students had prior familiarity with. These results demonstrated that the lab curriculum was effective in growing student competence to passable levels across the technologies it introduced.

Confluence is the ability to integrate Open Web tools to create a working application, and requires understanding which objects and methods are invoked from each component and complete a web mapping workflow from start to finish. Students provided qualitative evidence of growth in confluence in their feedback compositions through discussion of learning how to incorporate example code, recognize methods belonging to different code libraries, and integrating those libraries with one another and with JavaScript, CSS, and HTML.

Finally, *confidence* is a cognitive and emotional state of awareness that one's skills are growing and faith in one's ability to learn what is necessary to complete a web mapping task. Evidence of growth in confidence was provided by students' ratings of their overall emotional experiences with each major lab assignment and the final project, collected by the exit survey. Students reported feeling increasingly positive with each successive assignment. Additional anecdotal evidence of student confidence was provided by statements in the instructor log and student feedback compositions that indicated many students felt a sense of excitement as their skills and accomplishments grew through the course of the semester.

While a single-semester web mapping course cannot turn web programming novices into web mapping experts, it can give students enough computational thinking, competence, confluence, and confidence to complete a web map and continue to develop their web mapping skillsets independently after the course is finished.

RQ3. How does student achievement of the identified learning outcomes for web mapping compare between in-person and modular, blended instruction?

The third study evaluated two subsequent semesters of Interactive Cartography and Geovisualization lab curriculum after that curriculum was modularized for delivery over the web in fully online and blended course formats (Chapter 5). Lessons were grouped into ten learning modules, each of which was written as a single web page with 2-4 lessons. Module pages were formatted as written step-by-step tutorials with conceptual explanation, figures, example code blocks, and links to online resources. Some topics from the curriculum described in Chapter 4 were expanded, others reordered, and some removed based on feedback from students in the first curriculum evaluation and program needs. The four-unit spiral structure of the curriculum was retained and reinforced, with scaffolding built into the modules as increasing opportunities for students to implement custom code solutions based on external resources in their lab assignments. As in Chapter 4, the products of the second, third, and fourth units were each of the two major lab assignments and the final project, respectively; however, weekly 'checkpoint' assignments that built toward the major deliverables were given with each module.

Identical evaluations of the modular lab curriculum were conducted with the 2016 and 2017 blended course offerings. Evaluation instruments consisted of an entrance survey, an extensive exit survey modeled on the survey described in Chapter 4, and student feedback compositions and grades on weekly module assignments. The entrance survey collected student demographics, prior experience with coding and web development, and learning goals for the course. The exit survey included Likert scale and open-ended questions aimed at assessing computational thinking, competence, confluence, and confidence outcomes, in addition to challenges experienced by students. Weekly student feedback compositions gathered student reflections on their progress throughout the course. Student grades were collected and averaged for each weekly assignment to help gauge the difficulty level of each module.

The 2016 and 2017 curriculum evaluations found that there was little difference between in-person and modular curriculum on students' competence with various web mapping tools, which increased significantly in both environments. Computational thinking and confluence both saw positive gains from the course, though differences in proficiency between the two classes indicated that these outcomes were partially dependent on class attributes that were not well controlled in the study. Unlike the Chapter 4 study, the modular curriculum appeared to produce steady or declining confidence over the course of the semester, indicating that the modular curriculum might attempt to do too much given the time allotted for the course. However, student burnout did not appear to be a major issue. Overall, the evaluation showed outcomes of the modular curriculum to be largely comparable to the curriculum presented in Chapter 4. It thus indicated that the modular curriculum was effective at supporting learner needs in a blended course setting, but there remained room for improvement, particularly efforts aimed at cutting back the time commitment required to complete the lab assignments.

Together, the findings related to the three research questions provide important information for GIScience instructors looking to create or revise a web mapping course or integrate web mapping into existing curricula. However, the study findings are limited in a number of ways. Due to time constraints and to examine similar institutional contexts, the interview study reported in Chapter 3 only recruited participants from Anglophone colleges and universities in the United States and Canada. Further work could be done to examine how web mapping is taught in other countries, characterize differences across institutional contexts, and possibly draw from novel concepts and practices that have not yet penetrated North American higher education.

The research reported in Chapters 4 and 5 only considered the lab component of the Interactive Cartography and Geovisualization curriculum, and did not attempt to assess student mastery of the design concepts presented in the course lecture. Further, qualitative measures based on student self-reporting and instructor observation were used to evaluate the curriculum efficacy. Assessment instruments with a narrower focus on individual assignments, collected by an independent observer rather than the instructor or students themselves, could be used to better quantify and delineate the dimensions of student learning involved in web mapping.

The study reported in Chapter 5 attempted to measure the efficacy of the modular lab curriculum against the learning outcomes presented in Chapter 4. However, the differences in performance between the two assessed classes indicated that there were uncontrolled confounding variables affecting learning outcomes. Additionally, the lack of questions regarding computational thinking and confluence skills on the 2014 exit survey made it difficult to directly compare these outcomes between the original and modular curricula. Controlling for variables such as amount of time spent on outside commitments and instructor teaching style could generate improved measurements of student progress. As with the Chapter 4 study, narrowing the focus to individual lessons and assignments and using more third-party assessment could improve understanding of student learning processes. Finally, additional evaluations could be conducted on the fully online version of the course, which was not evaluated for this study due to unanticipated difficulties with data collection that could be resolved for a future offering of the online course.

6.2 Disciplinary Contributions

Overall, the intended impact of this dissertation is to expand the franchise of web mapping education to enough additional GIScience programs that it becomes a standard component of GIScience curriculum, thus giving all future cartographers and GIS professionals the skills necessary to make an interactive map on the Open Web. The research findings of the dissertation provide knowledge contributions furthering this goal within the three fields of inquiry introduced in Chapter 1: Cartography, GIScience Education, and Online Education. This section will review the dissertation's contributions to each field in turn.

6.2.1 Contributions to Cartography

This work contributes to the field of Cartography by describing web maps and situating their design and development as essential to modern cartographic practice. Sections 2.1 and 2.2 are necessary background for understanding what exactly is being taught by instructors who teach web mapping. These components of the dissertation also stand alone as useful information for Cartography researchers, practitioners, and educators who work with web maps and web mapping. The information contained in these sections consolidates and synthesizes much of the literature published to date on web maps. Section 2.1 provides clear and succinct definitions of terms related to web mapping, includes a web map typology, and categorizes the technologies currently available for making web maps. Section 2.2 provides a succinct framework for understanding the components of a web map, describes where each component resides in the internet ecosystem, and issues a set of design recommendations for web map aesthetics and interactions.

Much of the material in Sections 2.1 and 2.2 has undergone peer review and is published as the *Web Mapping* entry of the GIScience and Technology Body of Knowledge (Sack, 2017). As described in Section 2.3, the Body of Knowledge serves as a living document—one that can be updated as cartographic technology and theory progress—and as an open educational resource, a document that provides free online instructional material. Its content is and will remain scholarly, yet it is structured more like a blog or wiki post than a traditional journal article, increasing its accessibility. In the current social media-driven information dissemination environment, such a sharable online resource can more quickly spread the necessary theoretical groundwork for teaching web mapping while promoting empirically-derived cartographic principles for web map construction.

Chapter 4 further contributes to the field of Cartography through better defining the cartographic process for designing and developing interactive web maps. Through its analysis of threshold concepts encountered by students, Section 4.6 extracts key skills necessary for mapping on the open web. The impact of this analysis on Cartography education—a subset of GIScience Education—is discussed further below.

6.2.2 Contributions to GIScience Education

This dissertation contributes to GIScience Education in three ways: by surveying current best practices in use for teaching web mapping skills, by providing example web mapping curricula that instructors may adapt to their own program contexts, and by delineating a set of learning outcomes necessary for mastery of Open Web mapping skills.

The interview study reported in Chapter 3 confirms that the constructivist pedagogies of active learning, scaffolding, and spiral curriculum described in Section 2.3 are indeed in use by web mapping instructors. It further shows that blended methods are also commonly used to teach web mapping. The study findings add weight to these pedagogies as sound instructional design strategies. The descriptions of scope and topics taught by web mapping instructors inform the set of suggested learning objectives for web mapping courses included in the Body of Knowledge entry described above, shown in Table 2.3. They are also distilled into a framework for matching the scope of a web mapping course to the appropriate types of web technologies, presented in Figure 3.1. Like the Cartography contributions discussed above, the findings and implications of the interview study are published in peer-reviewed form on an open access platform—*Cartographic Perspectives*—to encourage their dissemination among GIScience educators (Sack, 2018).

Another resource this dissertation provides to GIScience educators is multiple example curricula from which to draw ideas. The web mapping lab curriculum evaluated for Research Question 2 is presented in Section 4.3, with the topic sequence shown in Table 4.1. The description of this curriculum can inform GIScience educators looking to construct a class focused on Open Web Platform mapping skills. This curriculum is published along with most of the material covered by Chapter 4 as a peer reviewed article in the *Journal of Geography in Higher Education* (Sack and Roth, 2016). While not an open journal, this publication is accessible to many potential web mapping instructors through university journal subscriptions and available from the authors on demand. Revisions to the curriculum after its first use are discussed in Section 5.2, with the revised version presented in Table 5.2. Section 6.3 synthesizes the dissertation's research findings into a proposed web mapping curriculum for an intermediate-level college course with no programming prerequisite. The curricula presented in

Sections 5.2 and 6.3 have yet to be published, but open-access venues will be sought for their future publication.

Finally, the curriculum evaluation described in Chapter 4 resulted in a set of learning outcomes for web mapping that GIScience and Computer Science educators can use to measure the impact of their instructional strategies. The four outcomes—*Computational Thinking, Confluence, Competence,* and *Confidence*—provide a framework for understanding how students learn to develop web applications. Only one of these four skillsets—Computational Thinking—is recognized in the computer science education literature as necessary for coding, and definitions of it vary where it appears (see Section 2.4).

Table 6.1: A set of focused learning objectives that promote the four web mapping outcomes, derived from the Body of Knowledge Web Mapping learning objectives at the "Create" cognitive level.

Body of Knowledge "Create" Learning Objectives	
Design, construct, and publish an interactive web map.	
Format the styling, text, layout, image resolution, and file type of a static map so that it can be included in a well-designed web page.	
Publish a web map service or web map tile service.	
Focused Learning Objectives	**Outcome(s)**
Construct, publish, and share a customized thematic web map using a graphic online mapping platform	Competence
Publish a web map service and web feature service using a map server	Competence, Confluence
Design and embed in a web page a static map image that effectively represents a real-world problem or issue	Competence, Confluence
Construct a basic web page and publish it to a localhost server on their machine	Competence, Confluence
Construct a basic interactive web map using appropriately formatted data and HTML, CSS, and JavaScript	Competence, Confluence, Computational Thinking
Find and use online tutorials, examples, and resources to solve problems in program code	Competence, Confluence, Computational Thinking, Confidence

The recognition of the three additional outcomes suggests the need for GIScience educators and Computer Science educators more broadly to take them into account in their teaching strategies. Given that all four outcomes are situated at the "Create" level of Bloom's cognitive taxonomy, they can be applied to the Body of Knowledge web mapping learning objectives at that level to develop a more focused and specific set of learning objectives that promote one or more of the outcomes. A possible set of focused learning objectives is proposed in Table 6.1, with the relevant outcome(s) listed for each new objective.

6.2.3 Contributions to Online Education

This dissertation contributes an evaluation of blended pedagogy to discussions surrounding online education. As described in Section 2.5, several benefits have been proposed for blended learning environments over both traditional in-person instruction and fully online instruction. Chapter 5 describes the transitioning of the web mapping lab curriculum presented in Chapter 4 to a blended learning environment, and the evaluation of the new blended curriculum over two course offerings. Section 5.1 and Appendix 3 present a model for online delivery of complex technical material through a learning management system, which may help other online instructors format their instructional content.

One of the major findings presented in Section 5.5 is that the same curriculum that was highly successful in an in-person setting was seen by some students in the blended course as too overwhelming for the allotted time, and this appeared to negatively impact student confidence. This finding presents a cautionary note that appears to have been missed in the enthusiastic reporting of blended learning successes: shifting to a blended model requires at least some recalibration of the time commitment for each curriculum component. Chapter 5 has yet to be published, but will be submitted to a peer-reviewed outlet focused on online or blended teaching for publication.

6.3 Synthesis: A 'Zero-to-Map' Web Mapping Curriculum

The findings of the research reported in this dissertation can best be used moving forward to inform development of additional web mapping courses. This section discusses how the research findings have informed the creation of one such course, intended as a model for other GIScience educators to adopt in whole or part. The curriculum for the course is described here as a 'Zero-to-Map' curriculum because it is intended for intermediate undergraduate students with some exposure to GIScience concepts but little or no prior coding or web development experience. This curriculum will be taught by the author in the Spring of 2019 at a community college.

Fond du Lac Tribal and Community College, located in Cloquet, Minnesota, offers two-year Associate of Arts degrees covering general education transfer curriculum and Associate of Science degrees in subjects including Environmental Science, Nursing, Law Enforcement, Social Work, Electric Utility Technology, and Geographic Information Systems. The Associate of Science Degree in Geographic Information Systems requires 30 general education credits and 30 program and elective credits. Required courses include *Introduction to Statistics, Databases and Data Spreadsheets, Using GPS, Introduction to GIS, Applications in GIS, Cartography and Visualization, Remote Sensing of the Environment, Programming in GIS, Introduction to Digital Graphics, Web Mapping,* and a research or internship experience. The program also offers a 16-credit certificate requiring *Using GPS,* the two GIS courses, *Remote Sensing of the Environment,* and either *Cartography and Visualization* or *Web Mapping*. *Web Mapping* was added as a new requirement for the degree and option for the certificate in 2018.

The term 'Zero-to-Map' expresses the overall vision for the *Web Mapping* course. It is the same vision expressed by the most interview study participants: to prepare students for future GIS jobs (see Section 3.3.1). GIS graduates with programming and web mapping skills are in high demand (Woodruff, 2011; Underwood, 2013). Yet many GIS programs, including the program at Fond du Lac, are implemented as a one- or two-year course sequence, with little room for computer science prerequisites. The interview study found the lack of prior coding skills among students to be one of the greatest challenges faced by instructors (see Sections 3.3.7 and 3.4). A 'Zero-to-Map' curriculum thus begins with graphic applications requiring no prerequisite coding skills, and builds toward proficiency in creating a custom interactive web map. Its scope is web mapping as a standalone topic, without a broader thematic focus on critical geography, spatial analysis, or cartographic design. It does, however, assume foundational knowledge of the special nature of geographic information, and uses prior knowledge of geographic data as an entry point to the first unit of the course. It is thus designed to maximize development of the core web mapping skills atop foundational GIScience knowledge in one semester-length, four-credit course.

The *Web Mapping* course is intended to meet the learning objectives presented in the GIS&T Body of Knowledge Web Mapping topic, listed in Table 2.3, and the focused learning objectives listed in Table 6.1. Table 6.2 presents a planned sequence of topics and activities for the course. Following the format of the curriculum presented in Chapters 4 and 5, topics are divided into four, four-week units, with the first three each centered around a major lab assignment, and the fourth focused on a student-directed final project. The full course syllabus is included as Appendix 6.

Table 6.2: Web Mapping topic and activity sequence.

Week	Topic	Activities
1	Client-server architecture; definition and types of web maps; browser tools	Web Map Scavenger Hunt
2	Layer 1: Data—models, geometries, types, and levels	Lab 1—ArcGIS Online Map: Prepare thematic datasets from U.S. Census
3	Layer 2: Representation—symbolization, visual hierarchy, tilesets	Lab 1: Symbolize each data layer and choose an appropriate basemap
4	Layer 3: Interaction—stages, operators, interface affordances & feedbacks	Lab 1: Use Web AppBuilder to add interactions and publish the map
5	Storytelling on the web, responsive web design	Self-critique of Lab 1 Lab 2—Map Story Web Page: Determine theme, storyboard, and prepare ArcGIS layout
6	Raster image formatting and style guidelines for static web maps	Lab 2: Symbolize, label, and export PNG maps Codecademy HTML certificate due
7	Introduction to HTML; text editors; web reference guides; the DOM; browser Elements tab	Lab 2: Create a web page with embedded PNG maps, captions, and scrolling links
8	Introduction to CSS; browser styles sandbox; Midterm exam	Lab 2: Add CSS page styles and media queries Codecademy CSS certificate due
9	Projections on the Web; OGC web services; SLD stylesheets	Self-critique of Lab 2 Lab 3—Leaflet Slippy Map: Create GeoServer Web Map Service and Web Feature Service
10	Web directory setup; JavaScript data types, functions, methods; Console and debugging	Lab 3: Set up a localhost server and web directory Debugging Practice Assignment
11	JavaScript control flow; Leaflet API; GeoJSON; AJAX	Lab 3: Load Leaflet basemap, WMS, WFS layers Codecademy JavaScript certificate due
12	Leaflet interactions	Lab 3: Symbolize WFS layer, add pop-ups and layers control, finalize page styling
13	Concept review and final project assistance	Self-critique of Lab 3 Final Project—Published Web Map
14	Final project assistance	Final Project
15	Final project assistance and exam review	Final Project
16	Final project	Final Project
17	Final exam	Final Project due

The curriculum is intended to build toward proficiency in basic web mapping through a four-level curriculum spiral (Foote, 2011). Figure 6.1 is modeled on Figure 2.8 and visualizes the spiraling of course topics, with each instructional unit represented as a horizontal row, and the cross-cutting themes of Open Web technologies, data, representation, and interaction as columns.

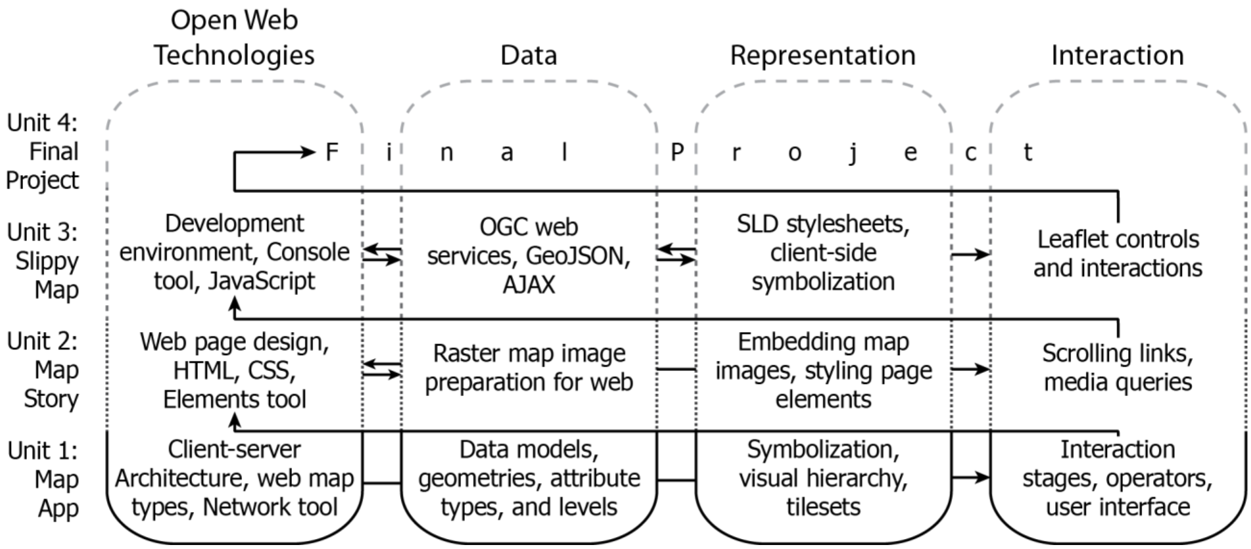


Figure 6.1: The curriculum spiral of the proposed Web Mapping course.

In Figure 6.1, the borders defining each column's theme start off solid and dark, representing the rigid divisions placed between themes to organize them within students' cognitive schemata during the first four weeks of the semester (Reigeluth, 2007). With each successive unit, these divisions become blurrier, as the themes become more integrated within the web mapping processes introduced. The arrows do not represent a perfectly aligned sequence. The parallel arrows between the left two columns in Unit 2, and between the left three columns in Unit 3, represent the iterative sequencing for those themes, in which topics from multiple themes may be introduced in the same week and topics from each theme may be spread across multiple weeks (see Table 6.2). However, each unit successively revisits, adds depth to, and builds bridges across all four themes, so that students can integrate the themes to independently complete a web mapping workflow in the fourth unit.

6.3.1 Unit 1: ArcGIS Online Web Map

The first unit of the course is designed to introduce students to web maps, closely following the concepts covered in the Web Mapping entry in the GIScience and Technology

Body of Knowledge (Sack, 2017) and Sections 2.1 and 2.2 of this dissertation. Week one of the course introduces students to the different types of web maps and their underlying structure, following the discussion in Section 2.1. In concert with the conceptual presentation, students are given a brief assignment requiring them to find two examples each of static and interactive maps, and describe the resources sent from server to client for each example using the browser developer tools Network tab. This serves not only to allow students to "see" client-server architecture at work, but also introduces them to the developer tools packaged with every major browser that are used heavily later in the course—following the finding of the Chapter 4 curriculum evaluation that students benefit from the early introduction of such tools (see Section 4.5). Weeks two, three, and four each focus on one of the layers of a web map, as described in Section 2.2, introducing core concepts related to each layer and to the overall Web Mapping Workflow (see Section 2.4).

The Unit 1 topic sequence is supported by weekly stages of the first major lab assignment, which walks students through creating an interactive web map using Esri's ArcGIS Online integrated Web GIS platform. Many of the advantages of using ArcGIS Online in a web mapping course are highlighted by the interview study described in Chapter 3, and are reviewed in Section 3.4. The platform provides an approachable first experience with web mapping for those who lack coding skills while building on foundational GIScience concepts. The ArcGIS ecosystem is already familiar to students who have taken an introductory GIS or Cartography course, and it is still the dominant platform in most GIS workplaces. A final advantage described by interview study participants is that it costs nothing beyond the price of an existing institutional site license, to which most collegiate GIScience programs have access.

While customization is limited within browser-based ArcGIS Online tools, there are enough built-in symbolization and interface design options to allow students some flexibility in the final products they create. The platform also effectively demonstrates the data-representation-interaction workflow by segregating each of these stages into distinct, sequential tasks: the Map Viewer application initially promotes adding one or more properly formatted geographic datasets, then walks the user through styling each new layer, and finally leverages a secondary application—the Web AppBuilder—to add customized interaction functionality and share the final web map (Figure 6.2).

