

Geography 485L/585L Weekly Breakdown

Karl Benedict

Spring 2016

Contents

1	Goals and Objectives	5
2	Week 1 - Introductions, Course Outline & Web Concepts	7
	Class Prep	7
	Reference Materials	7
	Weekly Milestone - Creating Your GitHub Repository and First Web Page	7
3	Week 2 - Module 2a - Web-based Mapping Clients: HTML, CSS & Javascript	11
	Class Prep	12
	Reference Materials	12
	Weekly Milestone - Create a More Complex Web Page and Style It	12
4	Week 3 - Module 2a - Web-based Mapping Clients. Google Maps API	15
	Class Prep	15
	Reference Materials	16
	Weekly Milestone - Creation of a Web Page with an Embedded Google Map	16
5	Week 4 - Module 2a - Web-based Mapping Clients. Google Maps API	17
	Class Prep	17
	Reference Materials	18
	Weekly Milestone - Styling of an Embedded Google Map	18
	Deep Dive - Creation of a a Google Maps Web Page with Custom Points and Labels	18

Chapter 1

Goals and Objectives

Internet mapping technologies are an important component of geospatial data capture, sharing, visualization, and delivery. This course provides a survey of current and emerging internet and geospatial interoperability standards, technologies, and capabilities. The emphasis of the work in this class will be hands-on experience in four critical aspects of Internet-enabled mapping:

- The basic concepts behind web development and web mapping technologies that enable the delivery of maps and mapped data through web browsers
- The Open Standards that facilitate the exchange of map images and geospatial data over the internet
- The use of published standards-based services in desktop mapping applications that implement those standards
- The deployment of standards-based geospatial map and data services that other systems and users may make use of

The specific class objectives that relate to these activities and departmental curriculum objectives for undergraduate and graduate students in the Geography Department include the following:

- Students will understand the concepts geospatial data and service interoperability
- Students will be able to define the specific requirements of a particular analysis or project and identify the interoperability standards that are capable of meeting those requirements
- Students will be knowledgeable in the core technologies that they may use to produce their own internet-enabled mapping capabilities
- Students will understand the strengths and limitations of current internet mapping technologies for generating cartographically effective map products.

The weekly goals, objectives and assignments are outlined below

Chapter 2

Week 1 - Introductions, Course Outline & Web Concepts

This week we will review the content and structure for the course and spend some time getting to know each other. Following this we will spend some time setting up some of the tools that you will be using for the course in developing your portfolio of materials.

Class Prep

- [Wikipedia article - History of the World Wide web](#)
- [Lynda.com tutorials](#)
 - *Web Design Fundamentals*
 - * Introduction
 - * 1. Exploring Web Design
 - *Version Control for Everyone*
 - * Introduction
 - * 1. Introducing Version Control
 - * 2. Version Control Basics
 - * 3. Setting Up Your First Project

Reference Materials

[Class Syllabus](#)

Weekly Milestone - Creating Your GitHub Repository and First Web Page

Developing content to go onto the web has evolved from a solitary effort to one where teams work together in developing components of larger web sites. These teams need to have a variety of tools to enable their work. Some of the most important tools enable code sharing with the team, and in projects based on the [Open](#)

Source software model the rest of the world. The [GitHub](#) web platform uses the [Git](#) distributed [version control](#) system to enable sharing of code and hosting static web pages based on that shared code.

You will be using a private [GitHub](#) repository to build your class portfolio during the course. If you would like to make your portfolio available publicly you can also use GitHub as the platform for providing that public access. Regardless of your decision about providing public access to your portfolio, you will learn how version control operates, and how to provide comments and keep notes on your work and comment on the work of others (this will be part of our peer review process).

While the work we do this and next week will be directly through the editor integrated into the GitHub system, you will eventually need to install a desktop application (such as the [SourceTree](#) application recommended for the class) that allows you to develop your web pages on your local computer and then update the files on the GitHub system when you want to share a new version. Also, you can't add things like images to your web pages until you are adding them to a local repository on your computer and then sending them GitHub.

For this milestone we will walk through the process of creating your repository in GitHub, creating your first web page, previewing that page on your local computer, changing the page, and updating the page on LoboGit. For this milestone we will do this as a manual process which we will streamline in the coming weeks.

