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

Karl Benedict

GEOG 485L/585L - Spring 2018

Introduction

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

- Complete readings and video tutorials from week 1.

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, and
2. A variant of your *home page* that has been styled to modify its appearance.

Step 1 - Open the *home page* template in your web browser.

Step 2 - Copy the code in the home page template into a new file named `index.html` in your GitHub repository.

Step 3 - After you have committed the `index.html` file to your repository preview it in your browser to see what it looks like when accessed as a static web page. Verify that it looks the same as the preview of the template file as a web page.

Step 4 - Answer the questions in the Milestone 2 assignment in Learn

Step 5 - Flesh out the `index.html` page that you created above (*Step 2*) with information specific to you and your interests based upon the content areas in the page. After making sure that your `index.html` is in the same directory as the `hello-world.html` file you created in Milestone 1, add a *relative* link to your `hello-world.html` file to the "milestones" section of your `index.html` page by modifying the line

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

to look like this

```
<p><a href="hello-world.html">Hello World</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 `hello-world.html` page you have properly built your link.

Step 6 - Make a copy of your `index.html` page by copying the content of the page and pasting it into a new document in your repository named `index_styled.html`.

Experiment with some of the styling capabilities described in the W3Schools CSS Examples page (http://www.w3schools.com/css/css_examples.asp) 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.