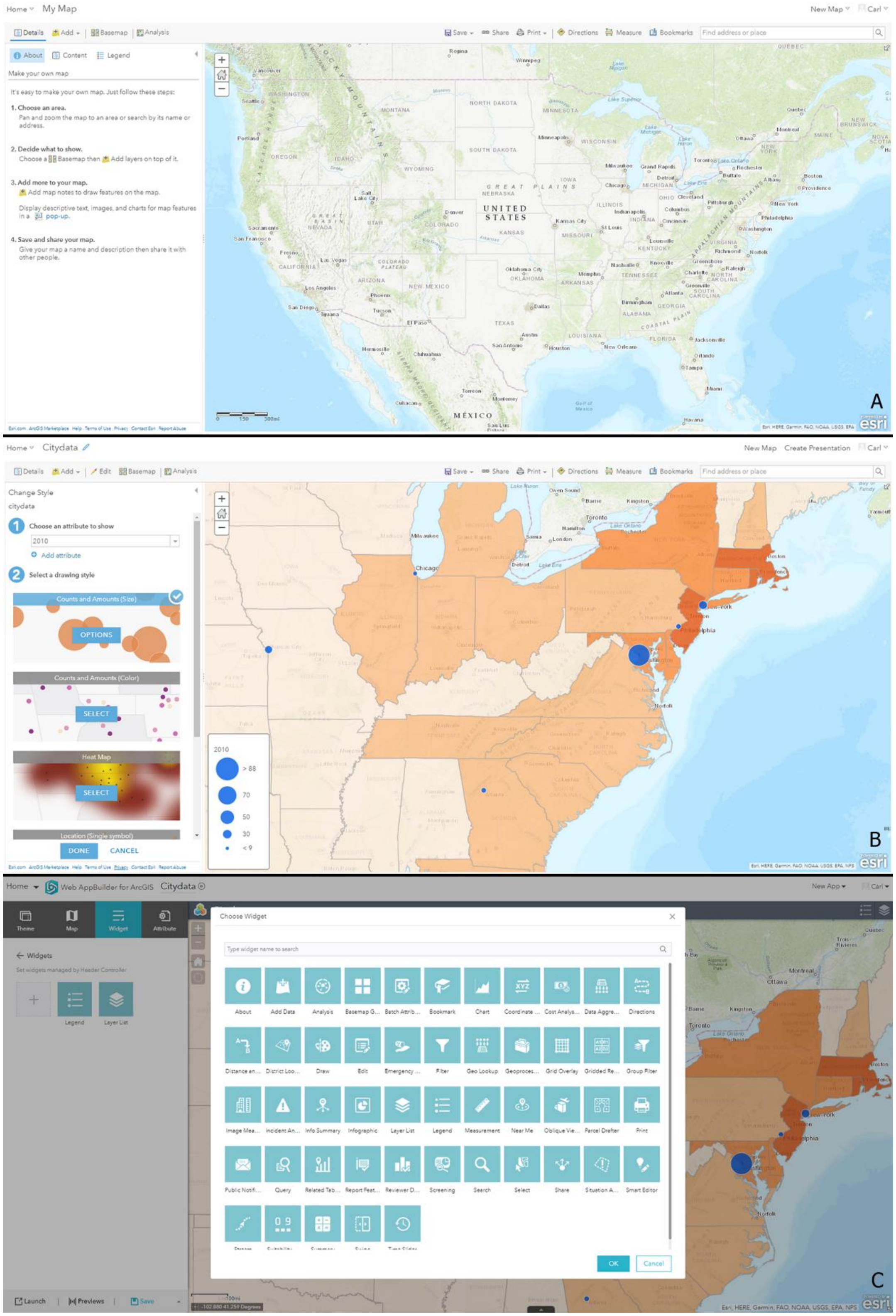


Figure 6.2: Screenshots of the ArcGIS Online platform showing A) the initial Map Viewer interface with data prompts in the left side panel, B) Map Viewer symbolization tools, and C) interaction widgets available to add to the map through the Web AppBuilder interface.

Dealing with data is generally the most time-consuming stage of any mapping project (Tolochko, 2016). As described in Section 5.5, data discovery and processing for each lab was found to be a major challenge for students in the Interactive Cartography and Geovisualization course, and rethinking data practices provided one of the best opportunities for reducing the stressful time crunch experienced by students. To maximize the available time for each lab assignment in the *Web Mapping* course, students use the same dataset for the first three major assignments. They collect, process, and format this dataset during Week 2, as a practical active learning exercise demonstrating the data concepts introduced that week. This topic sequencing has the added benefit of acting as a bridge between students' knowledge of geographic information concepts from prior GIScience courses and the application of that knowledge to the web mapping realm.

For the Week 2 assignment, students are required to select data from the U.S. Census Bureau's American FactFinder website (https://factfinder.census.gov/) covering any set of geographic units within the United States. The data must either come from one decadal census and cover five related quantitative attributes, or contain a single quantitative attribute measured across five decadal censuses. The data must also be able to convey a coherent story or message about the American people. These parameters are intended to make the initial data curation task clear and manageable, while giving students leeway to explore datasets that they find interesting and relevant, thus maintaining the real-world applicability of assignments valued by web mapping instructors and constructivism in general (see Sections 2.3, 3.3, and 4.1; Prager, 2011).

During the third week of the course, students are introduced to cartographic representation concepts as they apply to web maps. In addition to their core place in the Web Mapping Workflow, cartographic design principles remained among the web mapping topics most taught by interview study participants (see Table 3.4; Donohue, 2014). The specific concepts introduced in Week 3 include map symbolization through the visual variables, visual hierarchy for different web map types, and how representation is performed on the client (e.g., vector data layers) versus the server (e.g., web map and tile services; Muehlenhaus, 2014; Peterson, 2014). For their lab exercise, students are required to load and appropriately symbolize their Census dataset in the ArcGIS Online Map Viewer, and to choose a base map tileset that provides an appropriate level of underlying information density and visual contrast.

In Week 4, the final week of Unit 1, the focus moves to interaction, with discussion of Norman's Stages of Action model (see Figure 2.6), interaction operators (see Table 2.2), and interface affordances and feedbacks, key concepts included in the Interactive Cartography and Geovisualization curriculum (Norman, 1988; Roth, 2012). As the final stage of the lab assignment, students are required to use the ArcGIS Online Web AppBuilder to identify appropriate interaction operators for their map given its purpose and to add the accompanying interface elements (or *widgets* in ArcGIS Online terminology) to finalize their map. The early, direct connection between interaction concepts and widgets is intended bridge the gap between conceptual material on UI/UX design and technical material on web development identified by some students in the 2016 and 2017 curriculum evaluations (see Section 5.4.4).

6.3.2 Unit 2: Web Map Story

The second unit of the course revolves around storytelling with web maps. Almost half of the participants in the interview study made use of Esri's Story Maps platform, which provides highly usable tools for web map-based storytelling that integrate with ArcGIS Online. Many students may already be familiar with Story Maps from earlier Geography or GIScience courses (this is at least the case at Fond du Lac). While very applicable to data journalism and storytelling concepts, the Story Maps platform is ill-suited for moving students beyond the commercial graphic user interface and into the world of coding with Open Web technologies. The Story Maps Builder web interface enforces path dependency on ArcGIS Online, and while published as an open source project, the source code of a Story Maps application is far too complex to serve as an introduction to coding.

The second major lab assignment uses the story map *concept* as the backdrop for introducing students to online image publishing and web page design using basic HTML and CSS, without reliance on Esri tools. It requires students to create a responsive, scrollable web page from basic HTML elements and CSS styles. Through this process, students learn the core components of web page design, but are not yet required to deal with more complex scripting concepts. As many students reported finding online tutorials useful in the 2016 and 2017 curriculum evaluations, students in the *Web Mapping* course will be required to complete the Codecademy HTML and CSS tutorials in addition to the in-class lab assignments during the unit.

The first week of the unit (Week 5) covers basic storytelling concepts along with the principles and techniques of responsive web design. Neither of these topics is covered in great depth—each could constitute an entire separate college course if taught in full—but both are introduced at a sufficient level to allow students to envision how they will present their Census dataset in an insightful and impactful way. For their lab assignment, students write out the theme of their story, storyboard its constituent parts (including a lede, body, and conclusion), and prepare a layout for their dataset using familiar ArcGIS desktop software.

During Week 6, class discussion focuses on key cartographic design concepts for static web maps presented in Section 2.2, including symbolization, labeling, map elements, visual hierarchy, and image size, resolution, and data storage formats. This discussion adds detail to the representation layer of web maps first introduced in Week 2. The week’s lab deliverable is a complete set of appropriately-sized, screen resolution portable network graphic (PNG) map images for use in the student's map story web page.

Week 7 prepares students to enter the world of writing code. It introduces students to HTML, including its structure, core elements, and attributes, as well as the threshold concept of the DOM (see Section 4.5). It also covers two key elements of the web development environment: fully-featured text editors and the Elements or Inspector browser tool, which allows students to see how HTML elements are rendered by the browser. Finally, it introduces students to the online reference documentation, identified by students as a key resource in the 2016 and 2017 curriculum evaluations (see Table 5.7). For the Week 7 lab assignment, students are given an HTML document with template sections to edit, reflecting the common use of templates highlighted by the interview study (see Section 3.3.6). The week's deliverable is a basic web page that includes the static map images as well as supporting content and link tags to enable dynamic page scrolling.

The final week of the second unit (Week 8) introduces CSS concepts and demonstrates how styles are applied to HTML elements using both a text editor and the styles sandbox within the browser Elements tool. The use of the sandbox creates a visual link between the two Open Web languages that begins to build the concept of confluence (see Section 4.5). Students finalize their lab assignment by adding CSS style rules and some basic media queries that dynamically adjust page elements based on the client's screen size.

6.3.3 Unit 3: Leaflet Slippy Map

The third unit builds on the skills and concepts learned during the first two units while adding a new challenge: JavaScript. As with HTML and CSS in Unit 2, students are required to complete the online Codecademy JavaScript tutorial during Unit 3. The goal of the third major lab assignment is to build a full-stack, customized interactive web map using GeoServer to host geospatial web services and the Leaflet code library to provide representation and interaction control.

GeoServer (http://geoserver.org/) is an open source map server software package that handles requests for geospatial data and map images formatted according to Open Geospatial Consortium (OGC) standards (see Table 2.1). Two interview study participants used GeoServer to create web services, while nine used ArcGIS Server for the same task; however, the difficulty of setting up ArcServer was a common complaint, in contrast to GeoServer's straightforward installation, data ingestion, and layer configuration (Figure 6.3; see Sections 3.3.4 and 3.3.7). As introduced in Chapter 4, Leaflet provides a useful entry point to client-based web map coding by smoothly integrating HTML elements, CSS styles, and JavaScript tasks and interactions to render a web map (Figure 6.4). In addition to its use in the Interactive Cartography and Geovisualization curriculum, Leaflet was the most widely used FOSS web mapping library among interview study participants, implemented by 35% of them (see Table 3.5). Using GeoServer and Leaflet together provides students with practical experience creating client-server interactions and synthesizing the data, representation, and interaction layers of a web map.

The class discussion in Week 9 returns to the data layer, covering OGC web services and interactive web map projections, but mixes in some server-side representation through introduction of Styled Layer Descriptor (SLD) stylesheets, which are necessary to create a web map service. As discussed above, many interview participants saw geospatial web services, including both web map services (WMS) and web feature services (WFS), as important to include. For the first step of Lab 3, students use GeoServer to publish their data as both WFS and WMS, altering a template Styled Layer Descriptor (SLD) stylesheet to render the background geography of their dataset for the latter service.

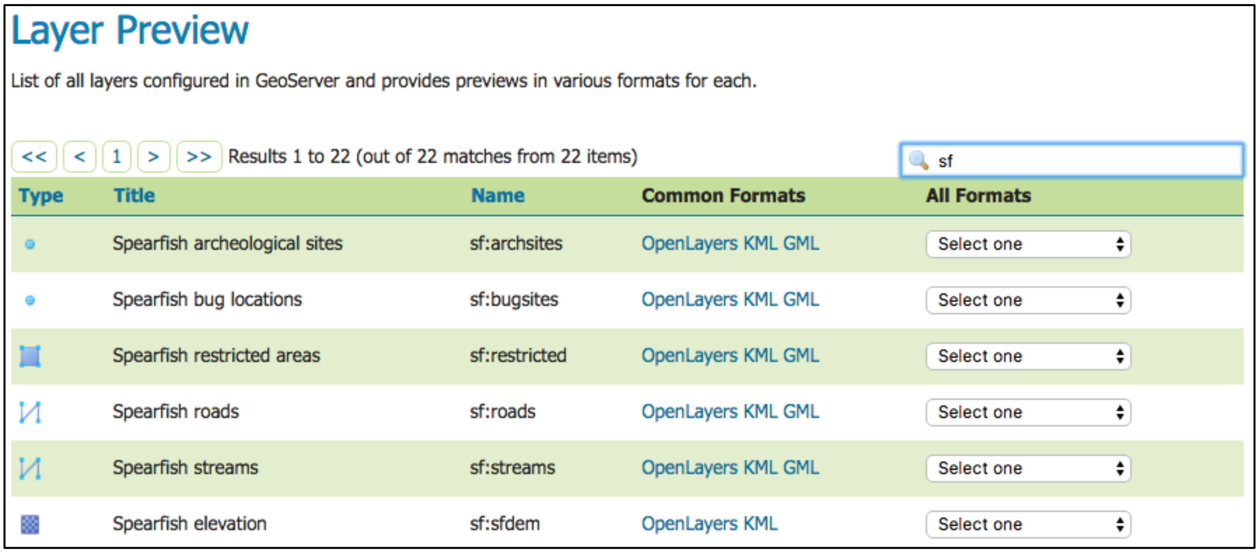


Figure 6.3: GeoServer layers preview interface, showing data layers published as geospatial web services (from the GeoServer documentation website, http://docs.geoserver.org/)

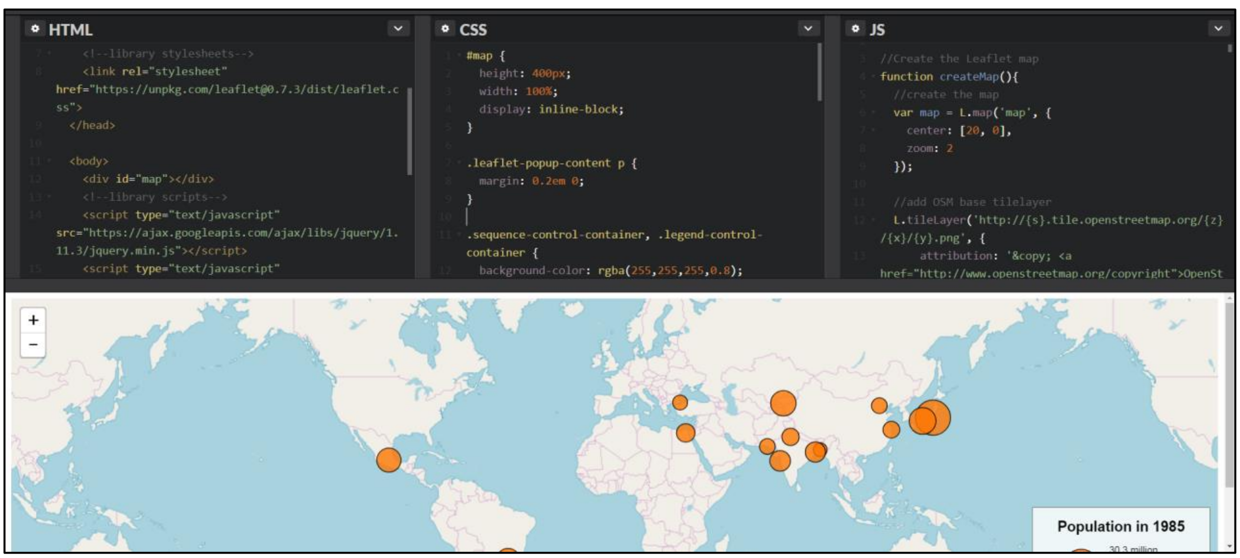


Figure 6.4: A CodePen (https://codepen.io) screenshot demonstrating how Leaflet integrates HTML, CSS, and JavaScript to create an interactive web map

The content for Week 10 steps back to the development environment, elaborating on the practices introduced in Weeks 7 and 8 by separating HTML, CSS, and JavaScript into separate markup, stylesheet, and script files within an organized web directory. It also begins to expose students to computational thinking in an Open Web context through discussion of core

JavaScript concepts and a lesson on debugging with the browser Console tool, an early threshold concept identified by the first curriculum evaluation study (see Section 4.5). For the week's lab exercise, students must set up a local server to host their web map and organize their website directory. They also are given a small assignment requiring them to debug a provided script to practice their problem-solving skills, similar to debugging assignments that students found useful in the 2016 and 2017 evaluations (see Section 5.4.5).

Week 11 introduces the Leaflet code library through its online tutorials and API documentation and explores critical enabling technologies—specifically the GeoJSON data format and the use of AJAX to dynamically load data into client memory. To reinforce AJAX—a key threshold concept identified by the first curriculum evaluation (see Section 4.5)—the lab assignment requires students to call and render WMTS (basemap tiles), WMS, and WFS layers using Leaflet. WMTS and WMS layers can each be loaded and rendered by a single Leaflet method, whereas WFS layers must first be loaded as GeoJSON data using a JavaScript AJAX function, then manipulated within a callback function to create Leaflet data layers, and finally styled within a function that symbolizes each feature based on the value of a selected attribute. The final week of Unit 3 introduces interactions and widgets available through Leaflet and reviews any prior concepts that need clarification. To complete the lab assignment, students add pop-ups and a layer control widget to the map and finalize the web page with supporting elements and CSS styles. Because custom interactions were found to be barriers to lab completion in each curriculum evaluation, they will not be required for the assignment (see Sections 4.5 and 5.5). However, Leaflet plugins will be introduced as tools that confident students may use to add functionality to their final projects.

6.3.4 Final Project and Assessment

The fourth and final unit of the course is dedicated to a final project, in which students use what they have learned so far to independently create an interactive web map from start to finish using an entirely new dataset that addresses a real-world issue or problem of their choice. Requiring a student-directed final project was seen as a core component of a web mapping course by the majority of interview study participants (see Sections 3.3.6 and 3.4). Final projects were also successful components of the curriculum discussed in Chapters 4 and 5. The project withdraws the scaffolding to allow students to exercise their skills independently and test their own capabilities (Palincsar, 1986).

During Week 13, the instructor provides a flexible, student-driven review of concepts that were presented earlier in the course to refresh students' memories and address any misconceptions. For the remainder of the course, the instructor provides open-ended assistance to students on an individual basis, while students apply their knowledge to complete the project. The project deliverable is a web page with a custom Leaflet map presenting a real-world issue or problem in which the student is interested.

Throughout the course, student learning is assessed using three mechanisms: weekly deliverable grades, lab assignment grades, and exams. Similar to the modular curriculum described in Chapter 5, weekly deliverable grades are assigned by the instructor based on whether the student has made adequate progress in completing each week's component of the unit's lab assignment. This allows the instructor to see how well students are progressing through the material and adjust the scaffolding accordingly. After completing each lab assignment, students are required to critique their work and give themselves a grade, thus promoting metacognition regarding their learning during the unit (Fouberg, 2013; Schultz, 2011). The use of peer critique was a pedagogical strategy highlighted as effective by multiple instructors in the interview study (see Section 3.3.6). Finally, students are given two comprehensive exams: a Midterm Exam and a Final Exam. Each exam reviews the vocabulary and concepts introduced in the class discussion and includes a practical component requiring students to analyze or critique a web map.

6.4 Future Directions

This dissertation explored three research questions related to teaching web mapping. Extending from the limitations of the research discussed in Section 6.1, the contributions discussed in Section 6.2, and the research synthesis proposed in Section 6.3, there are several possible directions for future research into instructional design for web mapping courses. This section closes the dissertation by suggesting four such possibilities: 1) addressing the needs of web mapping learners in diverse disciplinary, language, and physical ability contexts, 2) developing a framework for scoping web mapping curriculum, 3) evaluating the efficacy of fully online instruction for teaching web mapping, and 4) exploring methods for teaching web mapping in pre-collegiate and informal education settings.

6.4.1 Addressing Diversity in Web Mapping Education

As discussed in Section 6.1, the interview study described in Chapter 3 only recruited web mapping instructors from Anglophone colleges and universities in North America, where GIScience programs are commonly (though not exclusively) housed within Geography departments. This disciplinary setting entails a common population of students which has been described at various points in this dissertation as lacking in advanced computer science and coding skills (Muller and Kidd, 2014). Student populations differ in other settings. In institutions that teach GIScience within more technology-focused Engineering or Geomatics programs, might students require less content on basic coding concepts? What levels of achievement are possible in such programs, if the conceptual starting point of the student population is at a higher level?

Non-Anglophone institutions may also have a unique set of challenges in teaching web mapping related to the dominance of English as the basis of many Open Web Platform technologies. Open Web standards rely on high-level coding languages that use keywords based on English language syntax, and many commonly used web technologies are only supported with English-language APIs and documentation (Thompson, 2011). What types of instructional support are required to teach web mapping to learners for whom English is not their primary language? Are there certain tools and technologies that may supplement or replace those described in this dissertation to support non-English web mapping instruction?

A final, yet critical, area of diversity that should be addressed is how best to teach compliance with Web Content Accessibility Guidelines for web maps (W3C, 2016; see Section 2.2). Unfortunately, few interactive web maps can claim to be in full compliance with these guidelines, particularly those for the visually impaired (Carnegie Museums of Pittsburgh Innovation Studio, 2018). Affordances that enable screen readers to interpret text and static images on a web page are generally not available for map interactions that require using a mouse or finger on a graphic interface element and interpreting the change in the system visually. The best alternatives are text-based workarounds that do not capture the full interactive experience or visual complexity of a map, and even these workarounds are often missing from web pages that host interactive maps. How should web mapping curriculum integrate existing best practices for designing accessible web maps? How can non-visual (i.e., haptic and/or auditory) affordances and feedbacks be integrated into web maps to increase their accessibility? What new collaborations can cartographers develop with product engineers and computer scientists to forward web map accessibility?

6.4.2 Developing a Scoping Framework for Web Mapping Curriculum

A second area for future research involves the development of a framework for web mapping curriculum scope based on program vision. As defined in Section 3.3, *vision* describes the big-picture social or academic role that the instructor imagines a web mapping course to fulfill. The vision largely determines the *scope* of the course, or the breadth and depth of topics taught within a curriculum (Foote, 2011). Section 3.4 delineated four general scope categories for web mapping courses currently being taught: standalone web mapping, web mapping with critical theory, web GIS, and web cartography. This is one possible framework for course scope, but not a prescriptive one.

Further research is needed to develop a prescriptive framework that suggests an effective scope, along with critical concepts to be taught, based on the vision of the program to which the course belongs. One such framework is proposed by Ricker and Thatcher (2017), but is not specific to web mapping. Such a framework would answer questions such as, how should the scope of a web mapping course differ between a program intended to output government and small enterprise GIS technicians who might program the occasional web map, one intended for students seeking Cartography jobs, and one for academic Geographers seeking Ph.D.s and future careers in research? What course topics are necessary to cover the knowledge base required for each vision? Such research might involve interviews or focus groups with different kinds of employers as an empirical basis for delineating key skills required for different career paths.

6.4.3 Evaluating Fully Online Web Mapping Curricula

Third, the curriculum evaluation described in Chapter 5 was originally intended to extend the evaluation described in Chapter 4 to a comparison of outcomes for blended in-person and fully online distance education versions of the Interactive Cartography and Geovisualization course. However, extenuating circumstances prevented the evaluation of a fully online distance education version of the course. Online distance education differs significantly from blended courses in that it is usually fully asynchronous, with no defined lab period involving in-person instruction or assistance (Allen and Seaman, 2013). Students in fully online courses may still receive real-time instructor assistance through videoconferencing software, but usually on a one-on-one, unstructured basis. In general, academic leaders see blended instruction as producing superior outcomes to fully online courses (Allen et al., 2016). This suggests a need to evaluate the learning outcomes of web mapping curricula in a fully online environment before fully online delivery can be assumed to be as successful as delivery in a blended environment.