Step 1 - Create Your GitHub Account and Portfolio Repository

For your work in this class you will build your portfolio within an organization (<https://github.com/UNM-GEOG-485-585>) within GitHub that has been created for the class. The first step in the process of creating your portfolio is to create a new *repository* in GitHub within which you will put your portfolio materials for sharing within the class. Please follow the following steps to create your repository:

1. Go to the [GitHub homepage](#) and follow the onscreen instructions for creating a new account. If you already have an account you can skip this step.
2. Come to the front of the class and tell me your GitHub username so that I can add you to the organization and create your repository for you within the organization.

Step 2 - Create Your First Web Page

To create your first web page within your portfolio repository you need to first enter your repository, add a new file, modify its contents, and commit your modifications back to the repository to save your changes.

1. Go to the class organization page ((<https://github.com/UNM-GEOG-485-585> - logging in if necessary) and click on your repository name in the list.
2. On the page that comes up listing the files in your repository, click the “New File” button above the list of files.
3. Enter the name of the file that you are creating as “hello-world.html”
4. Enter the following text into the text entry area under the filename field.

```

1 <html>
2   <head>
3   </head>
4   <body>
5     Hello World !!!
6   </body>
7 </html>
```

5. Add a brief comment (such as “Created hello-world.html from provided text”) in the first field under the “Commit new file” title. You can optionally add a more detailed description in the next field if you like.

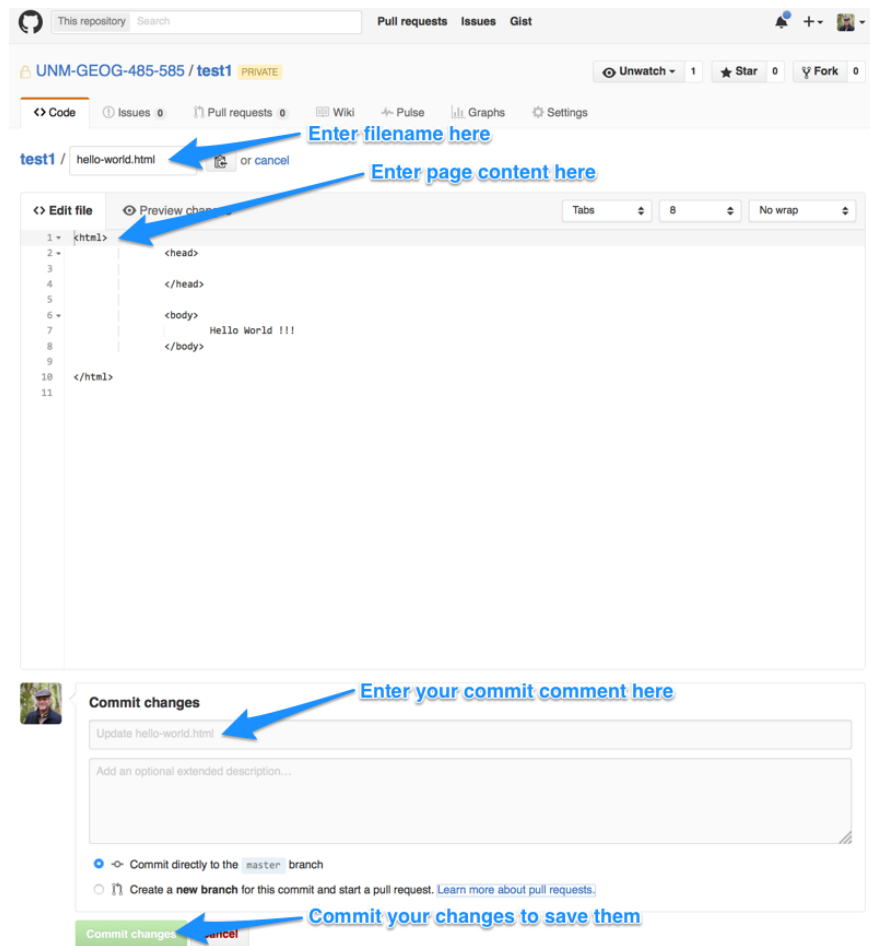


Figure 2.1: GitHub file creation/editor page

6. Keep the default option to “Commit directly to the master branch”
7. Click the “Commit New File” button to commit your change and save the file

Step 3 - Preview Your Web Page in a Browser

While we will later discuss strategies for hosting your web content on a system (GitHub for example) that supports direct access by web clients, to preview the web page you just created you need to download the repository to your local computer where you can open the locally stored file in a browser.

