Lab 1: Webpages

Due: Tuesday, July 10th, 11:59pm

***NOTE**: This lab is based on a homework assignment originally written by William H Sterner for CMSC 10100 (and updated and edited by a variety of instructors and TAs throughout the years)*

This homework makes use of information found primarily in these Castro & Hyslop Chapters:
Step 1: Chs. 3, 4, 5, & 6. Step 2: Chs. 7, 8, 9,10 & 11.

This a broad reach, but we will target some key components of HTML and design. Just do everything locally and I'll explain how to get it to work online on the GitHub server in the end.

Part A (50 pts.) Step 1: Constructing the semantic markup for your own personal homepage (You may look up designs online for ideas).

Progressive Enhancement

Download the lab1_sources.zip file from piazza into your local repository (lab1 folder).

The folder contains a plain html file with the basic directory structure we are looking for. You are to edit the ***index.html*** file, this will be your homepage. Make sure to include comments about what your markup is doing and to update the revision and markup validation status of your pages. Use these websites for validating files and ensure there are no error messages:

- Html: https://validator.w3.org/
- CSS: http://jigsaw.w3.org/css-validator/

Step 1:
Your assignment in Part A is to create a homepage for a business of your own choice where you are the "owner". You should have a navigation bar, welcoming message, what the page is about, what services you provide, and a section that states that you are the owner, and have a picture of yourself. Provide the semantic markup for the text and image content. That is, make an HTML5 web page with the content without any style or presentation. Simply make sure all the information is there. The precise user interface and presentation will be determined in step 2 via an external CSS stylesheet. The following lists comprise both semantic and presentation components.

You are to make use of at least these elements for the semantic markup:
<header>, <nav>, <ul>,<li>, <main>, <article>, <h1> & <h2>,
<section>,<p>,<img>,<a>,<footer>,<blockquote>, <link>

And the **"id", "alt", "role"** attributes for semantics, and later these for css presentation:
class, padding, margin, auto, float, border, font-style, font-family, display, none.

You will of course need to use other elements, and possibly attributes as well, but that is up to your discretion.

While some of the above are not strictly necessary for one single page and amount of content, they are anticipatory for larger, more complicated web pages and sites. They are also important “social” structures for web crawlers looking to identify content correctly. This helps your website get proper notice.

The semantic markup must reflect the structure of the text and be consistent. Also, include the 'alt' attribute and some descriptive text in the <img> elements.
Again, for positive identification, ALL your HTML pages should have your name and the homework number in the TITLE of the page HEAD section. In addition, the date of your last edit and validation should appear in a comment in the HEAD. Be sure to give citations for any external code or markup sources. Make sure you don’t forget to link your style sheets, and have all your links be relative, you shouldn’t have any external links.

Step 2:
Developing an External CSS Style Sheet for the presentation of the content in Step 1. Check out http://www.csszengarden.com/ for wed design ideas. Working with your semantic HTML page, shape its presentation to include these effects as visible in the model document:

1. A masthead (header/navigation) that presents links to different sections of your webpage in a straight line across the top of the web page. Play with the borders.
2. A footer with the same upper and lower opposing borders as the masthead - reversed.
3. Links between the masthead items and the first, second, and etc. sections. They should work in both directions.
4. Images positioned as noted in the source text and according to the model. Use CSS class identifiers for the various left, right, and center positionings. Float, auto, and margin styles can be of use here.
5. Put your CSS styles into a separate document named: “styles.css,” and link to that document with this element:
   <link rel="stylesheet" href="styles/styles.css" />
6. Create left and right margins for the content.
7. Only your final versions should be put in your lab1 directory. It should contain files for:
   a. A semantically marked up text with no link to a CSS file.
      i. Make a copy of the *index.html* file that doesn’t include the style sheet. Name it *index_plain.html*
   b. The combined semantic and presentation version (index.html).
   c. Your styles.css file.
   d. A folder with the original images you used.

Your homework must work from a web browser.

Part B (50 pts.)

Step 1:

Create a second html file, name it ***profile.html.*** In this page, you will basically create a professional profile of yourself, almost like a resume, think LinkedIn. This page should have the same navigation bar that can take you between your first page and this one. Also add a back button within the body of the page that takes you to the homepage.

The first page, where your picture is, should have a hyperlink that takes you to the profile page, this could be the picture itself or your name.

Step 2:

Similarly to part A, here you will add more styles to stylesheet of the first page and make your page look appealing.

Websites usually have the same theme around different pages, so make sure your styles don't conflict between pages. The homepage should be like the skeleton and other pages can add more flavor to your design.

Getting your webpage on the web

Now that you've made your webpages, how do you put them on the web? Well, GitHub makes it easy for us. Simply make a docs directory in your root CAAP-cs directory. Now put all the files you have for the page in this folder. In GitHub, go to settings and scroll down to GitHub Pages and under source, select only from docs folder and save it. Your page is now online!