As discussed in Section 6.1, confounding variables might be avoided in such an evaluation by focusing more narrowly on the outcomes of individual assignments that promote one learning outcome from the set of four identified in Chapter 4 (*computational thinking, competence, confluence,* and *confidence*). Evaluation instruments could be designed with an even more granular focus on the individual sub-components of each outcome described in Chapter 5. They could be tied to new assessments that better integrate course lecture topics with laboratory skills aimed at the four learning outcomes, such as exams that include a debugging component, or lab exercises that require critique and improvement of an existing interface design. In a fully online setting, what activity formats best support learning in each of the four web mapping outcomes? What fully online instructional strategies can reproduce the support of in-person instruction for learning the technical skills of web mapping?

6.4.4 Expanding the Domain of Web Mapping Education

Finally, the research reported in this dissertation focused on teaching web mapping in higher education. However, there are increasing calls for computer programming skills to be taught at the secondary and even the primary levels (Brennan and Resnick, 2012; Raja, 2014). Interactive web maps have a high degree of public visibility, practicality, and usability, making them appropriate media for the introduction of web development skillsets to age groups below the college level. In the U.S., few secondary schools require students to take a computer science course, but computer science can be taken as an advanced placement course in high schools that have the faculty skills, time availability, resources, and student interest to make those courses available. Further research could develop pilot web mapping units or courses for elementary, middle, and high school levels, thus promoting the broader inclusion of both spatial thinking and web development skills at these developmental levels. Instructional designers could examine such questions as, how can web mapping promote the core computer science skills identified in the advanced placement computer science curriculum standards? What is an appropriate developmental level at which to first introduce web mapping concepts to young students? What current K-12 educational standards can be effectively met through the inclusion of web mapping? What new K-12 standards might promote web mapping while benefitting students through exposure to 21st-Century STEM skills?

Other ages and levels of learners in addition to K-12 and college students may benefit from exposure to web mapping skills. Informal education settings include pre-conference workshops at GIScience, Geography, and Cartography conferences, professional trainings for companies, government agencies, researchers, and educators, and workshops offered to the interested public. How can web mapping be most effectively taught in such informal education settings? Can the 'Zero-to-Map' approach introduced in Section 6.2 be applied to a distilled workshop curriculum taught over the course of a week, or perhaps even a day? What key concepts and tools are most useful to include in such trainings? How might domain users of Cartography and GIS benefit from professional development training in web mapping concepts, tools, and technologies?

6.5 Finale

This dissertation points the way forward for GIScience programs and educators interested in developing web mapping course offerings, and encourages the implementation of such offerings. Web mapping is an increasingly essential skillset for GIS and Cartography careers. It can be the focus of a course that promotes exposure to code-based Open Web Platform technologies, with or without critical inquiry into their epistemologies; or it can be used as a supplement to broader GIS and Cartography concepts through reliance on full-stack commercial mapping platforms. When taught using Open Web technologies, web mapping includes four key competencies—computational thinking, competence, confluence, and confidence—that are best supported through constructivist pedagogy, particularly the instructional design techniques of scaffolding, and spiral curriculum. It can be taught in person or in a blended setting through modular online delivery with similar results, though modular online delivery can add to the time required to convey concepts. Finally, a 'Zero-to-Map' approach can be used to conceptualize new curriculum that effectively conveys the core competencies necessary for web mapping to students without prior coding background. This research should enable future exploration into the most effective instructional design considerations for various settings and formats for teaching web mapping.

Glossary

Accessibility: the ability of people with disabilities, the elderly, and residents of rural areas and developing countries to use websites and web applications

Active learning: an instructional practice that directly engages students in hypothesis testing and problem-solving activities

Affordances: clues embedded in a user interface that reveal how the user can interact with it

Animated web map: a web map that changes frequently and automatically, using time to represent one or more data attributes

Application programming interface (API): a set of instructions used by computer programs used to communicate with one another

Asynchronous JavaScript and XML (AJAX): a set of procedures enabled by JavaScript code that allows a client to send requests to a server and receive data without reloading the web page

Canvas: an HTML element that uses hardware acceleration to support fast rendering of vector data as raster images; also the brand name of a learning management system

Cascading Style Sheets (CSS): the Open Web Platform code specification used to style web pages

Client: a software program that requests and accepts information from a server

Client-server architecture: a model of communication across networks involving a server software program that handles client requests

Comma-Separated Values (CSV): A simple text-based spreadsheet data storage format

Competence: the ability to understand and use a particular piece of software or technology

Computational thinking: the thought processes involved in understanding the structure and information processing functions of a computer program

Confidence: the expectation of eventual success in creating a working computer program

Confluence: the ability to understand what methods are available in a given Open Web technology and how different technologies work together to produce a working program

Constructivism: an educational philosophy that emphasizes direct experience in authentic learning environments, conceptualization of new information as cognitive schemata, and metacognition

Debugging: the process of solving problems that prevent the correct execution of a computer program

Development environment: the set of coding tools, debugging tools, and development software necessary to make a web map

Document Object Model (DOM): the collection and organization of elements, attributes, styles, data, script objects, and procedures that compose a web page

Dynamic web map: a map image that changes appearance as it is viewed

Feedback: the signals that a user interface gives to the user to reveal the result of an interaction

Flexibility: the number of interface components that can be used to implement the same interaction

Free and open source software (FOSS): programs and technologies that give users the freedom to run, copy, distribute, study, change, and improve them without notifying or paying royalties to prior distributors

Freedom: the degree of precision or finesse with which the user can implement a certain interaction

GeoJSON: a geospatial variant of JSON that stores vector data as sets of vertex coordinates (see *JavaScript Object Notation*)

Hyper-Text Markup Language (HTML): the Open Web Platform code specification used to create documents for rendering in the browser

Interaction: user actions that change a map and the system response to those actions

Interactive web map: a web map that changes in response to user input

Interface complexity: the number of unique map views a user can create in an interactive web map

Interface scope: the total number of interaction operators available in a web map

Internet Protocol address (IP address): a string of numbers and periods assigned to a machine that allows it to be located on a network by other machines

JavaScript: the Open Web Platform code specification used to script animations and interactions on web pages

JavaScript Object Notation (JSON): a JavaScript-based data storage format

Learning management system (LMS): a web-based platform for delivery of instructional content

Learning outcomes: measurable cognitive processes or tasks that students should be able to accomplish on their own after completing the course

Map composition: the way mapped information is seen and interpreted by the user (see also representation)

Map layout: the visual arrangement of interface elements on and around a web map

Map mashup: a web map that combines underlying map tiles with image overlays from separate data sources

Map server: a specialized piece of server software designed to transmit pre-rendered map images and raw spatial data

Metacognition: reflection by the learner on their learning processes

Module: a segment of online instructional content presented in a standardized and self-contained format

Open Educational Resource (OER): instructional materials that are made publicly available free of charge through the internet

Open Web Platform (Open Web): the set of royalty-free technologies and standards that power the internet

Open Web standards: a set of specifications for web programming languages and data handling processes published and maintained by the World Wide Web Consortium (W3C)

Overdesign: a map or information graphic state of containing an overwhelming quantity of data, elements, and/or interactions given its purpose

Pedagogy: the set of instructional design principles guiding curriculum structure and learning activities based on what is believed will most effectively accomplish learning objectives

Raster data model: spatial data consisting of a continuous grid of cell or pixel values following a field-based ontology

Reference web map: a web map that contains many datasets visualized to support wayfinding, location-based services, feature search, and landscape interpretation

Representation: the way mapped information is seen and interpreted by the user (see also map composition)

Scaffolding: an instructional practice that moves from direct instruction and heavy learner support when a new concept is introduced to progressively less instructor support as the learner becomes increasingly able to work on the concept independently

Scalable Vector Graphics (SVG): an open web standard for vector images that can be rendered directly in the browser

Scope: the depth and breadth at which course concepts are introduced

Sequence: the ordering of concepts or topics introduced in a course

Server: a software program that sends information stored in memory to a remote device (the client)

Slippy map: a web map that allows uninterrupted panning and zooming through the use of map tiles sent to the client as needed without reloading the web page (see *tiled web map*)

Spiral curriculum: a set of topics sequenced to build upon prior concepts at each new level of challenge

Static web map: a map image rendered in a web browser that does not change given user input

Stop-frame animation: animation composed of many individual images (or frames) appearing in rapid succession

Thematic web map: a web map that presents a small number of curated datasets with a specific intended message to the user

Threshold concept: a concept that transforms a student's way of understanding, interpreting, or viewing a subject matter and must be internalized before the learner can progress toward further mastery

Tiled web map: a web map that receives image or data tiles from the server for only those portions of the map viewed by the user (see *slippy map*)

TopoJSON: a geospatial variant of JSON that stores vector data topology (see *JavaScript Object Notation*)

Tweening: a smooth transition between two visual states created by program instructions

UI design: the iterative decisions made regarding the user interface

User experience (UX): the set of interactions enabled for a given web page

User interface (UI): the set of elements on a web page that the user can see and/or manipulate

UX design: the iterative set of decisions regarding user interactions leading to the user experiencing successful outcomes

Vector data model: spatial data consisting of discrete points, lines, and polygons following an object-based ontology

Visual hierarchy: the relative visual dominance of objects on a map

Web cartography: the visual design of web maps

Web geovisualization: a highly exploratory and interactive web map-based application with limited, pre-selected datasets

Web GIS: a highly interactive, highly flexible web mapping application that allows users to load their own datasets, perform spatial analysis, and create custom data visualizations

Web map: a map that is published and accessed via the internet

Web Mapping Workflow: the process of constructing an interactive web map from start to finish and the core competencies necessary to do so

Web mapping: the process of designing and developing an interactive web map

WebGL: a JavaScript API that uses hardware acceleration to support fast rendering of vector data into Canvas raster images

Widget: a user interface element tied to a specific data graphic or map interaction

Zone of proximal development: a model of learning that describes the cognitive gap between a student's current understanding or abilities and the student's potential given instructor and peer assistance

References

Abrahams, D. A. (2010). Technology adoption in higher education: a framework for identifying and prioritising issues and barriers to adoption of instructional technology. Journal of Applied Research in Higher Education, 2(2), 34–49. https://doi.org/10.1108/17581184201000012

Adams Becker, S., Cummins, M., Davis, A., Freeman, A., Hall Giesinger, C., & Ananthanarayanan, V. (2017). NMC Horizon Report: 2017 Higher Education Edition (p. 56). The New Media Consortium.

Adobe Corporate Communications. (2015, November 30). Flash, HTML5 and open web standards [Blog]. Retrieved March 10, 2017, from https://blogs.adobe.com/conversations/2015/11/flash-html5-and-open-web-standards.html

Allen, I. E., & Seaman, J. (2013). Changing Course: Ten Years of Tracking Online Education in the United States (p. 47). Babson Park, MA: Babson Survey Research Group, Quahog Research Group, LLC.

Allen, I. E., Seaman, J., Poulin, R., & Straut, T. T. (2016). Online Report Card: Tracking Online Education in the United States (p. 62). Babson Survey Research Group, Quahog Research Group, LLC.

Anderson, L. W., & Krathwohl, D. R. (Eds.). (2001). A taxonomy for learning, teaching, and assessing: a revision of Bloom's taxonomy of educational objectives (Complete ed). New York: Longman.

Ballatore, A., Tahir, A., McArdle, G., & Bertolotto, M. (2011). A comparison of open source geospatial technologies for web mapping. International Journal of Web Engineering and Technology, 6(4), 354. https://doi.org/10.1504/IJWET.2011.043440

Bampton, M. (2012). Addressing misconceptions, threshold concepts, and troublesome knowledge in GIScience education. In D. J. Unwin, K. E. Foote, N. J. Tate, & D. DiBiase (Eds.), Teaching Geographic Information Science and Technology in Higher Education (pp. 117–132). West Sussex, UK: Wiley-Blackwell.

Baran, E., Correia, A.-P., & Thompson, A. (2011). Transforming online teaching practice: critical analysis of the literature on the roles and competencies of online teachers. Distance Education, 32(3), 421–439. https://doi.org/10.1080/01587919.2011.610293

Barnes, T. J. (2004). Placing ideas: genius loci, heterotopia and geography's quantitative revolution. Progress in Human Geography, 28(5), 565–595. https://doi.org/10.1191/0309132504ph506oa

Battersby, S. E., Finn, M. P., Usery, E. L., & Yamamoto, K. H. (2014). Implications of Web Mercator and Its Use in Online Mapping. Cartographica: The International Journal for Geographic Information and Geovisualization, 49(2), 85–101. https://doi.org/10.3138/carto.49.2.2313

Berners-Lee, T. (2014, August 23). Tim Berners-Lee on the Web at 25: the past, present and future. Wired. Retrieved from http://www.wired.co.uk/article/tim-berners-lee

Bloom, B. S. (1956). Taxonomy of educational objectives: the classification of educational goals (Vol. 1). New York: Longmans, Green.

Bose, P. S. (2014). Technofetishism and online education: globalizing geography through virtual worlds. Journal of Geography in Higher Education, 38(1), 28–39. https://doi.org/10.1080/03098265.2013.801070

Bostock, M., & Davies, J. (2013). Code as Cartography. The Cartographic Journal, 50(2), 129–135. https://doi.org/10.1179/0008704113Z.00000000078

Bostock, M., Ogievetsky, V., & Heer, J. (2011). D3: Data-Driven Documents. IEEE Transactions on Visualization and Computer Graphics, 17(12), 2301–2309. https://doi.org/10.1109/TVCG.2011.185

Bozkurt, A., Akgun-Ozbek, E., Yilmazel, S., Erdogdu, E., Ucar, H., Guler, E., … Aydin, C. H. (2015). Trends in Distance Education Research: A Content Analysis of Journals 2009-2013. International Review of Research in Open and Distributed Learning, 16(1), 330–363.

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In Proceedings of the 2012 annual meeting of the American Educational Research Association. Vancouver. Retrieved from http://scratched.gse.harvard.edu/ct/files/AERA2012.pdf

Bruner, J. S. (1977). The process of education. Cambridge: Harvard University Press.

Buckingham, W. R., & Dennis Jr., S. F. (2009). Cartographies of Participation: How the changing natures of cartography has opened community and cartographer collaboration. Cartographic Perspectives, 64, 55–61.

Buckler, C. (2016, September 2). Browser Trends September 2016: Are the Browser Wars Over? [blog]. Retrieved May 23, 2017, from https://www.sitepoint.com/browser-trends-september-2016-browser-wars/

Bureau of Labor Statistics. (2017). Occupational Employment Projections Table 2016-2026. Retrieved from https://data.bls.gov/projections/occupationProj

Caballero, L. (2011, October 13). An Introduction to WebGL - Part 1. Retrieved March 17, 2017, from https://dev.opera.com/articles/introduction-to-webgl-part-1/

Carnegie Museums of Pittsburgh Innovation Studio. (2018). Maps. Retrieved August 6, 2018, from http://web-accessibility.carnegiemuseums.org/content/maps/

Case, D. E., & Davidson, R. C. (2011). Accessible online learning. New Directions for Student Services, 2011(134), 47–58. https://doi.org/10.1002/ss.394

Cerba, O., & Cepicky, J. (2012). Web Services for Thematic Maps. In M. P. Peterson (Ed.), Online Maps with APIs and WebServices (pp. 141–155). Berlin, Heidelberg: Springer.

Chickering, A. W., & Gamson, Z. F. (1987). Seven Principles for Good Practice in Undergraduate Education. American Association for Higher Education Bulletin, 39(7), 3–7.

Crampton, J. W. (2010). Mapping: a critical introduction to cartography and GIS. Malden, Mass: Wiley-Blackwell.

Crawford-Ferre, H. G., & Wiest, L. R. (2012). Effective Online Instruction in Higher Education. The Quarterly Review of Distance Education, 13(1), 11–14.

DeMers, M. N. (2009). Using Intended Learning Objectives to Assess Curriculum Materials: the UCGIS Body of Knowledge. Journal of Geography in Higher Education, 33(sup1), S70–S77. https://doi.org/10.1080/03098260903033980

DiBiase, D. (1990). Visualization in the Earth Sciences. Earth and Mineral Sciences, 59(2). Retrieved from http://www.geovista.psu.edu/publications/others/dibiase90/swoopy.html

DiBiase, D. (2012). GIS&T in the open Educational Resources Movement. In D. J. Unwin, K. E. Foote, N. J. Tate, & D. DiBiase (Eds.), Teaching Geographic Information Science and Technology in Higher Education (pp. 421–437). Chichester, UK: John Wiley & Sons, Ltd. Retrieved from http://doi.wiley.com/10.1002/9781119950592.ch27

DiBiase, D., DeMers, M., Johnson, A., Kemp, K., Luck, A. T., Plewe, B., & Wentz, E. (Eds.). (2006). Geographic Information Science and Technology Body of Knowledge (First). Washington, D.C.: Association of American Geographers.

DiBiase, D., & Kidwai, K. (2010). Wasted on the Young? Comparing the Performance and Attitudes of Younger and Older US Adults in an Online Class on Geographic Information: JGHE Annual Lecture. Journal of Geography in Higher Education, 34(3), 299–326. https://doi.org/10.1080/03098265.2010.490906

Donohue, R. G. (2014). Web Cartography with Web Standards: Teaching, Learning, and Using Open Source Web Mapping Technologies (Dissertation). University of Wisconsin-Madison, Madison, WI.

Donohue, R. G., Sack, C. M., & Roth, R. E. (2014). Time Series Proportional Symbol Maps with Leaflet and jQuery. Cartographic Perspectives, (76), 43–66. https://doi.org/10.14714/CP76.1248

Ellis, H. J. C. (2003). Transfer of knowledge in a web design and development course (Vol. 3, p. S1C_15-S1C_20). IEEE. https://doi.org/10.1109/FIE.2003.1265911

Foote, K. E. (2011). Scope and sequence in GIS&T education: learning theory, learning cycles and spiral curricula. In D. J. Unwin, K. E. Foote, N. J. Tate, & D. DiBiase (Eds.), Teaching Geographic Information Science and Technology in Higher Education (pp. 81–95). West Sussex, UK: Wiley-Blackwell.

Forrest, C. (2015, November 16). When it comes to jobs, Generation Z may not be the “tech” generation after all. Retrieved June 13, 2018, from https://www.techrepublic.com/article/when-it-comes-to-jobs-generation-z-may-not-be-the-tech-generation-after-all/

Fouberg, E. H. (2013). “The world is no longer flat to me”: student perceptions of threshold concepts in world regional geography. Journal of Geography in Higher Education, 37(1), 65–75. https://doi.org/10.1080/03098265.2012.654467

Free Software Foundation. (2016, January 1). What is free software? GNU Operating System. Retrieved from http://www.gnu.org/philosophy/free-sw.html

Fu, P. (2015). Getting to know web GIS. Redlands, California: Esri Press.

Gaff, B. M., & Ploussios, G. J. (2012). Open Source Software. Computer, 45(6), 9–11. https://doi.org/10.1109/MC.2012.213

Gardner, L. (2016). Where Does the Regional State University Go From Here? Chronicle of Higher Education, 62(37).

Garrison, D. R., & Kanuka, H. (2004). Blended learning: Uncovering its transformative potential in higher education. The Internet and Higher Education, 7(2), 95–105. https://doi.org/10.1016/j.iheduc.2004.02.001

Garrison, D. R., & Vaughan, N. D. (2008). Blended learning in higher education: framework, principles, and guidelines (1st ed). San Francisco: Jossey-Bass.

Gibson, R., & Erle, S. (2006). Google Maps hacks (1st ed). Beijing ; Sebastopol, CA: O’Reilly.

Glazer, F. S. (2012). Introduction. In F. S. Glazer (Ed.), Blended learning: Across the disciplines, across the academy (pp. 1–12). Stylus Publishing.

Goett, J. A., & Foote, K. E. (2000). Cultivating Student Research and Study Skills in Web-based Learning Environments. Journal of Geography in Higher Education, 24(1), 92–99. https://doi.org/10.1080/03098260085162

Goodchild, M. F. (2007). Citizens as sensors: the world of volunteered geography. GeoJournal, 69(4), 211–221. https://doi.org/10.1007/s10708-007-9111-y

Haklay, M. (2010). Interacting with geospatial technologies. Chichester, West Sussex, UK; Hoboken, NJ: John Wiley. Retrieved from http://public.eblib.com/EBLPublic/PublicView.do?ptiID=514453

Hallgren, K. A. (2012). Computing Inter-Rater Reliability for Observational Data: An Overview and Tutorial. Tutorials in Quantitative Methods for Psychology, 8(1), 23–34. https://doi.org/10.20982/tqmp.08.1.p023

Harley, J. B. (1989). Deconstructing the map. Cartographica: The International Journal for Geographic Information and Geovisualization, 26(2), 1–20. https://doi.org/10.3138/E635-7827-1757-9T53

Harrower, M. (2004). A Look at the History and Future of Animated Maps. Cartographica: The International Journal for Geographic Information and Geovisualization, 39(3), 33–42. https://doi.org/10.3138/7MN7-5132-1MW6-4V62

Harrower, M., & Brewer, C. A. (2003). ColorBrewer.org: An Online Tool for Selecting Colour Schemes for Maps. The Cartographic Journal, 40(1), 27–37. https://doi.org/10.1179/000870403235002042

Harvey, F. (2012). To Volunteer or to Contribute Locational Information? Towards Truth in Labeling for Crowdsourced Geographic Information. In Crowdsourcing geographic knowledge volunteered geographic information (VGI) in theory and practice (pp. 31–42). Springer.

Hermansen, S. (2010). Teaching Cartography in Academia: A Historical Reflection and Discussion of a 2007 Survey of Canadian Universities. Cartographica: The International Journal for Geographic Information and Geovisualization, 45(1), 5–18. https://doi.org/10.3138/carto.45.1.5

Jancer, M. (2017, November). Mapping the Future: Cartography Stages a Comeback. Wired, 25(11), 1.

Jesdanun, A. (2015, December 7). Apple Maps, once a laughinstock, now dominates iPhones. Associated Press. Retrieved from http://bigstory.ap.org/article/df90458e58564f19b4b7c8510f9baa67/apple-maps-once-laughingstock-now-dominates-iphones

Jobs, S. (2010, April). Thoughts on Flash. Retrieved August 20, 2015, from http://www.apple.com/hotnews/thoughts-on-flash/

Johnson, L., Adams Becker, S., Estrada, V., & Freeman, A. (2015). NMC Horizon Report: 2015 Higher Education Edition. Austin Texas; [S.l.: New Media Consortium ; EDUCAUSE Learning Initiative. Retrieved from http://cdn.nmc.org/media/2015-nmc-horizon-report-HE-EN.pdf

Lewy, A. (Ed.). (1977). Handbook of curriculum evaluation. Paris : New York: Unesco ; Longman.

Lienert, C., Jenny, B., Schnabel, O., & Hurni, L. (2012). Current Trends in Vector-Based Internet Mapping: A Technical Review. In M. P. Peterson (Ed.), Online Maps with APIs and Web Services (pp. 23–36). Berlin-Heidelberg: Springer.

Longley, P., Goodchild, M. F., Maguire, D. J., & Rhind, D. W. (2015). Geographic information science & systems (Fourth edition). Hoboken, NJ: Wiley.

