

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

Karl Benedict

GEOG 485L/585L - Spring 2018

Overview

- Web Development
- Parts of a web page
- Web Site Components
 - Structure (X/HTML)
 - Presentation (CSS)
 - Behavior (Javascript)
- Simple Web Pages
- More Complete Web Page Example

Web Development

- Requirements
 - Web Server
 - File location that the web server accesses for requested content
 - Files must be readable by all users
- General Process
 - Create basic content in HTML or XHTML (structure)
 - Change appearance of content through the definitions of styles using CSS (presentation)
 - Add dynamic capabilities to content through Javascript (behavior)
 - REPEAT over and over and over and over again

Parts of a Web Page

```
1 <html>
2   <!-- The HTML block is the container for all of your page content -->
3   <head>
4     <!-- The head is where you include pointers to external resources
5        (i.e. style sheets and javascript files), blocks of Javascript code
6        , styles, etc. -->
7     <title>The page title also goes in here</title>
8   </head>
9   <body>
10    <!-- The body is where you put all of the content for the page
11        (i.e. the material that will be displayed in the web browser) -->
```

```

12     <h1>Headers</h1>
13     <div>Generic blocks of content</div>
14     <p>Paragraphs</p>
15     <table>Tables</table>
16     <img ...>Images</img>
17     <form ...>Forms</form>
18     <ul>Unordered Lists</ul>
19     <ol>Ordered Lists</ol>
20     <li>List Items</li>
21
22     <!-- Javascript can go here as well -->
23 </body>
24 </html>

```

[Link to example](#)

Web Site Components - Structure

Content is defined in terms of the structural elements available in HTML/XHTML

- Sample HTML/XHTML Tags
 - Paragraphs (i.e. blocks of text) are contained within `<p>...</p>` tags
 - Headings (i.e. section headings, sub-headings) are contained within numerically defined header tags: `<h1>...</h1>`, `<h2>...</h2>`, `<h3>...</h3>`, etc.
 - Tabular data are within `<table>...</table>` tags
 - List are specified within `...` or `...` tags, depending upon whether the list is ordered (numbered) or unordered (e.g. bulleted)
 - User input elements are put within `<form>...</form>` tags
 - Blocks of content (i.e. sections or divisions) are defined within `<div>...</div>` tags
- Structure is translated into the Document Object Model (DOM) for later use by CSS and Javascript

Web Site Components - Presentation

Modifications to default rendering of HTML/XHTML elements are made through styles defined in CSS

- Styles may be
 - defined in an external file that is referenced within the `<head>` block (the preferred method when doing “real” web development)
 - directly defined within the `<head>` block of a web page
 - directly embedded in the elements to which they apply (generally not a “Good Thing”)
- When not embedded within an element, a style definition consists of
 - A selector
 - The style definition, enclosed in “curly-brackets”, separated by “semi-colons”
 - For example: `h1 {color:red; font-size:18px;}`

CSS Selectors

Selectors may be based on several criteria

- Element name: `h1`, `p`, `table`, `ul`, etc.
 - Element: `<h1>A top level heading</h1>`
 - Selector: `h1 {color:red; font-size:18px}`
- Element ID: a unique name assigned to HTML/XHTML elements within the structure of the document
 - Element: `<p id="para01">Some text goes here</p>`
 - Selector: `#para01 {color:blue; font-size:12px}`
- Class ID: a name assigned to multiple elements which may be modified through reference to their class
 - Element: `<p class="instructions">Here are some instructions</p>`
 - Another Element: `<p class="instructions">Here are some more instructions</p>`
 - Selector: `.instructions {color:red; font-size:12px; text-decoration:blink}`
- Selectors may be combined in a variety of ways

Web Site Components - Behavior

The most interoperable language for adding dynamic behavior to web sites is *Javascript* - supported by most browsers on most operating systems

- A full-fledged programming language
 - A non-trivial undertaking to become proficient in
 - Experience in other programming languages can contribute to learning Javascript
- Defines actions that may be taken on/by DOM elements
- Allows for modification of existing DOM elements, creation of new DOM elements after the page has finished loading from the server, retrieval of new content after page loads
 - An interactive web page that may behave like a local desktop application

Reference Links

- w3schools.com
 - [HTML 5.0 Introduction](#)
 - [HTML 4.0 / XHTML 1.0 Tag Reference](#)
 - [Cascading Style Sheet \(CSS\) selectors and elements](#)
 - [Javascript reference](#)
- World Wide Web Consortium (W3C)
 - [HTML and CSS Background](#)
 - [HTML and CSS Tutorial Links Page](#)
 - [Validators Page](#)
- Webmonkey.com
 - [HTML Cheat Sheet](#)
 - [CSS Guide](#)

Simple Web Page

```
1 <html>
2   <head>
3     <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
4     <title>This is a simple web page</title>
5   </head>
6   <body>
7     <h1>They don't get any simpler than this!</h1>
8     <p>OK, not much simpler than this.</p>
9     <p>Hello World?</p>
10  </body>
11 </html>
```

[link to example](#)

Simple Web Page with CSS

```
1 <html>
2   <head>
3     <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
4     <title>This is a simple web page - with styling</title>
5     <style type="text/css">
6       h1 {color:blue; font-size:large}
7       p.para {color:#777777; font-size:small}
8       #annoying {color:red; text-decoration:line-through}
9     </style>
10  </head>
11  <body>
12    <h1>They don't get any simpler than this!</h1>
13    <p class="para">OK, not much simpler than this.</p>
14    <p id="annoying" class="para">Hello World?</p>
15  </body>
16 </html>
```

[link to example](#)

Simple Web Page with Javascript

```
1 <html>
2   <head>
3     <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
4     <title>This is a simple web page with Javascript</title>
5     <script type="text/javascript">
6       function genericAlert() {
7         alert("You just did something ...")
8         document.getElementById("clickMe").style.color = "red"
9       }
10    </script>
11  </head>
12  <body>
13    <h1>They don't get any simpler than this!</h1>
```

```
14         <p>OK, not much simpler than this.</p>
15         <p>Hello World?</p>
16         <p id="clickMe" onclick="genericAlert();">What happens when you click me?</p>
17     </body>
18 </html>
```

[link to example](#)

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