1. Go to the class organization page ((<https://github.com/UNM-GEOG-485-585> - logging in if necessary) and click on your repository name in the list.
2. On the page that comes up listing the files in your repository, click the “Download Zip” button above the list of files. You may be prompted to provide a download location - if not you will need to find the default download location. Often it is the “Downloads” folder in your home directory.
3. Extract the contents of the downloaded .zip file using the appropriate utility program on your computer. On Macs and Windows computers this functionality is available through right-clicking on the file name in their respective file browsers.
4. Once you have extracted the contents of the zip file open the `hello-world.html` file that you created in a web browser - typically if you double-click on the file it will open in your default browser. You can also open it from within your browser of choice by using the “Open File” (or similar) option in the browser’s file menu.
5. Confirm that the display resembles something like the following:

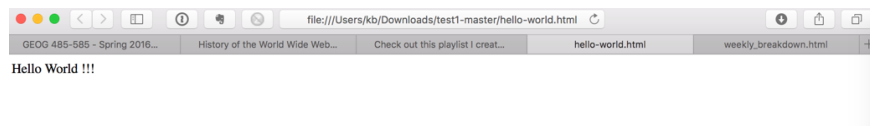


Figure 2.2: Sample `hello-world.html` file when viewed in a web browser

6. If the page does not appear as you like, edit it on GitHub and repeat 2-5 above until you see something like the sample figure.

Chapter 3

Week 2 - Module 2a - Web-based Mapping Clients: HTML, CSS & Javascript

This week we will begin to build our foundation for developing material to be shared over the Internet via the World Wide Web. In particular we will cover the basic process of web development, define the parts of a web page, and spend some time learning about the different *languages* and define the key components of a web page: its structure, presentation, and behavior.

The presentation of information over the Internet is dependent upon the use of standards that have been developed for defining the *structure*, *presentation*, and *behavior* of content. This week we will begin working with the key technologies that define these three components of web content.

These concepts will be illustrated through reference to several simple web pages which are progressively modified to integrate all three of these components.

Expected Outcomes

By the end of this class module you should understand the following:

- The basic process of web development
- The parts of a web page
- The role of the three web page components: *structure*, *presentation*, and *behavior*
- Be able to write your own basic web page with your own content and make it available over the web

Key Concepts

- Parts of a web page
- Structure = X/HTML
- Presentation = CSS
- Behavior = Javascript
- Iterative Development

Class Prep

- [Lynda.com tutorials](#)
 - *Web Design Fundamentals*
 - * 3. Getting Started
 - * 4. Exploring Tools
 - *Version Control for Everyone*
 - * 5. Basic Project Sharing
 - * Conclusion
- Duckett, Jon, and Larsen, Rob. *Beginning HTML and CSS*. Somerset, NJ, USA: John Wiley & Sons, 2013. ProQuest ebrary. Web. 28 December 2015. This book is available online through the [University Library](#) - Chapters 1, 7, 10

Reference Materials

- Duckett, Jon, and Larsen, Rob. *Beginning HTML and CSS*. Somerset, NJ, USA: John Wiley & Sons, 2013. ProQuest ebrary. Web. 28 December 2015. This book is available online through the [University Library](#) - Chapters 2,3,4 and 8
- [Lynda.com tutorials](#)
 - *CSS Fundamentals*
 - *Javascript for Web Designers*

Weekly Milestone - Create a More Complex Web Page and Style It

This week's milestone activity takes you through the process of creating two more web pages in preparation for next week's work with the Google Maps API in developing your first web mapping page. These pages will be:

1. A *home page* for your portfolio that will be the access point for all of the materials you create ([template/preview](#)), and
2. Your first web page containing materials related to this *milestone* assignment ([template/preview](#)).

Step 1 - Open the *home page* [template](#) linked above in your web browser and open the [preview](#) in a second tab or window so that you can view both at the same time.

Step 2 - Copy the code in the home page [template](#) into a new text file named `index.html` on your computer.

Step 3 - Open the *milestone* assignment [template](#) linked above.

Step 4 - Copy the code in the template into a new text file named `milestone_02.html` on your computer.

Step 5 - After you have saved the `index.html` and `milestone_02.html` files to your hard drive open them up in your browser to see what they look like when read through a web browser.