Luo, H., Robinson, A. C., & Detwiler, J. (2014). Effect of Geographic Distance on Distance Education: An Empirical Study. Journal of Educators Online, 11(3). Retrieved from http://www.thejeo.com/Archives/Volume11Number3/LuoRobinsonDetwiler.pdf

MacEachren, A. M. (1994). Visualization in modern cartography: Setting the Agenda. In A. M. MacEachren & D. R. F. Taylor (Eds.), Visualization in Modern Cartography (pp. 1–12). Oxford, England: Pergamon.

MacEachren, A. M., & Kraak, M.-J. (1997). Exploratory Cartographic Visualization: Advancing the Agenda. Computers and Geosciences, 23(4), 335–343.

MacEachren, A. M., & Kraak, M.-J. (2001). Research Challenges in Geovisualization. Cartography and Geographic Information Science, 28(1), 3–12. https://doi.org/10.1559/152304001782173970

Marcotte, E. (2011). Responsive web design. New York, NY: A Book Apart.

McCormick, B. (2015, September 13). ES5, ES6, ES2016, ES.Next: What's going on with JavaScript versioning? [Blog]. Retrieved May 17, 2017, from https://benmccormick.org/2015/09/14/es5-es6-es2016-es-next-whats-going-on-with-javascript-versioning/

Meece, J. L. (2002). Child and adolescent development for educators (2nd ed). Boston: McGraw-Hill.

Merriam, S. B. (1988). Case study research in education: a qualitative approach (1st ed). San Francisco: Jossey-Bass.

Meyer, J., & Land, R. (2003). Threshold Concepts and Troublesome Knowledge: Linkages to Ways of Thinking and Practising within the Disciplines (Occasional Report No. 4). Edinburgh, UK: University of Edinburgh.

Miles, M. B., Huberman, A. M., & Saldaña, J. (2014). Qualitative data analysis: a methods sourcebook (Third edition). Thousand Oaks, Califorinia: SAGE Publications, Inc.

Mitchell, M., Palacios, V., & Leachman, M. (2014). States are still funding higher education below pre-recession levels (p. 27). Washington, D.C.: Center on Budget and Policy Priorities.

Molnar, M. (2015, June 12). U.S. Millenials Know Technology, But Not How to Solve Problems With It, Study Says. Retrieved June 13, 2018, from http://blogs.edweek.org/edweek/DigitalEducation/2015/06/us_millennials_know_technology.html

Moser, F. Z. (2007). Faculty Adoption of Educational Technology. Educause Quarterly, 30(1), 66–69.

Muehlenhaus, I. (2014). Web cartography: map design for interactive and mobile devices. Boca Raton, FL: CRC Press.

Muller, C. L., & Kidd, C. (2014). Debugging geographers: teaching programming to non-computer scientists. Journal of Geography in Higher Education, 38(2), 175–192. https://doi.org/10.1080/03098265.2014.908275

Mundkur, A., & Ellickson, C. (2012). Bringing the Real World in: Reflection on Building a Virtual Learning Environment. Journal of Geography in Higher Education, 36(3), 369–384. https://doi.org/10.1080/03098265.2012.692073

NCER. (2014). Enrollment in Distance Education Courses, by State: Fall 2012 (Web Tables No. NCES 2014-023) (p. 43). Washington, D.C.: National Center for Education Statistics. Retrieved from http://nces.ed.gov/pubsearch/pubsinfo.asp?pubid=2014023

Neisser, U. (1976). Cognition and reality: principles and implications of cognitive psychology. San Francisco: Freeman.

Norman, D. A. (1988). The psychology of everyday things. New York: Basic Books.

Ooms, K., & Skarlatidou, A. (2018). Usability Engineering and Evaluation. Geographic Information Science & Technology Body of Knowledge, 2018(Q1). https://doi.org/10.22224/gistbok/2018.1.9

Open Source Initiative. (2006, September 19). Open Standards Requirements for Software - Rationale. Open Source Initiative. Retrieved from https://opensource.org/osr-rationale

Painho, M., & Curvelo, P. (2011). Building dynamic, ontology-based alternative paths for GIS&T curricula. In D. J. Unwin, K. E. Foote, N. J. Tate, & D. DiBiase (Eds.), Teaching Geographic Information Science and Technology in Higher Education (pp. 97–115). West Sussex, UK: Wiley-Blackwell.

Palincsar, A. S. (1986). The Role of Dialogue in Providing Scaffolded Instruction. Educational Psychologist, 21(1 & 2), 73–98.

Peterson, M. P. (2008). International Perspectives on Maps and the Internet: An Introduction. In M. P. Peterson (Ed.), International Perspectives on Maps and the Internet (pp. 3–10). Berlin, Heidelberg: Springer.

Peterson, M. P. (2012). The Tile-Based Mapping Transition in Cartography. In L. Zentai & J. Reyes Nunez (Eds.), Maps for the Future: Children, Education and Internet (pp. 151–163). Berlin, Heidelberg: Springer.

Peterson, M. P. (2014). Mapping in the cloud. New York ; London: The Guilford Press.

Pickles, J. (2006). Ground Truth 1995-2005. Transactions in GIS, 10(5), 763–772. https://doi.org/10.1111/j.1467-9671.2006.01027.x

Prager, S. D. (2011). Using the GIS&T Body of Knowledge for curriculum design: different design for different contexts. In D. J. Unwin, K. E. Foote, N. J. Tate, & D. DiBiase (Eds.), Teaching Geographic Information Science and Technology in Higher Education (pp. 63–80). West Sussex, UK: Wiley-Blackwell.

Prager, S. D., & Plewe, B. (2009). Assessment and Evaluation of GIScience Curriculum using the Geographic Information Science and Technology Body of Knowledge. Journal of Geography in Higher Education, 33(sup1), S46–S69. https://doi.org/10.1080/03098260903034012

Reichenbacher, T. (2003). Adaptive Methods for Mobile Cartography. In Proceedings of the 21st International Cartographic Conference (ICC) (pp. 1311–1322). Durban, South Africa.

Reigeluth, C. M. (2007). Order, First Step to Mastery: An Introduction to Sequencing in Instructional Design. In Ritter, Frank E., J. Nerb, & E. Lehtinen (Eds.), In Order to Learn: How the Sequence of Topics Influences Learning (pp. 19–40). Oxford, England: Oxford University Press.

Ricker, B., & Roth, R. (2018). Mobile Maps and Responsive Design. Geographic Information Science & Technology Body of Knowledge, 2018(Q2). https://doi.org/10.22224/gistbok/2018.2.5

Ricker, B., & Thatcher, J. (2017). Evolving technology, shifting expectations: cultivating pedagogy for a rapidly changing GIS landscape. Journal of Geography in Higher Education, 41(3), 368–382. https://doi.org/10.1080/03098265.2017.1315533

Robinson, A. C., Kerski, J., Long, E. C., Luo, H., DiBiase, D., & Lee, A. (2015). Maps and the geospatial revolution: teaching a massive open online course (MOOC) in geography. Journal of Geography in Higher Education, 39(1), 65–82. https://doi.org/10.1080/03098265.2014.996850

Rød, J. K., Ormeling, F., & van Elzakker, C. (2001). An agenda for democratising cartographic visualisation. Norwegian Journal of Geography, 55, 38–41.

Román-González, M., Pérez-González, J.-C., & Jiménez-Fernández, C. (2017). Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test. Computers in Human Behavior, 72, 678–691. https://doi.org/10.1016/j.chb.2016.08.047

Roth, R. E. (2012). Cartographic Interaction Primitives: Framework and Synthesis. The Cartographic Journal, 49(4), 376–395.

Roth, R. E. (2013). An Empirically-Derived Taxonomy of Interaction Primitives for Interactive Cartography and Geovisualization. IEEE Transactions on Visualization and Computer Graphics, 19(12), 2356–2365. https://doi.org/10.1109/TVCG.2013.130

Roth, R. E. (2016). Rethinking cartography curriculum to train the contemporary cartographer. In Proceedings of the International Conference on Cartography and GIS. Albena, Bulgaria.

Roth, R. E. (2017). User Interface and User Experience (UI/UX) Design. Geographic Information Science & Technology Body of Knowledge, 2017(Q2). https://doi.org/10.22224/gistbok/2017.2.5

Roth, R. E., Çöltekin, A., Delazari, L., Filho, H. F., Griffin, A. L., Hall, A., … van Elzakker, C. P. J. M. (2017). User Studies in Cartography: Opportunities for Empirical Research on Interactive Maps and Visualizations. International Journal of Cartography.

Roth, R. E., Donohue, R. G., Sack, C. M., Wallace, T. R., & Buckingham, T. M. A. (2014). A Process for Keeping Pace with Evolving Web Mapping Technologies. Cartographic Perspectives, (78), 25–52. https://doi.org/10.14714/CP78.1273

Roth, R., Young, S., Nestel, C., Sack, C., Davidson, B., Janicki, J., … Zhang, G. (2018). Global Landscapes: Teaching Globalization through Responsive Mobile Map Design. The Professional Geographer, 1–17. https://doi.org/10.1080/00330124.2017.1416297

Sack, C. (2017). Web Mapping. Geographic Information Science & Technology Body of Knowledge, 2017(Q4). https://doi.org/10.22224/gistbok/2017.4.11

Sack, C. M. (2018). The Status of Web Mapping in North American Higher Education. Cartographic Perspectives, 0(89), 25–43. https://doi.org/10.14714/CP89.1429

Sack, C. M., Donohue, R. G., & Roth, R. E. (2014). Interactive and Multivariate Choropleth Maps with D3. Cartographic Perspectives, 0(78), 57–76. https://doi.org/10.14714/CP78.1278

Sack, C. M., & Roth, R. E. (2017). Design and evaluation of an Open Web Platform cartography lab curriculum. Journal of Geography in Higher Education, 41(1), 1–23. https://doi.org/10.1080/03098265.2016.1241987

Schultz, R. B. (2011). Active pedagogy leading to deeper learning: fostering metacognition and infusing active learning into the GIS&T classroom. In D. J. Unwin, K. E. Foote, N. J. Tate, & D. DiBiase (Eds.), Teaching Geographic Information Science and Technology in Higher Education (pp. 133–143). West Sussex, UK: Wiley-Blackwell.

Shneiderman, B. (1998). Designing the user interface: strategies for effective human-computer-interaction (3rd ed). Reading, Mass: Addison Wesley Longman.

Singh, H. (2003). Building Effective Blended Learning Programs. Educational Technology, 43(6), 51–54.

Smith, D. (2015, October 8). NPAPI Plugin Perspectives and the Oracle JRE. Retrieved April 25, 2016, from https://blogs.oracle.com/java-platform-group/entry/npapi_plugin_perspectives_and_the

Spady, W. G. (1994). Outcome-based education: critical issues and answers. Arlington, Va: American Association of School Administrators.

Stallman, R. (2015). Free Software Is Even More Important Now. GNU Operating System. Retrieved from https://www.gnu.org/philosophy/free-software-even-more-important.html

Steele, R. D. (2012). The open-source everything manifesto: transparency, truth, and trust. Berkeley, Calif: Evolver Editions.

Stevens, S. S. (1957). On the psychophysical law. Psychological Review, 64(3), 153–181. https://doi.org/10.1037/h0046162

Telles, M. A., & Hsieh, Y. (2001). The science of debugging. Scottsdale, Ariz.: Coriolis.

Thatcher, J. E., & Imaoka, L. B. (2018). The poverty of GIS theory: Continuing the debates around the political economy of GISystems: The poverty of GIS theory. The Canadian Geographer / Le Géographe Canadien, 62(1), 27–34. https://doi.org/10.1111/cag.12437

Thompson, A. (2011, July 21). Why are all programming languages in English. Retrieved July 8, 2018, from https://blogs.msdn.microsoft.com/alfredth/2011/07/21/why-are-all-programming-languages-in-english/

Tolochko, R. C. (2016). Contemporary Professional Practices in Interactive Web Map Design (Thesis). University of Wisconsin-Madison, Madison, WI.

Turner, A. (2015, July 20). Vector Tiles preview. Retrieved March 13, 2017, from https://blogs.esri.com/esri/arcgis/2015/07/20/vector-tiles-preview/

UCGIS. (2017). GIS&T Body of Knowledge Project. Retrieved June 8, 2017, from http://gistbok.ucgis.org

Underwood, E. (2013). The New Cartographers. Science. https://doi.org/10.1126/science.caredit.a1300045

Unwin, D. J. (Ed.). (2011). Teaching geographic information science and technology in higher education. Oxford: Wiley-Blackwell.

U.S. News and World Report. (2017). Cartographer: Overview. Retrieved December 19, 2017, from https://money.usnews.com/careers/best-jobs/cartographer

Vos, H., & de Graaff, E. (2004). Developing metacognition: a basis for active learning. European Journal of Engineering Education, 29(4), 543–548. https://doi.org/10.1080/03043790410001716257

Vygotsky, L. S. (1978). Mind in society: the development of higher psychological processes. Cambridge: Harvard University Press.

W3C (World Wide Web Consortium). (2014, October 28). HTML5 is a W3C recommendation [Blog]. Retrieved May 17, 2017, from https://www.w3.org/blog/news/archives/4167

W3C (World Wide Web Consortium). (2015, May 30). Open Web Platform. W3C (World Wide Web Consortium). Retrieved from https://www.w3.org/wiki/Open_Web_Platform

W3C (World Wide Web Consortium). (2016). Accessibility. Retrieved March 17, 2017, from https://www.w3.org/standards/webdesign/accessibility

Weller, M. (2014). The Battle For Open: How openness won and why it doesn't feel like victory. Ubiquity Press. Retrieved from http://www.ubiquitypress.com/site/books/detail/11/battle-for-open/

Wing, J. M. (2011, March 6). Research Notebook: Computational Thinking--What and Why? The Link: The Magazine of Carnegie Mellon University's School of Computer Science. Retrieved from http://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why

Wiseman, A. (2015, June 25). When Maps Lie. Citylab. Retrieved from https://www.citylab.com/design/2016/06/when-maps-lie/396791/

Wood, D. (2003). Cartography is Dead (Thank God!). Cartographic Perspectives, 45, 4–7.

Wood, D., Bruner, J. S., & Ross, G. (1976). The role of tutoring in problem solving. Journal of Child Psychology and Psychiatry, 17, 89–100.

Wood, D., & Fels, J. (1992). The power of maps. New York: Guilford Press.

Wood, M. (2003). Some Personal Reflections on Change ... The Past and Future of Cartography. Cartographic Journal, The, 40(2), 111–115. https://doi.org/10.1179/000870403235001458

Woodruff, A. (2011). Introducing "On the Horizon." Cartographic Perspectives, (68), 83–86. https://doi.org/10.14714/CP68.10

Appendix 1: Interview Study Protocol

Description
This protocol will guide semi-structured interviews with educators who teach web mapping at 10-15 leading post-secondary institutions in North America. The interviews will consist of a mix of answers to leading questions and open-ended discussion, allowing the researcher maximum flexibility to follow interesting threads in the conversation. Transcripts will be coded using qualitative data analysis techniques. A coding scheme will be determined by the researcher; two coders will independently code the interview transcripts, and an inter-rater reliability score will be generated to ensure reliability of the codes. The codes will be reported by frequency, highlighting the themes that are currently of concern to web mapping educators.

Introduction (read aloud)

Thank you for agreeing to participate in this study of web mapping curriculum design in higher education. The study consists of interviews conducted with cartography educators in North America who teach web mapping. The information generated by these interviews will be reported as part of my Ph.D. dissertation and may be published through a peer-reviewed journal, website, blog, or other public forum. This interview will take approximately 60 minutes. I would like to record the interview for later transcription and data analysis, but your name and identifying information will be stripped from the data prior to analysis and publication.

Do I have your permission to record the interview?

[turn on recorder]

In this interview, I will ask you questions regarding your curriculum, instructional practices, and insights related to teaching web mapping. Please be as thorough in your answers as you would like. Feel free to share stories and anecdotes. However, you may also choose not to answer or elaborate on any question. Remember that anything recorded in the transcript will be considered "fair game" for anonymous publication unless you explicitly request otherwise.

During the interview, to protect the confidentiality of people who I will not be interviewing, please do not "name names" of others. Instead, please refer to any other individuals by their relationship to you, such as "my co-worker," "one of my students," "an administrator in my program," etc.

Do you have any questions for me before we begin?

(Start recording)

Questions

I. **Consent**

   1. First, have you read or had read to you the informed consent form for this study, did you understand the form, and do you consent to participate?

II. Background Information

1. **BACKGROUND:** Please take a minute to introduce interests, your education and training, and your current job responsibilities.

I will be using the term "web mapping" to mean the process of making an interactive map that is available through the internet.

2. **TEACHING EXPERIENCE:**
    i. Do you teach the web mapping courses?
    ii. If yes, which courses and how long have you taught them?
    iii. How many students are typically enrolled in the course(s)?
    iv. Are you always the instructor, or does the course rotate instructors?
3. **CURRICULUM**: I have a few basic questions about the courses included in your cartography/GIScience program that include web mapping:
    i. What courses in your program include web mapping in some form?
    ii. When was the web mapping course/were the courses first offered?
    iii. How many students are typically enrolled in each web mapping course?
    iv. What other Cartography/GIS courses do you offer, and where does web mapping typically fall in the course sequence?

III. Topics, Tools, and Technologies

Next, I will ask you questions regarding the content included in your web mapping classes, covering instructional topics, tools, and technologies.

1. **SCOPE:**
    i. What specific topics do you cover in your web mapping course/courses?
    ii. How does your curriculum balance teaching concepts related to web map *design* with concepts related to the technical aspects of web map *development*?
        1. Do emphasize one more than the other?
        2. What concepts are taught for each?
2. **SEQUENCE:**
    i. (If not covered above) What is the order of topics in the curriculum?
    ii. How did you arrive at this topic order?

iii. What topics are particularly tricky or challenging for students to grasp? (**THRESHOLD**)

3. **TOOLS:**
   i. What tools and technologies do you rely on for teaching web mapping? By "tools and technologies," I mean any software applications, programming languages, data formats, code libraries, frameworks, etc. that students receive instruction on how to use as part of learning how to make an interactive web map.
   ii. Why do you use these particular tools and technologies in the web mapping curriculum?
4. **SOFTWARE RIGHTS:** In the following questions, I will use the terms "free and open source" and "proprietary" for different kinds of software. Are you familiar with these terms? (If not) By "free and open source," I mean the software and its source code can legally be used for any purpose, studied, modified, and redistributed. It may or may not cost money to acquire. By "proprietary," I mean software that has a more restrictive license, usually costs money to acquire, and generally cannot easily or legally be modified by the user.
   i. Do you use free and open source software, proprietary software, or a mix of the two to teach web mapping?
   ii. Why do you use this blend / Why do you focus on one over the other?
   iii. Do you see particular advantages or disadvantages to using each kind of software for web mapping? What are these?
5. **TRENDS:**
   i. What changes have happened to the curriculum for the web mapping course/courses you teach over the time they've been offered?
   ii. Do you foresee any particular trends in future changes to web mapping tools and technologies?

IV. Classroom Setting

Next, I will ask you questions about the setting in which you teach. By "setting," I mean the extent to which you deliver course content through in-person interaction in a physical classroom versus through the internet.

1. **SETTING:**
    i. For each web mapping course you teach, is the course primarily taught in an in-person setting, in an online setting, or blended?
    ii. What factored into your decision to offer the web mapping courses in this format?
2. **DISTANCE (if offered online):**
    i. How does the online setting impact the way you teach web mapping?
    ii. How do the students taking online courses differ from resident students?
    iii. What do you see as the benefits and drawbacks of teaching online?
3. **BLENDED (if offered blended):**
    i. How does the blended setting impact the way you teach web mapping?
    ii. What content do you deliver in person, and what content do you deliver online?
    iii. What do you see as the benefits and drawbacks of the blended approach?
4. **MULTIPLE IN SAME COURSE:**
    i. What are the main differences between the different versions of your course/courses?
5. **OPEN EDUCATIONAL RESOURCES:** I will next ask about your use of open educational resources. Are you familiar with this term? (If not) By "open educational resources," I mean course materials that are made publicly available free of charge through the internet.
    i. Do you use any open educational resources to teach web mapping?
        1. If so, what resources?
    ii. Have you created any open educational resources on web mapping?
        1. If so, what content have you made available as open educational resources?
        2. What are the benefits and drawbacks to releasing your teaching materials as open educational resources?

V. Pedagogy

In this section, I will ask you about your teaching pedagogy and successes and failures you have experienced in teaching web mapping. Are you familiar with the term "pedagogy"? (If not) By "pedagogy," I mean the conscious decisions you make about the ways in which you design curriculum and/or teach.

1. **PEDAGOGY**:
    i. Do you utilize particular pedagogical theories or techniques in developing web mapping curriculum for your course/courses?
        1. If so, please describe these theories or techniques, and how you've adapted them for teaching web mapping.
2. **SUCCESSES/CHALLENGES:**
    i. To what extent do you feel you have been successful in teaching students web mapping? Why?
    ii. What specific successes have you experienced with teaching web mapping?
    iii. What specific challenges have you experienced with teaching web mapping?
3. **BEST PRACTICES:**
    i. How do you think your teaching of web mapping could be improved?
    ii. What specific techniques or practices do you see as "best practices" in designing curriculum for and teaching web mapping?
    iii. How do you keep your courses up-to-date as web mapping technology evolves?

VI. Conclusion

1. Is there anything you would like to add about teaching web mapping that I have not asked about?

Thank you very much for your time. If you have any syllabi or other educational resources you are able to share, I'd appreciate it if you could e-mail them to me. If you think of anything more you would like me to know, or any questions arise after our meeting today, please don't hesitate to contact me.

Appendix 2: 2014 Curriculum Evaluation Exit Survey Questions

Bold horizontal lines indicate page breaks. Shaded questions indicate page headers (bolded) or matrix headers (normal font).

