

In this assignment, you will be creating a web site, from scratch.

We want to develop this site on a server so create a subdirectory called *lab2* for this lab under your *webpages* directory.

```
$ cd /webpages/your_username  
$ mkdir lab2  
$ cd lab2
```

Your site can be found under your students webpage URL

[http://students.engr.scu.edu/~your\\_username/coen161/lab2/](http://students.engr.scu.edu/~your_username/coen161/lab2/)

[\(Links to an external site.\)](#)

*Note: If you're still having issues with your webpages directory you can type webpage directly into your terminal. It will bring an interactive utility that you can use to troubleshoot your webpage. Start by entering 5 to see the status of your page.*

## Part 1 - Header and Footer (20 pts)

We want to use HTML5 semantic tags as much as possible. Here will use them to define the header and footer of our page.

1. Create a file called *index.html* in your *lab2* directory
2. Add the *doctype*, *html*, *head*, and *body* tags. Don't forget to set the *title* in the *head* of the document.
3. Add a *nav* and *footer tag to your body*.
  - o The *nav* will be used to hold our navbar
  - o In the *footer*, add your name and any other contact email you would like
    - Hint: Grab the address tag from lab 1 and put it in your footer
4. Create a file called *styles.css* in your *lab2* directory
5. Link your CSS file to your HTML file using the [link tag](#)
6. Add the following styles to your CSS file to remove any of the browser's default stylings.

```
html, body {  
    margin: 0;
```

```
padding: 0;  
height: 100%;  
width: 100%;  
}
```

7. Make your footer is as wide as the page. Set the background and text color to any color of your choice.

## Part 2 - The Navbar (20 pts)

In class, we discussed how to create a style a simple navbar. We will use that example as the base for our navbar here.

1. Inside your `nav` tag, create an unordered list of links
2. Apply styles as discussed in class to make the list into a horizontal row of clickable links that change color when you hover over them.
  - o One good trick for hovering over nav links, rather than picking a darker green for the background, you can use the [rgba\(\)](#), [CSS function](#), to set the link to a transparent black when it is hovered over. You should be able to see the background color for the nav behind it.
3. Additionally, we want to display the name of the page to the left of the links. On most sites this is usually a link to the home page. Make sure to style this link so that it matches the rest of the nav elements.

## Part 3 - Adding Color (10 pts)

When designing a site, we usually want to limit the number of different colors on the page. Usually its good enough to pick a primary, a secondary, and an accent color.

This is also a good time to apply a CSS principle that we discussed in class. When applying these colors to our page, it can be tempting to individually select the elements we want to color and change their color properties. However, it's more efficient and effective to create color classes that you can apply directly in the class attribute of an element. When writing these classes, we want to make sure that they take precedence over any other color properties that may have been set. We can use `!important` on our color class properties to make sure they are the most specific colors on the page.

1. Create a few color classes for your primary, secondary, and accent `background-colors`

- These can be named something like `.primary` or more specific to the color like `.blue`
- 2. Create primary, secondary, and accent color classes for the text `color` of your elements
  - To differentiate from background colors, name these something like `.primary-text` or `.blue-text`
- 3. Use these classes in your HTML to change the color of your nav and footer

## Part 4 - Content Layout (20 pts)

In order to test the website, we are going to use some fake content to layout a page. The most popular way to create dummy text is called [lorem ipsum](#). It lets you generate a few paragraphs of dummy text that can help you layout content on a page. Grab a paragraph of [lorem ipsum](#) for this next part. You can even use this [spin-off version](#).

1. Create three divs between your nav and your footer. Give each the class `row`
2. In each row, create two columns with divs. Add a paragraph in each column with your lorem ipsum text.
3. The left column should take up 1/4 of the screen and the right column should take up 3/4 of the screen.

### Sample Webpage

The screenshot shows a web page with a blue header bar containing navigation links: "MY PAGE", "HOME", "PAGE 2", and "CONTACT". Below the header, there are three rows of content. Each row consists of two columns. The left column of each row contains a paragraph of lorem ipsum text. The right column of each row also contains a paragraph of lorem ipsum text. The entire page has a white background and a clean, modern design.

Column 1 (Left)	Column 2 (Right)
<p>... Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi elementum arcu et massa pellentesque fermentum. Nam molestie est in pretium finibus. Morbi vitae odio lectus. Integer viverra elementum velit, sed auctor ante feugiat vel. Proin et tortor purus. Donec tincidunt magna eget nulla convallis elementum. Aliquam malesuada massa non velit accumsan pharetra. Suspendisse et nulla tellus.</p>	<p>... Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi elementum arcu et massa pellentesque fermentum. Nam molestie est in pretium finibus. Morbi vitae odio lectus. Integer viverra elementum velit, sed auctor ante feugiat vel. Proin et tortor purus. Donec tincidunt magna eget nulla convallis elementum. Aliquam malesuada massa non velit accumsan pharetra. Suspendisse et nulla tellus.</p>
<p>... Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi elementum arcu et massa pellentesque fermentum. Nam molestie est in pretium finibus. Morbi vitae odio lectus. Integer viverra elementum velit, sed auctor ante feugiat vel. Proin et tortor purus. Donec tincidunt magna eget nulla convallis elementum. Aliquam malesuada massa non velit accumsan pharetra. Suspendisse et nulla tellus.</p>	<p>... Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi elementum arcu et massa pellentesque fermentum. Nam molestie est in pretium finibus. Morbi vitae odio lectus. Integer viverra elementum velit, sed auctor ante feugiat vel. Proin et tortor purus. Donec tincidunt magna eget nulla convallis elementum. Aliquam malesuada massa non velit accumsan pharetra. Suspendisse et nulla tellus.</p>

*Written by Alberto Diaz  
email - adiazostado@scu.edu*

## Part 5 - Adding another page (20 pts)

1. Your next task is to create another page on this site. It will look almost identical to the *index.html* page but will have different content. Call it *page2.html*
  2. On page two we will keep the same 3 rows but instead, each row has **4 columns** across the screen
  3. Update your nav so that you have one link to the home page (*index.html*) and one link to this second page. **Make sure the nav on both pages links to these pages.** We should be able to Navigate back and forth between index and page2

Page 2 should look something like this:

# Sample Webpage

# Submission

Once you have verified that your page meets all the criteria, upload your two (2) HTML files and your (1) CSS stylesheet that is shared between the two.