Step 6 - Add your responses to the following questions to the `milestone_02.html` document. - note: it is a good practice when you are developing a web page to make small changes, save them, and preview the page to make sure that you have not made an error in your code before adding the next item. Practice this by adding each answer, saving your page and previewing it and correcting any errors in your code before going onto the next question.

Question 1 From examining the display of `index.html` in your web browser and the structure of the source code in the page, what effect (if any) does the white space (i.e. tabs, blank lines, multiple spaces) have on what is displayed in the browser?

Question 2 How are the

```
<h1>
```

and

```
<h2>
```

elements from the source code displayed differently in the browser?

Question 3 What type of element would you use to create additional list elements in either the “topic” or “data type” (``) lists on the page.

Step 7 - Flesh out the `index.html` page that you created above (*Step 2*) with information specific to you based upon the content areas in the page. After making sure that your `index.html` and `milestone_02.html` are in the same directory, add a *relative* link to your `milestone_02.html` file to the “milestones” section of your `index.html` page by modifying the line

```
<p><a href="">Milestone 2</a></p>
```

to look like this

```
<p><a href="milestone_02.html">Milestone 2</a></p>
```

Save your change and test it in the browser by clicking the link on your `index.html` page in the browser. If it successfully opens your `milestone_02.html` page you have properly built your link.

Step 8 - Copy your `hello-world.html` file from *Milestone 1* into the same directory as your `index.html` file and modify the existing line in your `index.html` file

```
<p><a href="">Hello World</a></p>
```

to link to your `hello-world.html` file (follow the same pattern you used in *Step 7* above).

Step 8 - Make a copy of your `index.html` page by copying the content of the page and pasting it into a new document named `index_styled.html`.

Experiment with some of the styling capabilities described in Dave Raggett’s “Adding a Touch of Style” page (<http://www.w3.org/MarkUp/Guide/Style.html>) on `index_styled.html` page you created above. Make at least three stylistic changes to the `index_styled.html` page. Add a link to your `index_styled.html` page to your home page (`index.html`) under the milestones section.

Step 9 - Transfer your created files `index.html`, `milestone_02.html`, and `index_styled.html` to your GitHub repository (created in *Milestone 1*). Of course you could do this by copying and pasting the content of your files into corresponding files in GitHub (but that would not be very efficient or satisfying), but you should probably experiment with [SourceTree](#) as demonstrated in this week’s Lynda.com video tutorial as a way to work locally and transfer your files to GitHub for remote access and sharing.

Chapter 4

Week 3 - Module 2a - Web-based Mapping Clients. Google Maps API

This week we will begin our work with the popular Google Maps *Application Programming Interface* (API) in developing an interactive web-based mapping client. This development activity will build upon the the work you've done over the last couple of weeks in developing basic web pages by using the capabilities that Google has made available for building mapping interfaces based upon their Maps platform. You will begin working with javascript as a client programming language to both interact with Google's servers and to provide the needed information for Google's mapping tool in your web page.

Expected Outcomes

By the end of this class module you should understand the following:

- What an Application Programming Interface (API) is
- How Javascript can be used to define the behavior of elements in a web page
- What the basic structure of a javascript code block for defining a Google Maps - enabled page looks like
- How to write a basic web page that includes an interactive Google Map

Key Concepts

- Application Programming Interface (API)
- Javascript and its location within an HTML page
- The interaction between javascript behaviors and structural elements in a web page

Class Prep

- [Lynda.com tutorials](#)
 - *Javascript for Web Designers* (included as a reference source last week)
 - * 5. Using the Google Maps API

- Svennerberg, Gabriel. *Beginning Google Maps API 3*. Apress, © 2010. Books24x7. Web. Dec. 28, 2015. [Books 24x7 Library Database](#) - if this direct link to the book doesn't work for you, try [logging in first](#) and searching for Google Maps API - the Svennerberg book will be the first item on the list. 1-3 (skim chapter 2)

Continue reviewing:

- Duckett, Jon, and Larsen, Rob. *Beginning HTML and CSS*. Somerset, NJ, USA: John Wiley & Sons, 2013. ProQuest ebrary. Web. 28 December 2015. This book is available online through the [University Library](#) - Chapters 1, 7, 10

Reference Materials

- Duckett, Jon, and Larsen, Rob. *Beginning HTML and CSS*. Somerset, NJ, USA: John Wiley & Sons, 2013. ProQuest ebrary. Web. 28 December 2015. This book is available online through the [University Library](#) - Chapters 2,3,4 and 8
- [Google Maps API Tutorial](#)

Weekly Milestone - Creation of a Web Page with an Embedded Google Map

In preparation for creating a web page with an embedded Google Map you should first answer the following questions about what and how you want to map. As you define the type of map you want to build, think about a specific problem or topic that you would like to address with your map.