Question	Answer Format
PAGE 1/8: BACKGROUND. The following questions provide us with some basic background information about your interests.	
What is your name?	Text
What is your degree program and major? *(for example, BS in Cartography/GIS, MA in Journalism, etc.)*	Text
Please indicate your academic level:	Freshman, Sophomore, Junior, Senior, Certificate Student, Master's student, Ph.D. student, Other (text)
How many previous or concurrent classes (i.e., **not** including Geography 575) have you taken involving the following topics:	
Cartography	Slider 0-20
Visualization (outside of Cartography)	Slider 0-20
GIS	Slider 0-20
Programming/Development	Slider 0-20
Web Design	Slider 0-20
Please rate your interest in the following topics:	
Cartography	(1) No interest, (2), (3), (4) Intermediately interested, (5), (6), (7) Extremely interested
Visualization (outside of Cartography)	(1) No interest, (2), (3), (4) Intermediately interested, (5), (6), (7) Extremely interested
GIS	(1) No interest, (2), (3), (4) Intermediately interested, (5), (6), (7) Extremely interested
Programming/Development	(1) No interest, (2), (3), (4) Intermediately interested, (5), (6), (7) Extremely interested
Web Design	(1) No interest, (2), (3), (4) Intermediately interested, (5), (6), (7) Extremely interested
Why did you take Geography 575? *(optional)*	Text
PAGE 2/8: EXPERTISE. The following questions require you to self-assess your expertise with web specifications, web mapping technologies, and data formats. Please note that you are asked to assess your knowledge both before and after taking Geography 575.	
Please rate your expertise with the following web specifications/libraries/technologies **before taking** Geography 575:	
HTML	(1) I had never used this specification, (2), (3), (4) Intermediate level knowledge, (5), (6), (7) Expert level knowledge

CSS	(1) I had never used this specification, (2), (3), (4) Intermediate level knowledge, (5), (6), (7) Expert level knowledge
JavaScript	(1) I had never used this specification, (2), (3), (4) Intermediate level knowledge, (5), (6), (7) Expert level knowledge
jQuery	(1) I had never used this specification, (2), (3), (4) Intermediate level knowledge, (5), (6), (7) Expert level knowledge
AJAX	(1) I had never used this specification, (2), (3), (4) Intermediate level knowledge, (5), (6), (7) Expert level knowledge
The DOM	(1) I had never used this specification, (2), (3), (4) Intermediate level knowledge, (5), (6), (7) Expert level knowledge
Git/GitHub	(1) I had never used this specification, (2), (3), (4) Intermediate level knowledge, (5), (6), (7) Expert level knowledge
Please rate your expertise with the following web specifications/libraries/technologies **after completing** Geography 575:	
HTML	(1) I still am unfamiliar with this specification, (2), (3), (4) Intermediate level knowledge, (5), (6), (7) Expert level knowledge
CSS	(1) I still am unfamiliar with this specification, (2), (3), (4) Intermediate level knowledge, (5), (6), (7) Expert level knowledge
JavaScript	(1) I still am unfamiliar with this specification, (2), (3), (4) Intermediate level knowledge, (5), (6), (7) Expert level knowledge
jQuery	(1) I still am unfamiliar with this specification, (2), (3), (4) Intermediate level knowledge, (5), (6), (7) Expert level knowledge
AJAX	(1) I still am unfamiliar with this specification, (2), (3), (4) Intermediate level knowledge, (5), (6), (7) Expert level knowledge
The DOM	(1) I still am unfamiliar with this specification, (2), (3), (4) Intermediate

	level knowledge, (5), (6), (7) Expert level knowledge
Git/GitHub	(1) I still am unfamiliar with this specification, (2), (3), (4) Intermediate level knowledge, (5), (6), (7) Expert level knowledge
Please rate your expertise with the following web libraries/technologies **before taking** Geography 575:	
Leaflet	(1) I had never used this technology, (2), (3), (4) Intermediate level knowledge, (5), (6), (7) Expert level knowledge
D3	(1) I had never used this technology, (2), (3), (4) Intermediate level knowledge, (5), (6), (7) Expert level knowledge
Google Maps API	(1) I had never used this technology, (2), (3), (4) Intermediate level knowledge, (5), (6), (7) Expert level knowledge
OpenLayers	(1) I had never used this technology, (2), (3), (4) Intermediate level knowledge, (5), (6), (7) Expert level knowledge
ArcServer/ArcGIS Online	(1) I had never used this technology, (2), (3), (4) Intermediate level knowledge, (5), (6), (7) Expert level knowledge
Mapbox Studio/TileMill	(1) I had never used this technology, (2), (3), (4) Intermediate level knowledge, (5), (6), (7) Expert level knowledge
CartoDB	(1) I had never used this technology, (2), (3), (4) Intermediate level knowledge, (5), (6), (7) Expert level knowledge
Please rate your expertise with the following web libraries/technologies **after completing** Geography 575:	
Leaflet	(1) I still am unfamiliar with this technology, (2), (3), (4) Intermediate level knowledge, (5), (6), (7) Expert level knowledge
D3	(1) I still am unfamiliar with this technology, (2), (3), (4) Intermediate level knowledge, (5), (6), (7) Expert level knowledge
Google Maps API	(1) I still am unfamiliar with this technology, (2), (3), (4) Intermediate

	level knowledge, (5), (6), (7) Expert level knowledge
OpenLayers	(1) I still am unfamiliar with this technology, (2), (3), (4) Intermediate level knowledge, (5), (6), (7) Expert level knowledge
ArcServer/ArcGIS Online	(1) I still am unfamiliar with this technology, (2), (3), (4) Intermediate level knowledge, (5), (6), (7) Expert level knowledge
Mapbox Studio/TileMill	(1) I still am unfamiliar with this technology, (2), (3), (4) Intermediate level knowledge, (5), (6), (7) Expert level knowledge
CartoDB	(1) I still am unfamiliar with this technology, (2), (3), (4) Intermediate level knowledge, (5), (6), (7) Expert level knowledge
Please rate your expertise with the following web data formats **before taking** Geography 575:	
JSON	(1) I had never used this format, (2), (3), (4) Intermediate level knowledge, (5), (6), (7) Expert level knowledge
GeoJSON	(1) I had never used this format, (2), (3), (4) Intermediate level knowledge, (5), (6), (7) Expert level knowledge
TopoJSON	(1) I had never used this format, (2), (3), (4) Intermediate level knowledge, (5), (6), (7) Expert level knowledge
SVG	(1) I had never used this format, (2), (3), (4) Intermediate level knowledge, (5), (6), (7) Expert level knowledge
CSV	(1) I had never used this format, (2), (3), (4) Intermediate level knowledge, (5), (6), (7) Expert level knowledge
KML	(1) I had never used this format, (2), (3), (4) Intermediate level knowledge, (5), (6), (7) Expert level knowledge
Please rate your expertise with the following web data formats **after completing** Geography 575:	
JSON	(1) I am still unfamiliar with this format, (2), (3), (4) Intermediate level knowledge, (5), (6), (7) Expert level knowledge
GeoJSON	(1) I am still unfamiliar with this format, (2), (3), (4) Intermediate level knowledge, (5), (6), (7) Expert level knowledge
TopoJSON	(1) I am still unfamiliar with this format, (2), (3), (4) Intermediate level

	knowledge, (5), (6), (7) Expert level knowledge
SVG	(1) I am still unfamiliar with this format, (2), (3), (4) Intermediate level knowledge, (5), (6), (7) Expert level knowledge
CSV	(1) I am still unfamiliar with this format, (2), (3), (4) Intermediate level knowledge, (5), (6), (7) Expert level knowledge
KML	(1) I am still unfamiliar with this format, (2), (3), (4) Intermediate level knowledge, (5), (6), (7) Expert level knowledge
PAGE 3/8: LEARNING TOPICS. The following questions ask you about the technical topics that were taught in Geography 575.	
Please rate how challenging each of the following technologies and specifications was to learn during Geography 575:	
HTML	(1) Not challenging, (2), (3), (4) Intermediately challenging, (5), (6), (7) Extremely challenging
CSS	(1) Not challenging, (2), (3), (4) Intermediately challenging, (5), (6), (7) Extremely challenging
JavaScript	(1) Not challenging, (2), (3), (4) Intermediately challenging, (5), (6), (7) Extremely challenging
jQuery	(1) Not challenging, (2), (3), (4) Intermediately challenging, (5), (6), (7) Extremely challenging
AJAX	(1) Not challenging, (2), (3), (4) Intermediately challenging, (5), (6), (7) Extremely challenging
The DOM	(1) Not challenging, (2), (3), (4) Intermediately challenging, (5), (6), (7) Extremely challenging
JSON	(1) Not challenging, (2), (3), (4) Intermediately challenging, (5), (6), (7) Extremely challenging
GeoJSON	(1) Not challenging, (2), (3), (4) Intermediately challenging, (5), (6), (7) Extremely challenging
TopoJSON	(1) Not challenging, (2), (3), (4) Intermediately challenging, (5), (6), (7) Extremely challenging
SVG	(1) Not challenging, (2), (3), (4) Intermediately challenging, (5), (6), (7) Extremely challenging

Leaflet	(1) Not challenging, (2), (3), (4) Intermediately challenging, (5), (6), (7) Extremely challenging
D3	(1) Not challenging, (2), (3), (4) Intermediately challenging, (5), (6), (7) Extremely challenging
Git/GitHub	(1) Not challenging, (2), (3), (4) Intermediately challenging, (5), (6), (7) Extremely challenging
Below are the major technical topics presented in class, listed in the sequence in which they were first introduced. Please **reorder them** in any way you think would have better supported your learning. *(optional)*	
Local website directory setup	Order (1)
Basics of HTML	Order (2)
Basics of CSS	Order (3)
Basics of JavaScript	Order (4)
Basics of jQuery	Order (5)
Data levels and types (e.g., nominal/ordinal/interval/ratio, space/time, attribute)	Order (6)
Geographic coordinate systems	Order (7)
Data format specifications (e.g., CSV, GeoJSON, etc.)	Order (8)
Asynchronous JavaScript and XML (AJAX)	Order (9)
Using online reference documentation	Order (10)
Using search engines, online forums, and online examples	Order (11)
Slippy map tile concepts	Order (12)
Basics of Leaflet	Order (13)
Using in-browser developer tools to debug	Order (14)
Leaflet custom UI elements and interactions	Order (15)
Basics of GitHub	Order (16)
SVG elements and attributes	Order (17)
D3 Selections	Order (18)
D3 Generator functions and scales	Order (19)
D3 Geography (projections and path generators)	Order (20)
Please list any of the above topics that needed more repetition or reinforcement throughout the course. Include any suggestions you have for activities utilizing those topics. *(optional)*	Text
Please list any of the above topics that were not very useful for your learning and could be removed. *(optional)*	Text
PAGE 4/8: LEARNING RESOURCES (GENERAL). Please provide us with feedback about your experience with the following learning resources; please consider resources used in class as well as those you found on your own.	
Please rate your **reliance** on the following resources for learning how to make web maps:	
Online Video Tutorials	(1) I didn’t use this resource, (2), (3), (4) Intermediate reliance on this

	resource, (5), (6), (7) I relied heavily on this resource
Online Text Tutorials	(1) I didn't use this resource, (2), (3), (4) Intermediate reliance on this resource, (5), (6), (7) I relied heavily on this resource
Online Documentation	(1) I didn't use this resource, (2), (3), (4) Intermediate reliance on this resource, (5), (6), (7) I relied heavily on this resource
Online Forums	(1) I didn't use this resource, (2), (3), (4) Intermediate reliance on this resource, (5), (6), (7) I relied heavily on this resource
Online Example Code	(1) I didn't use this resource, (2), (3), (4) Intermediate reliance on this resource, (5), (6), (7) I relied heavily on this resource
Lab Assignments	(1) I didn't use this resource, (2), (3), (4) Intermediate reliance on this resource, (5), (6), (7) I relied heavily on this resource
Lecture Notes	(1) I didn't use this resource, (2), (3), (4) Intermediate reliance on this resource, (5), (6), (7) I relied heavily on this resource
In-lab Code Demonstrations	(1) I didn't use this resource, (2), (3), (4) Intermediate reliance on this resource, (5), (6), (7) I relied heavily on this resource
Posted Example Code from Lab	(1) I didn't use this resource, (2), (3), (4) Intermediate reliance on this resource, (5), (6), (7) I relied heavily on this resource
1-on-1 with a TA	(1) I didn't use this resource, (2), (3), (4) Intermediate reliance on this resource, (5), (6), (7) I relied heavily on this resource
Classmates (Peer Assistance)	(1) I didn't use this resource, (2), (3), (4) Intermediate reliance on this resource, (5), (6), (7) I relied heavily on this resource
Please provide an explanation for your above rankings regarding your reliance on the listed learning resources *(optional)*:	Text
Please rate the **effectiveness** of the following resources for learning how to make web maps:	
Online Video Tutorials	(1) Ineffective, (2), (3), (4) Intermediately effective, (5), (6), (7)

	Extremely effective, Not Applicable/I didn't try to use this resource
Online Text Tutorials	(1) Ineffective, (2), (3), (4) Intermediately effective, (5), (6), (7) Extremely effective, Not Applicable/I didn't try to use this resource
Online Documentation	(1) Ineffective, (2), (3), (4) Intermediately effective, (5), (6), (7) Extremely effective, Not Applicable/I didn't try to use this resource
Online Forums	(1) Ineffective, (2), (3), (4) Intermediately effective, (5), (6), (7) Extremely effective, Not Applicable/I didn't try to use this resource
Online Example Code	(1) Ineffective, (2), (3), (4) Intermediately effective, (5), (6), (7) Extremely effective, Not Applicable/I didn't try to use this resource
Lab Assignments	(1) Ineffective, (2), (3), (4) Intermediately effective, (5), (6), (7) Extremely effective, Not Applicable/I didn't try to use this resource
Lecture Notes	(1) Ineffective, (2), (3), (4) Intermediately effective, (5), (6), (7) Extremely effective, Not Applicable/I didn't try to use this resource
In-lab Code Demonstrations	(1) Ineffective, (2), (3), (4) Intermediately effective, (5), (6), (7) Extremely effective, Not Applicable/I didn't try to use this resource
Posted Example Code from Lab	(1) Ineffective, (2), (3), (4) Intermediately effective, (5), (6), (7) Extremely effective, Not Applicable/I didn't try to use this resource
1-on-1 with a TA	(1) Ineffective, (2), (3), (4) Intermediately effective, (5), (6), (7) Extremely effective, Not Applicable/I didn't try to use this resource
Classmates (Peer Assistance)	(1) Ineffective, (2), (3), (4) Intermediately effective, (5), (6), (7) Extremely effective, Not Applicable/I didn't try to use this resource
Please provide an explanation for your above rankings regarding the effectiveness of the listed learning resources *(optional)*:	Text
PAGE 5/8: LEARNING RESOURCES (GEOGRAPHY 575). Please provide us with feedback about your experience with the following learning resources that we used in Geography 575.	
Which JavaScript tutorial(s) did you complete at the beginning of the semester? *(check all that apply)*	

Lynda.com JavaScript Essentials	Checkbox
CodeAcademy JavaScript	Checkbox
DoIT Training Modules	Checkbox
Other (please specify):	Checkbox/Text
Other (list)	Checkbox/Text
Other (list)	Checkbox/Text
I did not fully complete a tutorial module	Checkbox
Please rate your agreement with the following statements about the learning resources (other than the two lab assignments) we used in Geography 575.	
The JavaScript tutorials I completed at the beginning of the semester were useful to me.	(1) Strongly Disagree, (2) Disagree, (3) Somewhat Disagree, (4) Neither Agree nor Disagree, (5) Somewhat Agree, (6) Agree, (7) Strongly Agree
Looking at different types of online documentation in lab was useful to me.	(1) Strongly Disagree, (2) Disagree, (3) Somewhat Disagree, (4) Neither Agree nor Disagree, (5) Somewhat Agree, (6) Agree, (7) Strongly Agree
The assignment to add comments to sample code was useful to me.	(1) Strongly Disagree, (2) Disagree, (3) Somewhat Disagree, (4) Neither Agree nor Disagree, (5) Somewhat Agree, (6) Agree, (7) Strongly Agree
The in-lab, instructor-led coding demonstrations were useful to me.	(1) Strongly Disagree, (2) Disagree, (3) Somewhat Disagree, (4) Neither Agree nor Disagree, (5) Somewhat Agree, (6) Agree, (7) Strongly Agree
Handouts provided in lab and through Learn@UW were useful to me.	(1) Strongly Disagree, (2) Disagree, (3) Somewhat Disagree, (4) Neither Agree nor Disagree, (5) Somewhat Agree, (6) Agree, (7) Strongly Agree
Sample code provided in lab and through Learn@UW was useful to me.	(1) Strongly Disagree, (2) Disagree, (3) Somewhat Disagree, (4) Neither Agree nor Disagree, (5) Somewhat Agree, (6) Agree, (7) Strongly Agree
Please provide an explanation for your above rankings about the learning resources we provided in lab; what worked, and how can we improve them in the future? *(optional)*	Text
Please rate your agreement with the following statements about the two lab assignments and the final project:	
I found it easy to relate the concepts presented in lecture to Labs #1 and #2.	(1) Strongly Disagree, (2) Disagree, (3) Somewhat Disagree, (4) Neither Agree nor Disagree, (5) Somewhat Agree, (6) Agree, (7) Strongly Agree
I found it easy to relate the concepts presented in Lab #1 to Lab #2.	(1) Strongly Disagree, (2) Disagree, (3) Somewhat Disagree, (4) Neither Agree nor Disagree, (5) Somewhat Agree, (6) Agree, (7) Strongly Agree
I found it easy to relate the concepts presented in Labs #1 and #2 to the final project.	(1) Strongly Disagree, (2) Disagree, (3) Somewhat Disagree, (4) Neither

	Agree nor Disagree, (5) Somewhat Agree, (6) Agree, (7) Strongly Agree
The final project proposal helped relate the concepts presented in lecture to the final project.	(1) Strongly Disagree, (2) Disagree, (3) Somewhat Disagree, (4) Neither Agree nor Disagree, (5) Somewhat Agree, (6) Agree, (7) Strongly Agree
Having parts of Labs #1 and #2 due each week helped keep me on track.	(1) Strongly Disagree, (2) Disagree, (3) Somewhat Disagree, (4) Neither Agree nor Disagree, (5) Somewhat Agree, (6) Agree, (7) Strongly Agree
After completing the labs and final projects, I know where and how to seek help independently on future projects.	(1) Strongly Disagree, (2) Disagree, (3) Somewhat Disagree, (4) Neither Agree nor Disagree, (5) Somewhat Agree, (6) Agree, (7) Strongly Agree
Please provide an explanation for your above rankings about the two lab assignments and the final project assignment; what worked, and how can we improve them in the future? *(optional)*	Text
PAGE 6/8: LEAFLET LAB. Please answer the following questions specifically considering your learning experience with Lab #1.	
Please rate your agreement with the following statements about Lab #1 (Leaflet).	
The Lab #1 instructions were easy to follow.	(1) Strongly Disagree, (2) Disagree, (3) Somewhat Disagree, (4) Neither Agree nor Disagree, (5) Somewhat Agree, (6) Agree, (7) Strongly Agree
I had a difficult time understanding the instructions in Lab #1.	(1) Strongly Disagree, (2) Disagree, (3) Somewhat Disagree, (4) Neither Agree nor Disagree, (5) Somewhat Agree, (6) Agree, (7) Strongly Agree
The Lab #1 instructions were comprehensive, including everything I needed to know.	(1) Strongly Disagree, (2) Disagree, (3) Somewhat Disagree, (4) Neither Agree nor Disagree, (5) Somewhat Agree, (6) Agree, (7) Strongly Agree
I sought out instructor help frequently to overcome obstacles to completing Lab #1.	(1) Strongly Disagree, (2) Disagree, (3) Somewhat Disagree, (4) Neither Agree nor Disagree, (5) Somewhat Agree, (6) Agree, (7) Strongly Agree
After completing Lab #1, I knew how to make an effective and well-designed web map using Leaflet.	(1) Strongly Disagree, (2) Disagree, (3) Somewhat Disagree, (4) Neither Agree nor Disagree, (5) Somewhat Agree, (6) Agree, (7) Strongly Agree
I feel like I would have a difficult time applying concepts and techniques in Lab #1 to make a new map in Leaflet.	(1) Strongly Disagree, (2) Disagree, (3) Somewhat Disagree, (4) Neither Agree nor Disagree, (5) Somewhat Agree, (6) Agree, (7) Strongly Agree
Please rate how challenging each of the following technical topics included in Lab #1 was to learn:	
Finding and formatting data	(1) Not challenging, (2), (3), (4) Intermediately challenging, (5), (6), (7) Extremely challenging

Importing data with AJAX	(1) Not challenging, (2), (3), (4) Intermediately challenging, (5), (6), (7) Extremely challenging
Using JavaScript and/or jQuery	(1) Not challenging, (2), (3), (4) Intermediately challenging, (5), (6), (7) Extremely challenging
Applying Leaflet methods to map the data	(1) Not challenging, (2), (3), (4) Intermediately challenging, (5), (6), (7) Extremely challenging
Styling the map	(1) Not challenging, (2), (3), (4) Intermediately challenging, (5), (6), (7) Extremely challenging
Implementing the symbol legend	(1) Not challenging, (2), (3), (4) Intermediately challenging, (5), (6), (7) Extremely challenging
Implementing the temporal slider	(1) Not challenging, (2), (3), (4) Intermediately challenging, (5), (6), (7) Extremely challenging
Implementing the temporal legend	(1) Not challenging, (2), (3), (4) Intermediately challenging, (5), (6), (7) Extremely challenging
Styling the legends or other interface elements	(1) Not challenging, (2), (3), (4) Intermediately challenging, (5), (6), (7) Extremely challenging
Adding a fifth interaction operator	(1) Not challenging, (2), (3), (4) Intermediately challenging, (5), (6), (7) Extremely challenging
General debugging	(1) Not challenging, (2), (3), (4) Intermediately challenging, (5), (6), (7) Extremely challenging
Other (please specify): (Text)	(1) Not challenging, (2), (3), (4) Intermediately challenging, (5), (6), (7) Extremely challenging
Other (please specify): (Text)	(1) Not challenging, (2), (3), (4) Intermediately challenging, (5), (6), (7) Extremely challenging
Please rate your **reliance** on the following learning resources for completing Lab #1:	
Lab Assignment Handout	(1) I didn’t use this resource, (2), (3), (4) Intermediate reliance on this resource, (5), (6), (7) I relied heavily on this resource
Posted Lab Example Code	(1) I didn’t use this resource, (2), (3), (4) Intermediate reliance on this resource, (5), (6), (7) I relied heavily on this resource
In-Lab Code Demonstrations	(1) I didn’t use this resource, (2), (3), (4) Intermediate reliance on this resource, (5), (6), (7) I relied heavily on this resource

Online Video Tutorials (not developed by instructors)	(1) I didn't use this resource, (2), (3), (4) Intermediate reliance on this resource, (5), (6), (7) I relied heavily on this resource
Online Text Tutorials (not developed by instructors)	(1) I didn't use this resource, (2), (3), (4) Intermediate reliance on this resource, (5), (6), (7) I relied heavily on this resource
Online Documentation (not developed by instructors)	(1) I didn't use this resource, (2), (3), (4) Intermediate reliance on this resource, (5), (6), (7) I relied heavily on this resource
Online Forums (not supported by instructors)	(1) I didn't use this resource, (2), (3), (4) Intermediate reliance on this resource, (5), (6), (7) I relied heavily on this resource
Online Example Code (not developed by instructors)	(1) I didn't use this resource, (2), (3), (4) Intermediate reliance on this resource, (5), (6), (7) I relied heavily on this resource
1-on-1 with your TA	(1) I didn't use this resource, (2), (3), (4) Intermediate reliance on this resource, (5), (6), (7) I relied heavily on this resource
Classmates (Peer Assistance)	(1) I didn't use this resource, (2), (3), (4) Intermediate reliance on this resource, (5), (6), (7) I relied heavily on this resource
What was your overall emotional experience with Lab #1? *(optional)*	
Overall Emotional Experience	(1) Extremely negative, (2), (3), (4) Neutral, (5), (6), (7) Extremely positive
What did you like about Lab #1? What worked well? *(optional)*	Text
What did you dislike about Lab #1? What can we improve in the future? *(optional)*	Text
PAGE 7/8: D3 LAB. Please answer the following questions specifically considering your learning experience with <u>Lab #2</u>.	
Please rate your agreement with the following statements about <u>Lab #2 (D3)</u>.	
The Lab #2 instructions were easy to follow.	(1) Strongly Disagree, (2) Disagree, (3) Somewhat Disagree, (4) Neither Agree nor Disagree, (5) Somewhat Agree, (6) Agree, (7) Strongly Agree
I had a difficult time understanding the instructions in Lab #2.	(1) Strongly Disagree, (2) Disagree, (3) Somewhat Disagree, (4) Neither Agree nor Disagree, (5) Somewhat Agree, (6) Agree, (7) Strongly Agree

