CSC 170 Lab 8: CSS for Navigation

Due: Thursday, October 12, 2017

This lab has three parts to it:

1. Add a plain HTML navigation element (NAV) you your webpage
2. Chop your webpage into multiple webpages
3. Style the navigation element in CSS

Note: if you’ve been doing these lab assignments correctly up to this point you should have enough content, setup in just the right way to do this lab successfully. If not, see your Lab TA or the Professor for advice on how to fix it.

# Part 1: Add a plain HTML navigation element you your webpage

## Make a copy of Lab 7 (CSS for Layout)

* Make a copy of all the files in it to a new folder named **lab08**
* In your **start.html** file, Change the title in the <title> tag to "Lab 8 - …"   
  e.g. <title>**Lab 8** - Shakespeare</title>

## Create a second CSS file

* In the head of the HTML file, under the existing <link> to the styles.css, create a new link to another CSS file that you’ll name: **navigation.css**
* In the file system, in the **css** folder, create a new blank plain-text file named **navigation.css**
* Test to make sure your new CSS link is working. Add something like: **body { color: red; }**  
  …save all your files and look at the start.html file in a browser. If you see red text, then the connection works. Go back and delete the *body { color: red; }* code.

## Code a navigation element in HTML

* **NAV element**  
  In your **start.html** file, insert a NAV element between the existing HEADER and ARTICLE as demonstrated in the lecture.
* **Class on the NAV element**  
  Add a class to the NAV element; the class name should reflect what you want to do with it[[1]](#footnote-1)
* **List in the NAV element**  
  Within the NAV element, build a menu *list* using the semantically correct HTML as demonstrated in the lecture
* **Anchor tags in each List Item**  
  In each list item, create an anchor tag. For the HREF of each anchor tag, you can simply use **href="#"** for now. For the text of each anchor tag, use the content from your H2s in the document. (There should be at least four.) For example, (using the professor’s Shakespeare demo) you would have something like this: [[2]](#footnote-2)
  + **About Shakespeare**
  + **Early Years**
  + **Life**
  + **Plays**
* **Edit the HREF attributes in each anchor tag**  
  Change the "#" in each anchor tag to an appropriate file name. (You’ll create the files later.) Base the file names on the four sections in your document. For example, (using the professor’s Shakespeare demo) your HREFs would contain something like this:

**href="about-shakespeare.html"**

**href="early-years.html"**

**href="life.html"**

**href="plays.html"**

Remember the naming rules for all files and folders on the web: all names must be all lowercase and no spaces. Also notice that **there is no “home page”** for this website. That’s okay. We’ll fix that later in the semester.

# Part 2: Fix your layout styles to accommodate the new NAV element

Since you've added a new element to your Grid layout, you'll notice that your page layout has gone to smash. You need to fix it.

* Edit your **styles.css** file to accommodate the new NAV element.  
  If you setup two track columns (as demo'd by the professor) then you simply have to add a sytle for the NAV element with the style: grid-column: 1 / 3; (same has the HEADER and FOOTER).

# Part 3: Chop your webpage into multiple webpages

In this part, you must take your **start.html** file and, in effect, turn it into a small four-page website. NOTE: this part of this assignment is not really *real-worldy*. This is not what you would normally do in a website. This is just what we must do so we can have a meaningful set of files to complete this lab assignment.

* Step 1: Close all files in your editor and go back to your computer’s file system
* Step 2: Make four copies of your **start.html** file. Name them appropriately, based on the four HREF attributes you wrote in your NAV element in Part 1. For example, (using the professor’s Shakespeare demo) you would name the copies of your **start.html** file something like this:

**about-shakespeare.html**

**early-years.html**

**life.html**

**plays.html**

Remember the naming rules for all files and folders on the web: all names must be all lowercase and no spaces.