In this exercise you will be generating the configuration for the base map (i.e. The Google Maps background layers). In future assignments you will add your own custom content to free-standing web pages that include a mapper based upon the base map you define here.

Create a web page (based upon the assignment [template](#)) that contains your milestone writeup (including the embedded Google Map required by question 5), and link it to the home page (`index.html`) file you created last week.

Respond to Question 1-4 with an understanding that you are generating a web page that is designed for public viewing (even if you don't choose to make it public at this time), and should be both clear and complete.

Question 1 What area do you want to depict in your map? Why?

Question 2 What is the center point (latitude and longitude) of your area of interest?

Question 3 What style of map (roads, satellite, hybrid, terrain) is appropriate for your map? Why?

Question 4 What is the scale of your map (local, regional, continental, global)? How will this translate into your selection of an appropriate default zoom level for your map?

Now that you have answered these questions about the map that you want to create, refer to the examples in the lecture notes, the [Google Maps Tutorial](#), and this week's reading ([link to the code for Svennerberg's Chapter 3 example](#)) and video tutorial assignment to create a custom Google map.

Question 5 Embed a Google Map in your writeup that is based upon your responses to questions 1-4 above.

Chapter 5

Week 4 - Module 2a - Web-based Mapping Clients. Google Maps API

This week covers some additional topics related to the Google Maps API, particularly focusing on styling the Maps base maps using the [styled maps wizard](#) and integrating the javascript generated by the wizard into the base web page code developed last week; and, using [Google's Fusion Tables](#) tool to create and manage tabular data for mapping and other visualization. We complete our work with the Maps API with an example of a more “real” example of a maps-enabled web page.

Expected Outcomes

By the end of this class module you should be able to:

- Generate a Google Maps JSON style using the *Styled Maps Wizard*
- Integrate that JSON into your map client page for styled basemap display

You should also understand

- The potential of *Fusion Tables* as an alternative source of data to integrate into a custom Google Map page
- The potential structure of an *operational* web page, including the physical separation of page components (structure, presentation, behavior) into separate files

Key Concepts

- Generating Google Maps styles
- Integrating styles into a Google Maps page
- Fusion tables as a data source for Google Maps maps
- Separation of structure, presentation, and behavior in web development

Class Prep

- [Lynda.com tutorials](#)
 - *Javascript for Web Designers* (continued)
 - * 5. Using the Google Maps API

Reference Materials

- Duckett, Jon, and Larsen, Rob. *Beginning HTML and CSS*. Somerset, NJ, USA: John Wiley & Sons, 2013. ProQuest ebrary. Web. 28 December 2015. This book is available online through the [University Library](#) - Chapters 2,3,4 and 8
- [Google Maps API Tutorial](#)
- Svennerberg, Gabriel. *Beginning Google Maps API 3*. Apress, © 2010. Books24x7. Web. Dec. 28, 2015. [Books 24x7 Library Database](#) - if this direct link to the book doesn't work for you, try [logging in first](#) and searching for Google Maps API - the Svennerberg book will be the first item on the list. 4-8
- [Google Maps Fusion Mapper](#)

Weekly Milestone - Styling of an Embedded Google Map

Make a free-standing web page based upon the Google Map that you created as part of last week's lab assignment. Use the Google [styled maps wizard](#) to define *at least* three modified base map styles and integrate the JSON generated by the wizard into your new Google Map page.

Deep Dive - Creation of a a Google Maps Web Page with Custom Points and Labels

In your milestone for Week 4 you built a styled Google Maps base map for a particular region of interest. For this *deep dive* assignment create a new free-standing web page that includes a brief description of the topical focus of your mapper:

- The type of information that you want to depict in your map
- Your reasons for selecting the specific area shown in the map
- A description of what you are trying to communicate with the map

Embed the base map that you initially created for your milestone into this new web page.

- Add 5 overlay objects to the map that relate to specific items of interest or importance. These overlay objects may be *markers*, *polylines*, or *polygons*. Make sure to include descriptive titles for each object.
- Add an *info box* to each object that contains additional detailed information about the object

This work by Karl Benedict is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.