The Lab #2 instructions were comprehensive, including everything I needed to know.	(1) Strongly Disagree, (2) Disagree, (3) Somewhat Disagree, (4) Neither Agree nor Disagree, (5) Somewhat Agree, (6) Agree, (7) Strongly Agree
I sought out instructor help frequently to overcome obstacles to completing Lab #2.	(1) Strongly Disagree, (2) Disagree, (3) Somewhat Disagree, (4) Neither Agree nor Disagree, (5) Somewhat Agree, (6) Agree, (7) Strongly Agree
After completing Lab #2, I knew how to make an effective and well-designed web map using D3.	(1) Strongly Disagree, (2) Disagree, (3) Somewhat Disagree, (4) Neither Agree nor Disagree, (5) Somewhat Agree, (6) Agree, (7) Strongly Agree
I feel like I would have a difficult time applying concepts and techniques in Lab #2 to make a new map in D3.	(1) Strongly Disagree, (2) Disagree, (3) Somewhat Disagree, (4) Neither Agree nor Disagree, (5) Somewhat Agree, (6) Agree, (7) Strongly Agree
Please rate how challenging each of the following technical topics included in Lab #2 was to learn:	
Finding and formatting data	(1) Not challenging, (2), (3), (4) Intermediately challenging, (5), (6), (7) Extremely challenging
Importing data with AJAX	(1) Not challenging, (2), (3), (4) Intermediately challenging, (5), (6), (7) Extremely challenging
Drawing the base map with D3	(1) Not challenging, (2), (3), (4) Intermediately challenging, (5), (6), (7) Extremely challenging
Styling the base map with D3	(1) Not challenging, (2), (3), (4) Intermediately challenging, (5), (6), (7) Extremely challenging
Styling the map	(1) Not challenging, (2), (3), (4) Intermediately challenging, (5), (6), (7) Extremely challenging
Implementing the choropleth color scheme with D3	(1) Not challenging, (2), (3), (4) Intermediately challenging, (5), (6), (7) Extremely challenging
Connecting multiple data sources (TopoJSON and CSV) using JavaScript loops	(1) Not challenging, (2), (3), (4) Intermediately challenging, (5), (6), (7) Extremely challenging
Implementing the supporting visualization (e.g., bar graph)	(1) Not challenging, (2), (3), (4) Intermediately challenging, (5), (6), (7) Extremely challenging
Implementing dynamic attribute selection (e.g., dropdown menu)	(1) Not challenging, (2), (3), (4) Intermediately challenging, (5), (6), (7) Extremely challenging
Implementing coordinated interactions (highlighting and dynamic label retrieve) on the map and graph	(1) Not challenging, (2), (3), (4) Intermediately challenging, (5), (6), (7) Extremely challenging

Adding additional interactions and interface elements	(1) Not challenging, (2), (3), (4) Intermediately challenging, (5), (6), (7) Extremely challenging
General debugging	(1) Not challenging, (2), (3), (4) Intermediately challenging, (5), (6), (7) Extremely challenging
Other (please specify): (Text)	(1) Not challenging, (2), (3), (4) Intermediately challenging, (5), (6), (7) Extremely challenging
Other (please specify): (Text)	(1) Not challenging, (2), (3), (4) Intermediately challenging, (5), (6), (7) Extremely challenging
Please rate your **reliance** on the following learning resources for completing Lab #2:	
Lab Assignment Handout	(1) I didn't use this resource, (2), (3), (4) Intermediate reliance on this resource, (5), (6), (7) I relied heavily on this resource
Posted Lab Example Code	(1) I didn't use this resource, (2), (3), (4) Intermediate reliance on this resource, (5), (6), (7) I relied heavily on this resource
In-Lab Code Demonstrations	(1) I didn't use this resource, (2), (3), (4) Intermediate reliance on this resource, (5), (6), (7) I relied heavily on this resource
Online Video Tutorials (not developed by instructors)	(1) I didn't use this resource, (2), (3), (4) Intermediate reliance on this resource, (5), (6), (7) I relied heavily on this resource
Online Text Tutorials (not developed by instructors)	(1) I didn't use this resource, (2), (3), (4) Intermediate reliance on this resource, (5), (6), (7) I relied heavily on this resource
Online Documentation (not developed by instructors)	(1) I didn't use this resource, (2), (3), (4) Intermediate reliance on this resource, (5), (6), (7) I relied heavily on this resource
Online Forums (not supported by instructors)	(1) I didn't use this resource, (2), (3), (4) Intermediate reliance on this resource, (5), (6), (7) I relied heavily on this resource
Online Example Code (not developed by instructors)	(1) I didn't use this resource, (2), (3), (4) Intermediate reliance on this resource, (5), (6), (7) I relied heavily on this resource
1-on-1 with your TA	(1) I didn't use this resource, (2), (3), (4) Intermediate reliance on this resource, (5), (6), (7) I relied heavily on this resource

Classmates (Peer Assistance)	(1) I didn't use this resource, (2), (3), (4) Intermediate reliance on this resource, (5), (6), (7) I relied heavily on this resource
What was your overall emotional experience with Lab #2? *(optional)*	
Overall Emotional Experience	(1) Extremely negative, (2), (3), (4) Neutral, (5), (6), (7) Extremely positive
What did you like about Lab #2? What worked well? *(optional)*	Text
What did you dislike about Lab #2? What can we improve in the future? *(optional)*	Text
PAGE 8/8: FINAL PROJECT. You're almost done! Please answer the following questions specifically considering how the course built into the final project, and your experience therein.	
Please rate your **reliance** on the following learning resources for completing the final project:	
Online Video Tutorials	(1) I didn't use this resource, (2), (3), (4) Intermediate reliance on this resource, (5), (6), (7) I relied heavily on this resource
Online Text Tutorials	(1) I didn't use this resource, (2), (3), (4) Intermediate reliance on this resource, (5), (6), (7) I relied heavily on this resource
Online Documentation	(1) I didn't use this resource, (2), (3), (4) Intermediate reliance on this resource, (5), (6), (7) I relied heavily on this resource
Online Forums	(1) I didn't use this resource, (2), (3), (4) Intermediate reliance on this resource, (5), (6), (7) I relied heavily on this resource
Online Example Code	(1) I didn't use this resource, (2), (3), (4) Intermediate reliance on this resource, (5), (6), (7) I relied heavily on this resource
Lab Assignments	(1) I didn't use this resource, (2), (3), (4) Intermediate reliance on this resource, (5), (6), (7) I relied heavily on this resource
Lecture Notes	(1) I didn't use this resource, (2), (3), (4) Intermediate reliance on this resource, (5), (6), (7) I relied heavily on this resource
In-lab Code Demonstrations	(1) I didn't use this resource, (2), (3), (4) Intermediate reliance on this resource, (5), (6), (7) I relied heavily on this resource

Posted Example Code from Lab	(1) I didn't use this resource, (2), (3), (4) Intermediate reliance on this resource, (5), (6), (7) I relied heavily on this resource
1-on-1 with a TA	(1) I didn't use this resource, (2), (3), (4) Intermediate reliance on this resource, (5), (6), (7) I relied heavily on this resource
Classmates (Peer Assistance)	(1) I didn't use this resource, (2), (3), (4) Intermediate reliance on this resource, (5), (6), (7) I relied heavily on this resource
What was your overall emotional experience with the final project? *(optional)*	
Overall Emotional Experience	(1) Extremely negative, (2), (3), (4) Neutral, (5), (6), (7) Extremely positive
What major breakthroughs did you experience while completing the final project? What new strategies, techniques, or solutions did you discover? *(optional)*	Text
What major frustrations did you experience while completing the final project? What are the remaining key gaps in your knowledge about web mapping and what would you still like to learn? *(optional)*	Text
How would you improve the overall Geography 575 learning experience for future students? *(optional)*	Text

Appendix 3: Interactive Cartography and Geovisualization Lab Curriculum Module 1

Note: the material below is copyrighted by the University of Wisconsin Board of Regents.

Lab Module 1: Setting Up Your Workspace

Introduction

Welcome to the first lab module of Geography 575! In this module, we will start with a look at some boilerplate HTML and a few popular frameworks you might use to structure your website. This information should review and build on the knowledge of website design you gained in Geography 572. Second, we will take a look at how to neatly and efficiently set up your workspace and host it through a localhost development server. Finally, we will set up a GitHub account you can use for cloud storage, version control, web hosting, and collaboration with other developers and add a repository for your website.

When you have finished this module, you should be able to:

- **Select a framework or boilerplate for use as the base of your website**
- **Build a website directory hosted on a localhost server**
- **Create a GitHub repository for your website and sync it with your local directory**

A bit of housekeeping first...

The lab modules for this course will all be formatted the same way. Each module page is broken down into 3-4 lessons, which in turn have sub-sections. As you read, you will often encounter links which point to additional reference material and sometimes software downloads you will need.

> Occasionally, there are directions for an activity to complete. They are formatted in the text like this. You should follow these directions, as they will contribute to the module deliverable that will be due in lab. Deliverables are listed at the end of each module with a link to the assignment page and due at the start of lab the following week (e.g., Module 1 deliverables are due at the start of the Week 2 lab period).

```
Examples are formatted in code blocks, like this.
You can copy-paste this code into a text editor to make it easier to reference the line numbers.
```

You will also run into `in-line code` where it is necessary to literally reference some element tag, function or variable name, value, etc. The literal names of *directories* and *files* are italicized.

At the end of each lesson is a Self-Check, a two-question mini-quiz you can take to make sure you understood the material. The questions are included in the module text, and the self-check heading is a link to a practice quiz with the same questions and their correct answers. These quizzes are unscored; they are solely a learning tool that we recommend you make use of.

If you have questions, please post them in Discussion forms or e-mail Carl at cmsack@wisc.edu.

Lesson 1: Boilerplates and Frameworks

I. Text Editors

By this point, you should already be familiar with one or more open-source text editors. Different editors include various features, such as color-coding specific to different code languages, automatic indentation and closing tags, and live preview. Since these are updated with new features regularly, now is a good time to review your choice of editor and compare it to other available editors to see if it may be worth switching. Some popular editors as of this writing are:

Notepad++ : a simple, lightweight text editor with a number of available plugins and a large user community. Windows only.

Sublime Text : a user-friendly text editor with a number of useful features such as programmable keyboard shortcuts, a robust find/replace tool, and autocomplete. Available for Windows, Mac OS X, and Linux.

Aptana Studio : a full-featured open-source IDE (Integrated Development Environment) with a design based on Eclipse, but specialized for web languages. Available for Windows, Mac OS X, and Linux.

Brackets : Adobe's open-source text editor includes a web directory tree, live preview, and slick design.

Atom : An open-source text editor by GitHub, similar to Sublime or Brackets. It integrates with your working Git repository to color-code files that have been added or changed in the file tree. Available for Windows, Mac OS X, and Linux.

II. The Boilerplate

What is a boilerplate? To quote from Wikipedia , "In computer programming, boilerplate code or boilerplate is the sections of code that have to be included in many places with little or no alteration." In other words, it's the minimal starter code that you build around.

It is assumed here that you are already at least somewhat familiar with HTML, so we won't get into how it works or what the various tags are. If you are new to HTML, you should complete a full HTML tutorial through W3Schools , CodeAcademy , or Lynda.com . If you just need a reference, we recommend W3Schools or the Mozilla Developer Network .

> Download my_website.zip from the Module 1 files, unzip it, and open ***index.html*** in your text editor.

The code should look like this:

Example 1.1: boilerplate code in *Index.html*

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width">
            <title></title>

        <!--put your external stylesheet links here-->
        <link rel="stylesheet" href="css/style.css">
        <!--[if IE<9]>
            <link rel="stylesheet" href="css/style.ie.css">
        <![endif]-->
    </head>

    <body>
        <!--put your initial page content here-->

        <!--you can also use this space for internal scripts or stylesheets;
        place these within <script> or <style> tags-->

        <!--put your external script links here-->
```

```
        <script type="text/javascript" src="js/main.js"></script>
    </body>
</html>
```

The different sections of this very minimal html file should look familiar to you. All of its components, except perhaps the `<!-- -->` comments and IE stylesheet, should *always* be present in your *index.html* file (keep in mind that *index.html* must be the name of the homepage to use the directory name as the endpoint of a web url). Note the neatly indented structure of the markup. This is not strictly required for the code to work; browsers interpret opening and closing tags, not newline or tab characters, as the start and end points of elements. However, proper indentation is a convention and a best practice that will make your life and the life of your TA (and anyone else reading your code) exponentially easier. Different levels of indentation represent the levels of the elements in the overall tree structure of the document. Think of this structure as analagous to Matryoshka dolls that nest one inside the other. In the code above, the `<html>` tag is the outermost "doll", the `<head>` and `<body>` tags sit inside of that, and within each of those "dolls" are all of the elements that are indented three tabs below their opening and above their closing tags.

The `<!DOCTYPE html>` tag indicates to the browser that this is an HTML5 file. The `<html lang="en">` tag opens the html code, telling the browser to interpret the code as English. In the `<head>` tag, there are two `<meta>` tags: the first declares the character set as UTF-8, which interprets the broadest possible range of ASCII characters; the second `<meta>` tag sets the width of the page to the width of the device monitor, ensuring that the content stays within the available frame regardless of device size. While there are other options available for the `<meta name="viewport">` tag, this is really the only setting that should ever be used, and it should *always* be included. The `<title>` open and closing tags declare the title of the page; text in between these tags will appear in the tab at the top of the browser page. The `<link>` tag links out to the website stylesheet, which will be covered further soon. Finally, the conditional statement `<!--[if IE<9]>`, although it looks like a comment, loads a different stylesheet formatted for old versions of Internet Explorer prior to version 9.

The `<body>` will contain the majority of the page content you write in HTML. Below your custom HTML is where you should place your `<script>` tags linking to the site's javascript. This will also be covered more later.

This is all you need to start on a basic HTML5 webpage. If you add something to the body, say:

Example 1.2

```
<!--Example 1 line 14-->
<body>
    Hello World!
</body>
```

and then open the file in a browser, you should see what you just added to the page (figure 1.1):

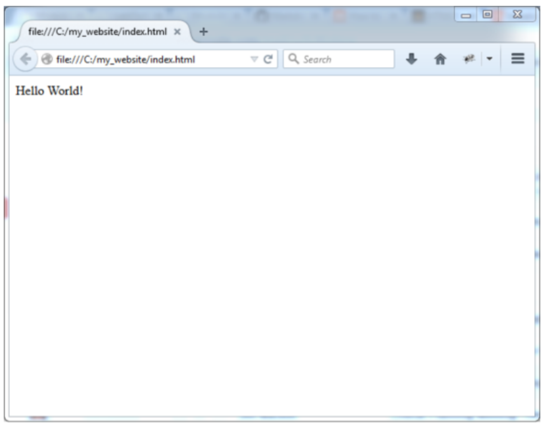


Figure 1.1: `<body>` content displayed on the page

> Add a title to the title tag, then load your ***index.html*** file in a browser and check that your title appears in the browser tab.

For simplicity's sake, this is the boilerplate we will be using as a starting point for each module activity and lab assignment throughout the course. But what if you want a shortcut to making a really spiffy website with the latest whizbang features without worrying about how to style everything yourself? That's where using a framework might be helpful.

III. Frameworks

A framework is a pre-packaged website directory containing a structure of files and folders with standardized code that you can customize on top of. Frameworks give a clean, familiar look and feel to your site. You probably visit sites that use frameworks regularly, and might recognize a site's look as being similar to that of others you've visited if you really pay attention.

There are all kinds of frameworks for both front-end (client-side code) and back-end (server-side code), but we will just touch on front-end frameworks here. Common front-end framework components include:

- Responsive web design: a series of strategies implemented through CSS and JavaScript that dynamically change the layout of a website depending on a user's screen size, especially important for websites to be usable on mobile devices (figure 1.2).

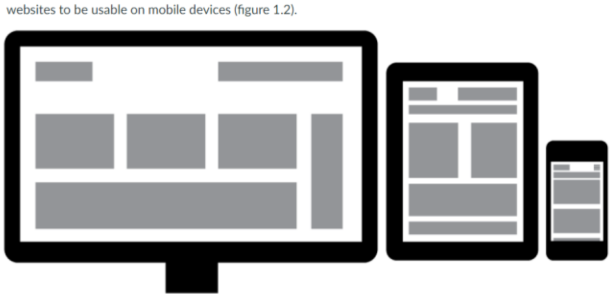


Figure 1.2: responsive web design demonstration

- *Normalization:* CSS styles to make a website appear the same regardless of the browser used, since different browsers interpret default page styles slightly differently (figure 1.3).

Figure 1.3: normalization demonstration

- *Custom syntax:* specific ways in which you must write your own code to make use of the framework's features, described in the documentation on the framework's website. The example below (example 1.3) is from the Bootstrap framework.

  Example 1.3: Bootstrap HTML syntax example

```
<div class="row">
    <div class="col-md-1">.col-md-1</div>
    <div class="col-md-1">.col-md-1</div>
    <div class="col-md-1">.col-md-1</div>
    <div class="col-md-1">.col-md-1</div>
    <div class="col-md-1">.col-md-1</div>
    <div class="col-md-1">.col-md-1</div>
    <div class="col-md-1">.col-md-1</div>
    <div class="col-md-1">.col-md-1</div>
    <div class="col-md-1">.col-md-1</div>
    <div class="col-md-1">.col-md-1</div>
    <div class="col-md-1">.col-md-1</div>
    <div class="col-md-1">.col-md-1</div>
</div>
```

- *Grids*: CSS styles included in many, but not all, frameworks to create a column-based page layout that makes the site appear clean and professional (figure 1.4).

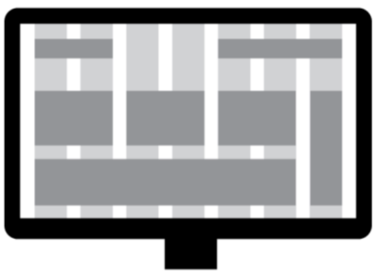


Figure 1.4: grids demonstration

Here are three popular examples you might be interested in using if you already have some experience with basic HTML and CSS (if not, you'll want to stick to the basic boilerplate for now).

HTML5 Boilerplate - The developers of this framework see it as a boilerplate because its components are so commonly used they might be considered basic to the modern website. These components include *normalize.css* and *modernizr.js*, which standardize cross-browser rendering and script-interpreting differences; jQuery, which we will cover in a later module and use subsequently throughout the course; and Google Analytics, which tracks website usage statistics. It adds a few of its own styles for responsive web design and server configuration settings to boost performance.

Bootstrap - You should already be familiar with this framework from Geography 572. Pioneered by Twitter, it is now offered freely on GitHub and is used by a number of big-name websites. It includes normalized JavaScript and CSS as well as a responsive grid system allowing for consistent spacing of elements on the page and easy transition between mobile and desktop screen sizes. If you use it, make sure you are following the classname conventions specified in the documentation on the Bootstrap website in any HTML markup you build.

Foundation - Foundation is similar to Bootstrap in its use of normalization and grids. It includes a number of website templates and signature user interface elements, and adds a number of optional JavaScript-based plugins, such as easy-to-implement popup windows. There is a bit of a learning curve to Foundation, so you may want to make use of their training courses if you'd like to build with it.

There are many other frameworks available that follow similar principles. If you'd like to see more, simply do a web search for "website frameworks."

> ### Lesson 1 Self-Check:
>
> 1. Where should you put `<link>` and `<script>` tags?
>   a. Both in the `<head>` section.
>   b. `<link>` in the `<head>` section and `<script>` in the `<body>` section.
>   c. `<script>` in the `<head>` section and `<link>` in the `<body>` section.
>   d. Both in the `<body>` section.
> 2. True/False: Normalization is a series of strategies implemented through CSS and JavaScript that dynamically change the layout of a website depending on a user's screen size.

Lesson 2: Web Directory Setup

I. One-file websites

The case of a one-file website is trivial; in real life web development, you will always want to link out to resources to use on your site through `<link>` tags and `<script>` tag `src` attributes. But it is useful to know that you *can* include styles and JavaScript right in the HTML document. For example, you could add to your boilerplate:

Example 2.1: adding styles in *index.html*

```
<body>
    <style>
        #mydiv {
            background-color: red;
            width: 80%;
            margin: 0 auto;
            height: 100px;
        }
    </style>
```

```
    <div id="mydiv">
        Hello World!
    </div>
</body>
```

You would get a red rectangle 100 pixels high that takes up 80 percent of the page width, with no margin, as seen below (figure 2.1).

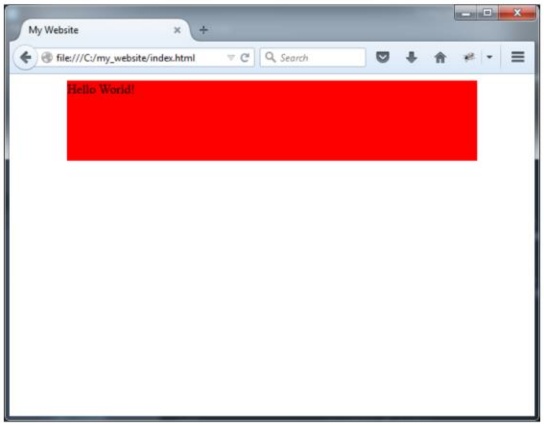


Figure 2.1: result of example 2.1

You can also add a little JavaScript using plain `<script>` tags without a `src` attribute below the `<div>`. In the example below (example 2.2), we add an event listener to the `<div>` using this method.

Example 2.2: adding script in *index.html*

```
<div id="mydiv">
    Hello World!
</div>

<script>
    var mydiv = document.getElementById("mydiv");
    mydiv.addEventListener("click", function(){
        alert("Hello World!");
    });
</script>
```

If you click on the div in your browser, you should see:

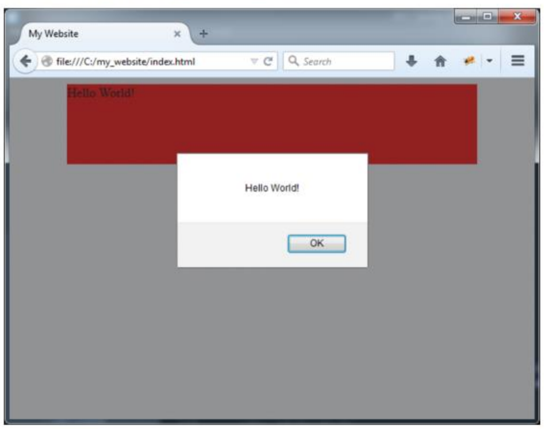


Figure 2.2: result of example 2.2

To reiterate, for the real-world projects you'll be working on, internal `<style>` and `<script>` tags are much too limiting. With hundreds of lines of JavaScript and CSS code, it's important to keep things neat and orderly, which requires separating the working pieces of your site into a well-organized structure of files and folders. You may even find that you want to create *multiple* JavaScript files and *multiple* stylesheets, with each applying to a different section or function of the site.

II. Directory Structure

You will thank yourself down the road if you start out with a good skeleton structure to hold all your files. Like shelves in a closet, this structure represents a strategy to keep things organized so you can find them later.

> In the my_website directory you downloaded, create the following new folders:
> - js
> - lib
> - css
> - data
> - img

Let's go through each of these in turn:

- **js**: This folder will hold all of your custom JavaScript files, in which you will write the script to make your site—including its web map—dynamic and interactive.
- **lib**: This folder holds any third-party JavaScript code libraries you choose to download and host. These will eventually include jQuery, Leaflet, D3, and others. Some code libraries come with their own css stylesheets and/or images; leave these within the library's directory and place the entire directory within the lib folder.
- **css**: This folder will hold all of your custom CSS stylesheets.
- **data**: You won't always need to keep something in this folder, but it's here because you'll need it when you add data to create your thematic web maps. This folder may hold CSV, GeoJSON, and other formats that we'll cover later. You might also wish to keep original ESRI Shapefiles in it; these should go in a *shapefiles* subdirectory within the data folder.

- **img**: This is where you keep the images you will be using in your site.

Note that in the boilerplate *index.html* file, there are already two links to files that don't exist yet! It's time to create them. In your text editor, create a new file and save it in the *css* folder as *style.css*. Optionally, you may wish to put a comment at the top declaring your authorship (example 2.3), since you will eventually be sharing this with the wide world through GitHub:

Example 2.3: authorship comment in *style.css*