* Step 3: Edit each of the four HTML files one at a time. Delete content that isn’t appropriate for each file.
  + Leave the **DOCTYPE** and **HEAD** as-is
  + Leave the **HEADER** and **NAV** as-is
  + Inside the **ARTICLE**, and the **ASIDE** leave the content that relates to the current file name, and delete the rest - fudge it as necessary! Move things around as necessary![[3]](#footnote-3)
  + Leave the **FOOTER** as-is

# Part 4: Style the navigation element in CSS

The particulars on how to code the following comes right out of Lecture 11 on navigation. See the presentation slides and demo files.

* Edit your **navigation.css** file (not the styles.css file). By targeting the menu using descendant selectors, add the typical code web developers use to nullify the effects of the unordered list[[4]](#footnote-4)
  + Get rid of the bullets
  + Remove the top and bottom margin
  + Remove the padding on the left
* Add a 1px solid black border to the **list items** to turn them into things that look like buttons[[5]](#footnote-5)
* Differentiate the anchor tags in the menu from the other anchor tags in this document:
  + Make the anchor tags in the menu a different, contrasting color
  + Make the hover effect for the anchor tags in the menu another color that contrasts well with the hovered text color.
* To give the user a bigger area to click, turn the **anchor tags** in the menu from inline (its default) to “block”, and then add some padding: 5px on the top and bottom, and 20px on the left and right.

Generally, when styling a menu system, you can use any of the approved layout techniques: inline-block, table-cell, flex, or grid. For this lab assignment, just use the inline-block technique. It's easiest in this case.

* Set this style on your list items using: display: inline-block;

Check your handiwork. Make sure all the links work and you can *navigate* among all four of your webpages.[[6]](#footnote-6)

# Part 5: Add the "you are here" indicator

As you navigate from page-to-page, the navigation system needs to tell the user what page they're on by differentiating the style of the menu item that refers to the current page.

* Edit the NAV element on each of the four webpages. For each page, on the appropriate **anchor tag** (the **<a…** tag) in the list, add a class (e.g. class="is-current") that indicates that that link represents the current page in the website.
* Edit the **navigation.css** file. Add a style that targets the "is-current" class (or whatever you used) so it differentiates the style of the current menu item wherever it's used.
  + E.g. something like this…  
    a.is-current {  
     color: white;  
     background-color: black;  
    }  
    …but you should do something different - something that works with your design.

# Upload your work

When you are done with your web site, close everything and use an FTP tool to access your account on **urcsc170.org** and upload your files:

* In a web browser (any), go to this address to check your handiwork:   
  **www.urcsc170.org/*accountname*/lab08**  
  (where “*accountname*” is your account name)

# Report your work

Remember: Use the W3C HTML validator (**validator.w3.org**) *on every page!* AND the W3C CSS validator (**http://jigsaw.w3.org/css-validator**) to check your work before you report it!

* In our Blackboard section, in Lab 8, post a link to your webpage to receive credit for this Lab.
* Note: this time you're not linking directly to your **start.html** page! (It's not used in this lab.)  
  **http://www.urcsc170.org/*accountname*/lab08**  
  Just provide a link to the directory. Your Grading TA will figure it out from there.

1. Hint: keep it simple; call the class “menu” or “main-menu” …something like that [↑](#footnote-ref-1)
2. If your H2s had a lot of text in them, you can shorten them here to one or two words [↑](#footnote-ref-2)
3. Again - this is not what you would normally do on a website. In this case it is OKAY to move content around - even if it doesn't make sense anymore. Just try to have enough content on every page and balance the content between the ARTICLE and ASIDE (somewhat) so each page looks about normal. (Subjective.) [↑](#footnote-ref-3)
4. The difficult part here is to target just the right elements in the right way. As shown in the lecture, remember…  
    Use descendant selectors to make sure you’re targeting just the menu, not the other ULs  
    The margin you need to get rid of is on the UL  
    The padding on the left you need to get rid of is on the UL  
    [↑](#footnote-ref-4)
5. For this and the rest of the styling instructions, be sure to use descendant selectors to make sure you target just the items in the menu, not the other ULs [↑](#footnote-ref-5)
6. Note: your **start.html** file is *not* linked. That’s okay. Also, there is no “home page” for this website. That’s okay too. We’ll talk about that and more in the next lecture. [↑](#footnote-ref-6)