```
/* Stylesheet by Buck E. Badger, 2015 */
```

Now do the same thing with *main.js* in the *js* folder. Again, if you desire, you can add a byline comment at the top. Authors of code libraries will often also add a descriptive note addressing what the code is designed to do and any license information. You don't have to worry about that just yet, though.

Your web directory is all set up. But as you develop your website, you will need to preview it in a browser to see what it looks like and to use the browser's helpful set of developer tools. For this to really work, you need to set up a development server.

III. Setting up a Development Server

A *server* is a piece of software that sends data to a web user's browser. A server may be across the room, across the country, or even on the other side of the world. But you can also put a server on your own machine to handle requests your browser makes for files and data on your machine. The figure below (figure 2.3) shows a basic server-client architecture, in which the client requests files from your website directory and the server sends them as requested.

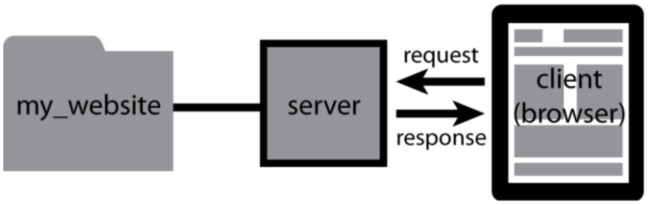


Figure 2.3: client-server architecture

While you technically *can* just double-click on an *index.html* file to open it in a browser, for security reasons some browsers (such as Chrome as of this writing) will *only* display websites correctly if the files are passed through a server. Thus, you should set one up on your machine. You have many, many options; here are a few:

- **Preprocessors** such as Prepros and CodeKit come with a server embedded. Their live preview automatically compiles your website and refreshes the browser every time you save something. This is likely to be the solution for a local development server with the easiest setup.
- **Python SimpleHTTPServer** should already be on your machine if you have Python installed (which you may have done for your first Geocomputing course). From the command line or shell, first navigate to the directory in which your website directory is located; for instance:

  Example 2.4: navigating to your web directory in the command shell

  ```
  c:\>cd c:\webdir
  c:\webdir>
  ```

  It is a good idea to create a directory in which you will place *all* of your websites so you can serve all of them at once (*webdir* in the example above). Once you are pointed toward the right directory, enter:

  Example 2.5: starting a python server in the command shell

  ```
  python -m SimpleHTTPServer 8080 &.
  ```

  This will fire up a server on TCP Port 8080 (for an explanation of these port numbers, see the Wikipedia page). If your *my_website* directory is within the directory being served, you can load your website into the browser with the url:

  Example 2.6: accessing your website in the browser URL bar

  ```
  http://localhost:8080/my_website/
  ```

  The downside to this approach is that you have to fire up the localhost server each time you want to use it, and it shuts down automatically when you exit the command shell. It is a good simple solution if you are already somewhat familiar with Python and the command line.
- **Server frameworks** including WAMP (for Windows), MAMP (for Mac), and LAMP (for Linux) are a relatively easy way to install an Apache server that runs constantly in the background, making your website files always available through the localhost. These packages include Apache, MySQL database, and PHP; for now, you really only need to worry about Apache. They will install a designated website directory (*www* or *htdocs*), which you simply move your own *my_website* folder into to serve your website. Carefully follow the installation and configuration directions for the framework you choose. Depending on the configuration, you may or may not need to use a port number within the `http://localhost` URL.

> Set up a development server using one of the options listed above and test it by loading your boilerplate into the browser through http://localhost/.

-

> ### Lesson 2 Self-Check:
>
> 1. Which website directory should hold your custom JavaScript files?
>   a. js
>   b. css
>   c. lib
>   d. data
>   e. both js and lib
> 2. True/False: It is a good idea to create one directory on your hard drive in which to place all of your website folders.

Lesson 3: GitHub Setup

I. What is GitHub?

[GitHub](#) is a website and project hosting service that uses the [Git](#) version control system. Git allows you to take a snapshot of your website files at any given time, backing them up and allowing them to be shared with others for collaboration by multiple developers at the same time. GitHub provides an online suite of tools for cloud storage, sharing, collaboration, and hosting of your projects. It has become standard practice for open-source software developers to keep their projects on GitHub, and you will find yourself accessing various *repositories* (or project directories) on the site often. For example, here is a screenshot of the GitHub repository for a previous version of this course (figure 3.1):

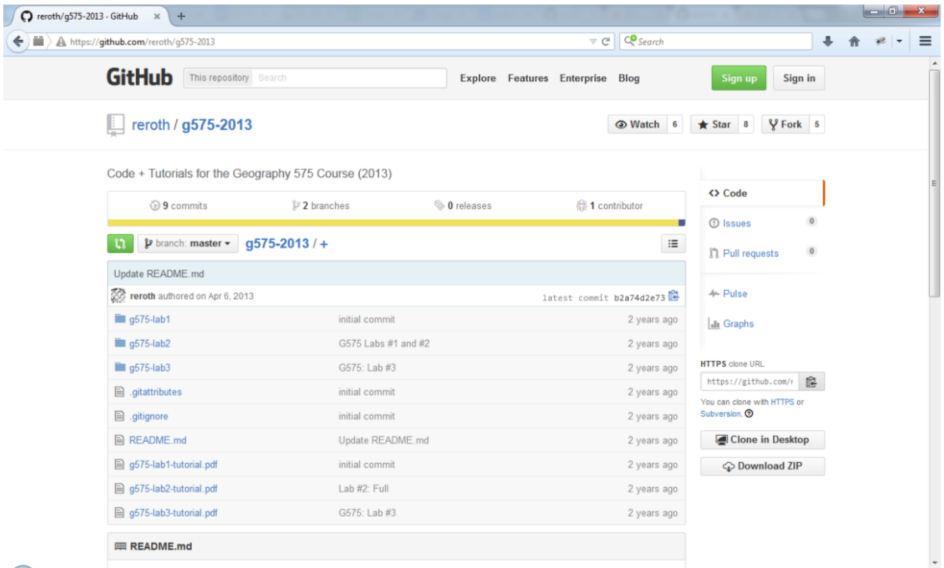


Figure 3.1: a GitHub repository for a previous version of this course

The way Git works on your computer is by creating a repository *within* your website directory, which is then synced with a remote repository on GitHub (or another remote host). It may at first seem counter-intuitive if you are used to uploading and downloading files from a website through your browser. While you can see your files in the remote repository through the GitHub website, you do not use the browser to access them. Rather, the original files sit in the website directory you created on your own machine, and you sync or *push* those files to the remote repository whenever you make changes. If you are collaborating on a project with someone else, they can *clone* your repository from GitHub to their own machine, make changes to website files, then submit a *pull request* asking you to add their changes to the main repository. You can then sync or *pull* their changes from their repository to the main repository and from there into your local website directory. To do all this, you use one of three pieces of software you can download and install onto your machine:

- **Git Bash**: Git's native command line tool, which installs automatically when you install Git from the [Git website](#) (not to be confused with the Git*Hub* website). Available for Mac, Windows, Linux, and Solaris operating systems.
- **Git GUI**: A graphical user interface that also installs with the original Git.
- **GitHub application**: GitHub developed their own graphical user interface to be more beginner-friendly and mesh specifically with the GitHub website (although Git Bash and the Git GUI will also work fine with the GitHub website). Available for [Mac](#) and [Windows](#) .

While the GitHub application provides the simplest, most intuitive interface for Git, it can be tough to make sense of when something goes wrong. Both the Git GUI and Git Bash have a bit of a learning curve to them, but they are both more powerful than the GitHub application and are useful fallbacks when the GitHub application fails. For now, we recommend getting to know your way around the GitHub application unless you already have some experience with Git.

II. Setting Up a Repository

> Read the GitHub Guide ["Getting your project on GitHub"](#) and follow its directions to create a repository in your ***my_website*** directory.

Following the GitHub Guide linked above, the first step to setting up GitHub is to create a GitHub account (if you don't already have one). Go to https://github.com/ , enter a username, e-mail, and password, and click "Sign up for GitHub." Then download the GitHub application using one of the links in the third bullet-point above. If you're a Linux user, you will need to instead install Git and follow the [Git tutorials](#) to learn how to use the Git command-line and GUI tools with GitHub.

If you are on Mac or Windows, once you get the GitHub application installed, open it and drag-and-drop your *my_website* directory to create a local repository for it, following the directions in the GitHub Guide. Alternatively, you can click the + button in the upper-left corner of the application, then click "Create," navigate to the directory *containing* your website directory, enter the name of your website directory, and click the checkmark next to "Create repository" (figure 3.2). Note that if you first navigate to your website directory instead of the folder above it, the application will create a new directory *inside* your website directory with the same name! Check the path shown in the "local path" text field to make sure it looks right.

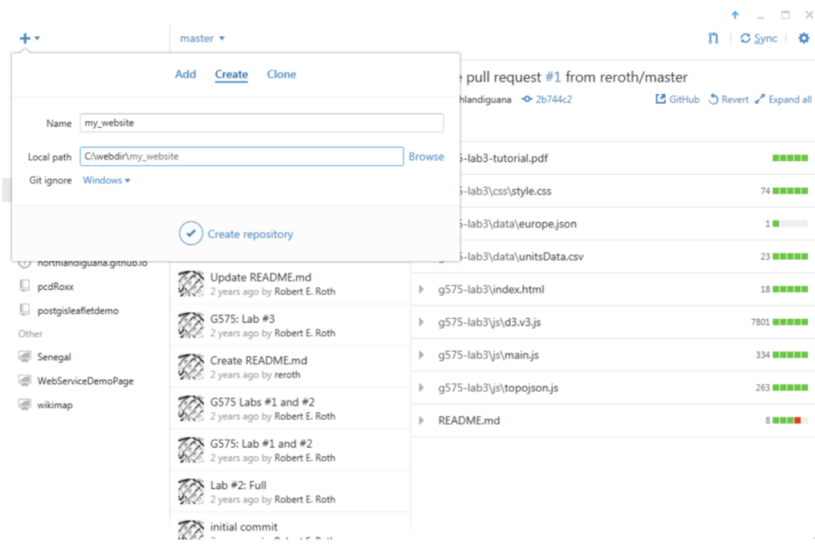


Figure 3.2: creating a repository in your *my_website* directory

Once you've created the repository, open your website directory and observe that three new files have been created (figure 3.3):

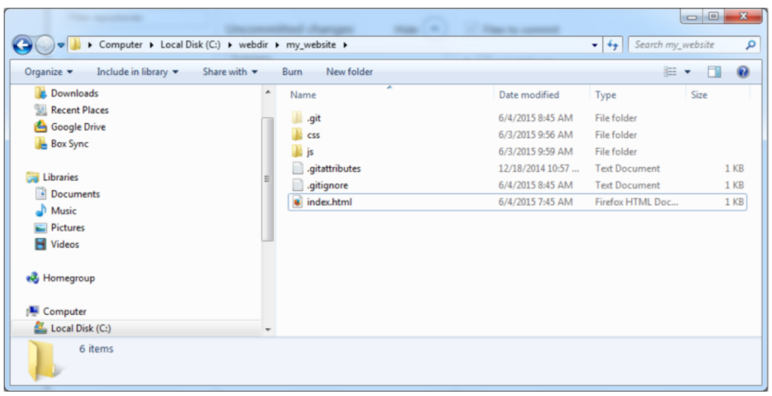


Figure 3.3: website directory with repository files

- **.git**, a hidden folder that holds the snapshots of your files,
- **.gitattributes**, a file specifying settings for the repository,
- **.gitignore**, a file that lists files in the directory that should not be tracked by the repository.

You don't have to worry about these extra files right now, and should *never* tamper with the directories and files inside of the *.git* folder. You can change settings and add to the *.gitignore* file through the GitHub application gear menu→Repository settings in the upper-right corner of the application window. You can also sign in to GitHub, manage your account, and configure Git through the gear menu→Options. If for some reason you wanted to delete a repository entirely, you can simply remove it from the list of repositories on the left side of the GitHub application (right-click→Remove) and delete the *.git* folder and the two other Git files from your website directory.

Follow the rest of the directions in the "Getting your project on GitHub" guide. You should end up with both a local repository in your website directory and a copy of the repository on your GitHub web page.

Note that it is also possible to work in reverse order—that is, create a new repository on the GitHub website and then copy it over, or *clone* it, to your machine. First, learn how to create a repository on the GitHub website:

> Read the GitHub Guide "Hello World".

If you create a repository on the website, there are two ways to add it to your local machine with the GitHub application:

1. On the GitHub repository web page, click the "Clone in Desktop" button in the lower-right corner of the page. Launch the application and save the repository in a logical place.
2. In the GitHub application, click the + button in the upper-left corner, then click "Clone." If you are properly logged in, you should see a list of repositories belonging to your GitHub account (figure 3.4). Click the repository you created, then click the checkmark next to "Clone" and the repository name. Choose a local directory to put your clone in. Once it saves, if you open that directory, you should see your cloned website directory containing all of the site's files, the *.git* folder, and the Git files associated with the repository.

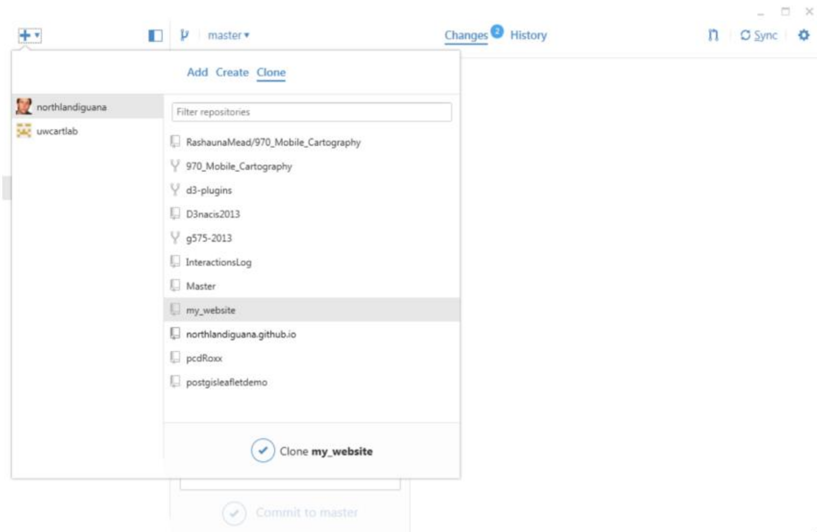


Figure 3.4: GitHub repositories available to clone to the desktop

When you are just figuring things out, you are likely to end up with one or two repositories on GitHub that you won't actually need in the future. If you are sure you no longer need a repository, you can delete it from GitHub by navigating to the repository web page, clicking the "Settings" link on the right-hand side, scrolling down to the "Danger Zone" at the bottom of the page, and clicking "Delete this repository." The GitHub crew was nice enough to take precautions to ensure that you *really* want to delete the repository before you do it! Thus, be absolutely sure you won't need it again, and no one else is contributing to it, before you delete.

III. Learning and Using GitHub

We have required you to read two of the short, excellent GitHub Guides to become familiar with the basics, but we recommend you read through all of them as you will eventually use most or all of the covered skills. There is also an entire YouTube Channel dedicated to GitHub training, and excellent tutorials and videos on the Git documentation page.

From this point, though, the key to really becoming comfortable with GitHub is to use it often. This means that *every time you make changes to your website files, you should commit your changes to your local Git repository and sync it with its remote repository on GitHub.* While you are working on your lab projects, your TA will expect to see your GitHub repository for the project updated frequently. This is an easy way for your TA—and you—to track your progress. It also starts to build a public track record of your work that future employers may look at. Thus, **commit frequently and wisely**!

Lesson 3 Self-Check:

1. How do you create a new Git repository on your machine using the GitHub Application?
   a. + button→Create, navigate to your web server directory, type the name of your website directory, click "Create"
   b. Create a new repository on the GitHub website and click "Clone in Desktop".
   c. Create a new repository on the GitHub website, go to + button→Clone, find the repository in the list, click "Clone", chose your web server directory, click "OK".
   d. Any of the above
2. True/False: The primary way to move files to and from GitHub is through your browser.

Module Deliverables

Lesson Activities—Due END of class THIS week

1. Create a website directory called ***my_website*** with an organized directory structure and boilerplate ***index.html***, ***style.css***, and ***main.js*** files.
2. Create a Git repository in your website directory and sync it to your GitHub account.
3. Submit a zip file (.zip) containing your website directory. In the submission comments, paste a link to your GitHub account page. ***Click the link above to submit your zip file and URL.***

On Your Own—Due BEGINNING of class NEXT week

Complete the Lynda.com "JavaScript Essentials" training ***or*** the CodeAcademy "Language Skills: JavaScript" training. Either training should take 5-6 hours. After you complete the training of your choice, print a PDF of your completion certificate (CodeAcademy) or the table of contents showing all modules have been viewed (Lynda.com) and submit your PDF at the link above.

Appendix 4: 2016-17 Curriculum Evaluation Entrance Survey Questions

Shaded questions indicate matrix headers.

Question	Answer Format
What is your name?	Text
How many classes involving computer programming in some form have you taken previously?	0, 1, 2, 3 or more
How many of these classes have involved the use of HTML, CSS, and/or JavaScript?	0, 1, 2, 3 or more
How many classes have you taken in the UW–Madison Geography Department's Cartography/GIS program previously?	0, 1, 2, 3 or more
How many classes have you taken involving Cartography and GIS outside of the UW–Madison Geography Department?	0, 1, 2, 3 or more
What is your gender?	Male, Female, Other
What is your race/ethnicity?	Text
What is your nationality (i.e, where you come from)?	Text
What is your academic level?	Freshman, Sophomore, Junior, Senior, GIS Certificate Student, Master's student, Ph.D. student, Other (text)
Please rate your expertise with the following Web tools as of today.	
Hyper-Text Markup Language (HTML)	1 – I have never used this tool, 2, 3, 4, 5 – I am an expert with this tool
Cascading Style Sheets (CSS)	1 – I have never used this tool, 2, 3, 4, 5 – I am an expert with this tool
JavaScript	1 – I have never used this tool, 2, 3, 4, 5 – I am an expert with this tool
jQuery	1 – I have never used this tool, 2, 3, 4, 5 – I am an expert with this tool
Asynchronous JavaScript and XML (AJAX)	1 – I have never used this tool, 2, 3, 4, 5 – I am an expert with this tool
The Document Object Model (DOM)	1 – I have never used this tool, 2, 3, 4, 5 – I am an expert with this tool
GitHub	1 – I have never used this tool, 2, 3, 4, 5 – I am an expert with this tool
JavaScript Object Notation (JSON)	1 – I have never used this tool, 2, 3, 4, 5 – I am an expert with this tool
GeoJSON	1 – I have never used this tool, 2, 3, 4, 5 – I am an expert with this tool
TopoJSON	1 – I have never used this tool, 2, 3, 4, 5 – I am an expert with this tool
Scalable Vector Graphics (SVG)	1 – I have never used this tool, 2, 3, 4, 5 – I am an expert with this tool

Comma-Separated Values (CSV)	1 – I have never used this tool, 2, 3, 4, 5 – I am an expert with this tool
Keyhole Markup Language (KML)	1 – I have never used this tool, 2, 3, 4, 5 – I am an expert with this tool
CartoCSS	1 – I have never used this tool, 2, 3, 4, 5 – I am an expert with this tool
Leaflet	1 – I have never used this tool, 2, 3, 4, 5 – I am an expert with this tool
D3	1 – I have never used this tool, 2, 3, 4, 5 – I am an expert with this tool
Google Maps API	1 – I have never used this tool, 2, 3, 4, 5 – I am an expert with this tool
OpenLayers	1 – I have never used this tool, 2, 3, 4, 5 – I am an expert with this tool
ArcGIS Online	1 – I have never used this tool, 2, 3, 4, 5 – I am an expert with this tool
Mapbox Studio	1 – I have never used this tool, 2, 3, 4, 5 – I am an expert with this tool
Carto (formerly CartoDB)	1 – I have never used this tool, 2, 3, 4, 5 – I am an expert with this tool
Optional: Why are you taking Geography 575?	Text
Optional: What would you particularly like to learn or get out of taking Geography 575?	Text

Appendix 5: 2016-17 Curriculum Evaluation Exit Survey Questions

Bold horizontal lines indicate page breaks. Shaded questions indicate matrix headers.

Question	Answer Format
What is your name?	Text
Please rate your agreement with the following statements regarding your knowledge of topics covered in Geography 575 lab after having taken the course.	
I can understand the order of execution of a script.	Strongly disagree, Disagree, Neither agree nor disagree, Agree, Strongly agree
I know how to use different types of script such as variables, functions, if-else statements, and loops.	Strongly disagree, Disagree, Neither agree nor disagree, Agree, Strongly agree
I know how to approach solving problems that make my application break.	Strongly disagree, Disagree, Neither agree nor disagree, Agree, Strongly agree
I still have trouble breaking down large program tasks into manageable functions.	Strongly disagree, Disagree, Neither agree nor disagree, Agree, Strongly agree
I still have trouble using correct syntax in my script.	Strongly disagree, Disagree, Neither agree nor disagree, Agree, Strongly agree
I still have trouble "thinking like a computer program."	Strongly disagree, Disagree, Neither agree nor disagree, Agree, Strongly agree
I know how to tell which code library is being used by which parts of a script.	Strongly disagree, Disagree, Neither agree nor disagree, Agree, Strongly agree
I know how to integrate multiple code libraries to accomplish a task.	Strongly disagree, Disagree, Neither agree nor disagree, Agree, Strongly agree
I can complete a web map from start to finish with minimal direct human assistance.	Strongly disagree, Disagree, Neither agree nor disagree, Agree, Strongly agree
I still have major difficulty with at least one step in the process of making a web map.	Strongly disagree, Disagree, Neither agree nor disagree, Agree, Strongly agree
I am still unclear on which script tasks I should use a code library for instead of native JavaScript.	Strongly disagree, Disagree, Neither agree nor disagree, Agree, Strongly agree
I am still unclear how to figure out which methods can be used with a particular code library.	Strongly disagree, Disagree, Neither agree nor disagree, Agree, Strongly agree
Please rate your expertise with the following Web tools *before* taking Geography 575.	
Hyper-Text Markup Language (HTML)	1 – I had never used this tool, 2, 3, 4, 5 – I was an expert with this tool
Cascading Style Sheets (CSS)	1 – I had never used this tool, 2, 3, 4, 5 – I was an expert with this tool
JavaScript	1 – I had never used this tool, 2, 3, 4, 5 – I was an expert with this tool
jQuery	1 – I had never used this tool, 2, 3, 4, 5 – I was an expert with this tool
Asynchronous JavaScript and XML (AJAX)	1 – I had never used this tool, 2, 3, 4, 5 – I was an expert with this tool
The Document Object Model (DOM)	1 – I had never used this tool, 2, 3, 4, 5 – I was an expert with this tool

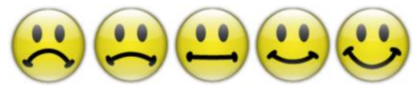
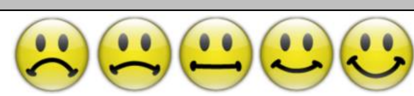
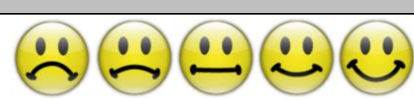
GitHub	1 – I had never used this tool, 2, 3, 4, 5 – I was an expert with this tool
JavaScript Object Notation (JSON)	1 – I had never used this tool, 2, 3, 4, 5 – I was an expert with this tool
GeoJSON	1 – I had never used this tool, 2, 3, 4, 5 – I was an expert with this tool
TopoJSON	1 – I had never used this tool, 2, 3, 4, 5 – I was an expert with this tool
Scalable Vector Graphics (SVG)	1 – I had never used this tool, 2, 3, 4, 5 – I was an expert with this tool
Comma-Separated Values (CSV)	1 – I had never used this tool, 2, 3, 4, 5 – I was an expert with this tool
Keyhole Markup Language (KML)	1 – I had never used this tool, 2, 3, 4, 5 – I was an expert with this tool
CartoCSS	1 – I had never used this tool, 2, 3, 4, 5 – I was an expert with this tool
Leaflet	1 – I had never used this tool, 2, 3, 4, 5 – I was an expert with this tool
D3	1 – I had never used this tool, 2, 3, 4, 5 – I was an expert with this tool
Google Maps API	1 – I had never used this tool, 2, 3, 4, 5 – I was an expert with this tool
OpenLayers	1 – I had never used this tool, 2, 3, 4, 5 – I was an expert with this tool
ArcGIS Online	1 – I had never used this tool, 2, 3, 4, 5 – I was an expert with this tool
Mapbox Studio	1 – I had never used this tool, 2, 3, 4, 5 – I was an expert with this tool
Carto (formerly CartoDB)	1 – I had never used this tool, 2, 3, 4, 5 – I was an expert with this tool
Please rate your expertise with the following Web tools *after* taking Geography 575.	
Hyper-Text Markup Language (HTML)	1 – I have never used this tool, 2, 3, 4, 5 – I am an expert with this tool
Cascading Style Sheets (CSS)	1 – I have never used this tool, 2, 3, 4, 5 – I am an expert with this tool
JavaScript	1 – I have never used this tool, 2, 3, 4, 5 – I am an expert with this tool
jQuery	1 – I have never used this tool, 2, 3, 4, 5 – I am an expert with this tool
Asynchronous JavaScript and XML (AJAX)	1 – I have never used this tool, 2, 3, 4, 5 – I am an expert with this tool
The Document Object Model (DOM)	1 – I have never used this tool, 2, 3, 4, 5 – I am an expert with this tool
GitHub	1 – I have never used this tool, 2, 3, 4, 5 – I am an expert with this tool

JavaScript Object Notation (JSON)	1 – I have never used this tool, 2, 3, 4, 5 – I am an expert with this tool
GeoJSON	1 – I have never used this tool, 2, 3, 4, 5 – I am an expert with this tool
TopoJSON	1 – I have never used this tool, 2, 3, 4, 5 – I am an expert with this tool
Scalable Vector Graphics (SVG)	1 – I have never used this tool, 2, 3, 4, 5 – I am an expert with this tool
Comma-Separated Values (CSV)	1 – I have never used this tool, 2, 3, 4, 5 – I am an expert with this tool
Keyhole Markup Language (KML)	1 – I have never used this tool, 2, 3, 4, 5 – I am an expert with this tool
CartoCSS	1 – I have never used this tool, 2, 3, 4, 5 – I am an expert with this tool
Leaflet	1 – I have never used this tool, 2, 3, 4, 5 – I am an expert with this tool
D3	1 – I have never used this tool, 2, 3, 4, 5 – I am an expert with this tool
Google Maps API	1 – I have never used this tool, 2, 3, 4, 5 – I am an expert with this tool
OpenLayers	1 – I have never used this tool, 2, 3, 4, 5 – I am an expert with this tool
ArcGIS Online	1 – I have never used this tool, 2, 3, 4, 5 – I am an expert with this tool
Mapbox Studio	1 – I have never used this tool, 2, 3, 4, 5 – I am an expert with this tool
Carto (formerly CartoDB)	1 – I have never used this tool, 2, 3, 4, 5 – I am an expert with this tool
Please rate how *challenging* you found each of the following Web tools to learn while taking Geography 575.	
JavaScript	1 – Very easy, 2, 3, 4, 5 – Very challenging
jQuery	1 – Very easy, 2, 3, 4, 5 – Very challenging
Asynchronous JavaScript and XML (AJAX)	1 – Very easy, 2, 3, 4, 5 – Very challenging
The Document Object Model (DOM)	1 – Very easy, 2, 3, 4, 5 – Very challenging
GitHub	1 – Very easy, 2, 3, 4, 5 – Very challenging
JavaScript Object Notation (JSON)	1 – Very easy, 2, 3, 4, 5 – Very challenging
GeoJSON	1 – Very easy, 2, 3, 4, 5 – Very challenging
TopoJSON	1 – Very easy, 2, 3, 4, 5 – Very challenging
Scalable Vector Graphics (SVG)	1 – Very easy, 2, 3, 4, 5 – Very challenging
Leaflet	1 – Very easy, 2, 3, 4, 5 – Very challenging
D3	1 – Very easy, 2, 3, 4, 5 – Very challenging
Below are the lab modules, listed in the sequence in which they occurred. Please answer *both* questions for each module. You may refer to the lab modules and your assignment submissions on the Canvas site for the course if needed.	
How challenging did you find the material presented in each module to understand?	

Module 1: Setting Up Your Workspace	1 – Very easy, 2, 3, 4, 5 – Very challenging, Did not complete
Module 2: Scripting and Debugging	1 – Very easy, 2, 3, 4, 5 – Very challenging, Did not complete
Module 3: Data and AJAX	1 – Very easy, 2, 3, 4, 5 – Very challenging, Did not complete
Module 4: Using Online Resources	1 – Very easy, 2, 3, 4, 5 – Very challenging, Did not complete
Module 5: Leaflet Interactions	1 – Very easy, 2, 3, 4, 5 – Very challenging, Did not complete
Module 6: The Internal Logic of Leaflet	1 – Very easy, 2, 3, 4, 5 – Very challenging, Did not complete
Module 7: D3 Foundations	1 – Very easy, 2, 3, 4, 5 – Very challenging, Did not complete
Module 8: Mapping in D3	1 – Very easy, 2, 3, 4, 5 – Very challenging, Did not complete
Module 9: Coordinated Visualizations	1 – Very easy, 2, 3, 4, 5 – Very challenging, Did not complete
Module 10: Coordinated Interactions	1 – Very easy, 2, 3, 4, 5 – Very challenging, Did not complete
How fun was each module to complete?	
Module 1: Setting Up Your Workspace	1 – Not fun at all, 2, 3, 4, 5 – Very fun, Did not complete
Module 2: Scripting and Debugging	1 – Not fun at all, 2, 3, 4, 5 – Very fun, Did not complete
Module 3: Data and AJAX	1 – Not fun at all, 2, 3, 4, 5 – Very fun, Did not complete
Module 4: Using Online Resources	1 – Not fun at all, 2, 3, 4, 5 – Very fun, Did not complete
Module 5: Leaflet Interactions	1 – Not fun at all, 2, 3, 4, 5 – Very fun, Did not complete
Module 6: The Internal Logic of Leaflet	1 – Not fun at all, 2, 3, 4, 5 – Very fun, Did not complete
Module 7: D3 Foundations	1 – Not fun at all, 2, 3, 4, 5 – Very fun, Did not complete
Module 8: Mapping in D3	1 – Not fun at all, 2, 3, 4, 5 – Very fun, Did not complete
Module 9: Coordinated Visualizations	1 – Not fun at all, 2, 3, 4, 5 – Very fun, Did not complete
Module 10: Coordinated Interactions	1 – Not fun at all, 2, 3, 4, 5 – Very fun, Did not complete
Please add any comments that might help us better understand your ratings above.	Text
Below are the lessons covered within the lab modules, listed in the sequence in which they occurred. Please reorder them in any way you think would have better supported your learning.	
Boilerplates and Frameworks	Order (1)

Web Directory Setup	Order (2)
GitHub Setup	Order (3)
Exploring the DOM	Order (4)
JavaScript Basics	Order (5)
jQuery Basics	Order (6)
Debugging in the Developer Console	Order (7)
Web data formats and their geospatial variants	Order (8)
AJAX concepts and syntax	Order (9)
AJAX callback functions	Order (10)
Leaflet tutorials and API	Order (11)
Using examples	Order (12)
Using help forums	Order (13)
Finding tilesets and data	Order (14)
Making Leaflet layers dynamic	Order (15)
Zoom, Pan, and Retrieve interactions (pop-ups and info panel)	Order (16)
Sequence interaction (slider and skip buttons)	Order (17)
Additional interaction operators	Order (18)
Object-oriented JavaScript	Order (19)
Extending Leaflet objects (temporal legend)	Order (20)
Using SVG Graphics (attribute legend)	Order (21)
D3 selections and blocks	Order (22)
Data (in D3)	Order (23)
Scales, Axes, and Text	Order (24)
D3 Helpers: TopoJSON, MapShaper, and Queue	Order (25)
D3 Projections and Path Generators	Order (26)
Dynamic map styling	Order (27)
Drawing a bar chart	Order (28)
Dynamic attribute selection	Order (29)
Transitions	Order (30)
Linking interactions between map and chart	Order (31)
Please list any of the above lessons that should have included more complete information or directions.	Text
Please list any of the above lessons that were not very useful for your learning and could be removed.	Text
Please list any of the above lessons that should have been modified in specific ways, and describe how you would modify them.	Text
Please list any other topics that you wish had been included but weren’t.	Text
Please rate how effectively each of the following resources supported your learning.	

Figures in lab modules	Very ineffective, Ineffective, Neither effective nor ineffective, Effective, Very effective, I did not use this resource
Code examples in lab modules	Very ineffective, Ineffective, Neither effective nor ineffective, Effective, Very effective, I did not use this resource
Professor assistance	Very ineffective, Ineffective, Neither effective nor ineffective, Effective, Very effective, I did not use this resource
Written directions in lab modules	Very ineffective, Ineffective, Neither effective nor ineffective, Effective, Very effective, I did not use this resource
Self-checks in lab modules	Very ineffective, Ineffective, Neither effective nor ineffective, Effective, Very effective, I did not use this resource
Lab assignment example code	Very ineffective, Ineffective, Neither effective nor ineffective, Effective, Very effective, I did not use this resource
Weekly module assignments	Very ineffective, Ineffective, Neither effective nor ineffective, Effective, Very effective, I did not use this resource
TA assistance	Very ineffective, Ineffective, Neither effective nor ineffective, Effective, Very effective, I did not use this resource
Lecture notes	Very ineffective, Ineffective, Neither effective nor ineffective, Effective, Very effective, I did not use this resource
Piazza Q&A boards	Very ineffective, Ineffective, Neither effective nor ineffective, Effective, Very effective, I did not use this resource
Canvas discussion boards	Very ineffective, Ineffective, Neither effective nor ineffective, Effective, Very effective, I did not use this resource
Peer assistance	Very ineffective, Ineffective, Neither effective nor ineffective, Effective, Very effective, I did not use this resource
The following questions ask about your experience learning **basic web development and data concepts** during the first unit of Geography 575 lab (*before* the Leaflet Lab was assigned). If you need a reminder of what specific topics were covered in Unit 1, please refer to the lab modules posted to the Canvas course website.	
What was your overall **emotional** experience with learning basic web development and data concepts presented in Unit 1 of Geography 575 Lab?	
How much **time** did you spend completing the lab modules in Unit 1?	A great deal, A lot, A moderate amount, A little, None at all
What major **breakthroughs or 'aha!' moments** did you experience during the first unit of lab?	Text

What major **<u>frustrations or blockages</u>** did you experience during the first unit of lab?	Text
After completing the course, how difficult would you find it to make a simple web application that loads data with AJAX and displays it in the browser?	Very difficult, Difficult, Neutral, Easy, Very easy
The following questions ask about your experience learning **<u>Leaflet</u>** during the <u>second unit</u> of Geography 575 lab, *including* while completing the Leaflet Lab assignment. If you need a reminder of what specific topics were covered in Unit 2, please refer to the lab modules posted to the Canvas course website.	
What was your overall **<u>emotional</u>** experience with learning <u>Leaflet</u> during Unit 2 of Geography 575 Lab?	
How much **<u>time</u>** did you spend completing the lab modules in Unit 2?	A great deal, A lot, A moderate amount, A little, None at all
What major **<u>breakthroughs or 'aha!' moments</u>** did you experience during Unit 2?	Text
What major **<u>frustrations or blockages</u>** did you experience during Unit 2?	Text
After completing the course, how difficult would you find it to make a custom web map using Leaflet?	Very difficult, Difficult, Neutral, Easy, Very easy
The following questions ask about your experience learning **<u>D3</u>** during the <u>third unit</u> of Geography 575 lab, *including* while completing the D3 Lab assignment. If you need a reminder of what specific topics were covered in Unit 3, please refer to the lab modules posted to the Canvas course website.	
What was your overall **<u>emotional</u>** experience with learning <u>D3</u> during Unit 3 of Geography 575 Lab?	
How much **<u>time</u>** did you spend completing the lab modules in Unit 3?	A great deal, A lot, A moderate amount, A little, None at all
What major **<u>breakthroughs or 'aha!' moments</u>** did you experience during Unit 3?	Text
What major **<u>frustrations or blockages</u>** did you experience during Unit 3?	Text
After completing the course, how difficult would you find it to make a custom web map using D3?	Very difficult, Difficult, Neutral, Easy, Very easy
The following questions ask about your experience completing the **<u>Final Project</u>** of Geography 575, *including* the project proposal process.	
What was your overall **<u>emotional</u>** experience with completing the <u>Final Project</u> of Geography 575?	
How much **<u>time</u>** did you spend completing the Final Project?	A great deal, A lot, A moderate amount, A little, None at all
What major **<u>breakthroughs or 'aha!' moments</u>** did you experience while working on the Final Project?	Text

What major **<u>frustrations or blockages</u>** did you experience while working on the Final Project?	Text
After completing the course, how difficult would you find it to work collaboratively with peers or colleagues to produce a professional web map-based application from design through publishing?	Very difficult, Difficult, Neutral, Easy, Very easy
How likely are you to use what you learned in Geography 575 professionally?	Very unlikely, Unlikely, Undecided, Likely, Very likely
How would you improve the overall Geography 575 learning experience for future students?	Text

Appendix 6: A Model Web Mapping Syllabus

Syllabus: Web Mapping

Geography 2056 - Fond du Lac Tribal and Community College - Spring 2019
Class meetings: TBD, GIS Lab (Room 208)

Syllabus and schedule are subject to minor changes at the discretion of the course instructor.

Instructor

Carl M. Lemke Oliver Sack (aka Carl Sack), carl.sack@fdltcc.edu

Office Hours

I maintain an open office in room W222 on the Fond du Lac campus most weekdays. My posted office hours are **TBD**. Otherwise come talk to me any time I'm in my office. In the event I am away from campus on a regular work day, I will post a note on my door and will respond to emails as I am able.

Course Overview

This course covers the creation of both static and interactive online maps. Course topics include the basics of internet architecture, web data formats, web services, web cartography, UI/UX (user interface/user interaction) design, and publishing on the web using HTML, CSS, and JavaScript. Students will create shareable web maps on real-world topics using both graphical mapping platforms and JavaScript code-based APIs. No prior coding experience is necessary.

Course Learning Objectives

- Distinguish between static and interactive web maps
- Construct a basic web page and publish it to a localhost server
- Construct, publish, and share a customized thematic web map using a graphic online mapping platform
- Construct a basic interactive web map using appropriately formatted data and HTML, CSS, and JavaScript
- Design and embed in a web page a static map image that effectively represents a real-world problem or issue
- Describe different interactions that can take place on an interactive web map
- Find and use online tutorials, examples, and resources to solve problems in program code

Additionally, this course is designed to meet the learning objectives included in the Geographic Information Science and Technology Body of Knowledge Web Mapping topic.

Textbook

There is no textbook for this course. We will rely on free and open resources published online.

What you can expect from me

I try to bring passion and enthusiasm to the topics I teach. I intend to lay out course expectations in a clear and concise manner, and to be open to constructive feedback. I will be hands-on in providing assistance, assessing your work regularly, and helping you to improve your skills. I will be as responsive as I can to e-mails I receive from students in the class; typically, this means that I will get back to you within 24 hours (please allow for a little longer on the weekends). If you need special accommodation, please contact me as soon as possible and let me know.

What I Expect from You

Some of the material we cover will be quite difficult and will require you to learn to think in new ways. I expect you to give it your best effort and not be daunted. Failure is not an option—it is inevitable, and it can be the best kind of learning experience. I expect you to show up to class on time for each meeting unless you have a legitimate reason (e.g., personal illness, child care disruption, family emergency, etc.) and have notified me in advance. I understand that things happen; please keep me up to date so it doesn't negatively impact your grade or cause you to fall too far behind in the coursework. I hope you will be enthusiastic about learning how to create web maps and come to class each time prepared to do something you've never done before.

Course Feedback

I will collect feedback on the course through anonymous surveys at midterm and at the end of the course. For specific problems or questions related to your work and grade, please contact me directly.

Course Structure and Activities

The topic sequence is summarized in the schedule at the end of the syllabus. Course activities include:

Integrative lab assignments and Final Project

The course is broken up into four units, with each unit centering around a different type of web map product you will create. The first three assignments will use the same U.S. Census dataset, which you will select and curate. The final project will use a different dataset of your choosing. After completing each assignment, you will be required to critique your work and assign yourself a score based on the usefulness and usability of the web map(s) you created. I will try to assign grades based on your self-assessments, but will adjust your score if I feel it is warranted. The four assignments are:

1. *An ArcGIS Online Web App.* This assignment will give you experience creating and configuring a simple interactive web map with the powerful graphic user interface (GUI) tools provided by Esri's ArcGIS Online platform.

2. *A Map Story Web Page:* This assignment will give you an understanding of web-based storytelling techniques and experience with designing and formatting static maps for the web, embedding them in a web page and creating other page content using HTML, and styling the web page using CSS to improve its usability and responsiveness at multiple screen sizes.

3. *A Leaflet Slippy Map:* This assignment will give you experience creating geospatial web services, setting up a web development environment, writing and debugging JavaScript, using Leaflet to load external data layers and add map interactions, and using JavaScript and CSS to add map symbolization and page styling.

4. *Final Project:* This assignment will require you to design and develop an interactive web map addressing a real-world problem or issue you are interested in. You are encouraged to go above and beyond the requirements of the three main lab assignments in the final project, implementing custom solutions that address your particular theme. You will be required to publish your final map on a live website.

Weekly deliverables and exercises
Different components of each lab assignment will be due each week, leading up to the final product. You will receive 10 points for completing each week's benchmark task, or a smaller amount depending on your progress toward completion at the end of the week. Some weeks, you will be given additional small exercises intended to hone in on a particular concept or skill introduced in class, each also worth 10 points toward this grade category.

Exams
The course will include a Midterm Exam and a Final Exam. Both will be comprehensive, covering the vocabulary and concepts taught up to that point in the course. Each exam will also include a practical component requiring you to analyze and/or critique a web map.

Grading

Percentages of your final grade:
Attendance and Participation: 10%
Integrative Lab Assignments and Project: 60% (15% each)
Weekly Deliverables and Exercises: 10%
Exams: 20%

Final grade breakdown:
A: 91-100%
B: 80-90%
C: 70-79%
D: 60-69%
F: <60%

I reserve the right to curve grades upward based on the class distribution of final grades. You will never get a lower grade based on your score than what is indicated above.

Late Work

Late work will be deducted 15% if turned in within one week of the due date, and 50% thereafter. Extensions will be granted on a case-by-case basis with at least 24 hours advanced notice given before the assignment due date. No work will be accepted beyond the end date of the course.

Plagiarism

Creating web maps is different from writing essays, in that you are not only allowed but encouraged to copy others' code that you find online, so long as you create a unique end product before claiming it as your own work. Open source software is built by communities of developers who copy, alter, and improve upon each others' work. The conditions are simply that you need to give proper credit to contributors in comments within the code, and sometimes you need to include the license that the code is published under. As with any mapping project, you also need to cite your data sources and use data in accordance with the terms of any license applied to it. Learning to read and abide by license terms is an important web mapping skill that will be covered in the course. Failure to give proper credit or abide by license terms will result in points deducted from the assignment, while passing off others' products that you have not significantly altered as your own work is considered plagiarism and will result in a 0. Multiple instances of plagiarism will be reported to the Dean of Students.

Course Schedule

This schedule should be considered approximate and subject to adjustment.

Week	Dates	Topic	Activities
1	1/14-18	Client-server architecture; definition and types of web maps; browser tools	Web Map Scavenger Hunt
2	1/22-25	Layer 1: Data—models, geometries, types, and levels	Lab 1—ArcGIS Online Map: Prepare thematic datasets from U.S. Census
3	1/28-2/1	Layer 2: Representation—symbolization, visual hierarchy, tilesets	Lab 1: Symbolize each data layer and choose an appropriate basemap
4	2/4-2/8	Layer 3: Interaction—stages, operators, interface affordances & feedbacks	Lab 1: Use Web AppBuilder to add interactions and publish the map
5	2/11-15	Storytelling on the web, responsive web design	Self-critique of Lab 1 Lab 2—Map Story Web Page: Determine theme, storyboard, and prepare ArcGIS layout
6	2/18-22	Raster image formatting and style guidelines for static web maps	Lab 2: Symbolize, label, and export PNG maps Codecademy HTML certificate due
7	2/25-3/1	Introduction to HTML; text editors; web reference guides; the DOM; browser Elements tab	Lab 2: Create a web page with embedded PNG maps, captions, and scrolling links
8	3/4-8	Introduction to CSS; browser styles sandbox; **Midterm exam**	Lab 2: Add CSS page styles and media queries Codecademy CSS certificate due
9	3/18-22	Projections on the Web; OGC web services; SLD Stylesheets	Self-critique of Lab 2 Lab 3—Leaflet Slippy Map: Create GeoServer Web Map Service and Web Feature Service
10	3/25-29	Web directory setup; JavaScript data types, functions, methods; Console and debugging;	Lab 3: Set up a localhost server and web directory Debugging Practice Assignment
11	4/1-5	Leaflet API; GeoJSON; AJAX	Lab 3: Load Leaflet basemap, WMS, and WFS layers Codecademy JavaScript certificate due
12	4/8-12	JavaScript control flow; Leaflet interactions	Lab 3: Symbolize WFS layer, add pop-ups and layers control, finalize page styling
13	4/15-19	Concept review and final project assistance	Self-critique of Lab 3 Final Project—Published Web Map

14	4/22-26	Final project assistance	Final Project
15	4/29-5/3	Final project assistance and exam review	Final Project
16	5/6-7	Final project	Final Project
17	TBD	**Final exam**	Final Project due 5/14