IST 263

Lab 07

## This lab covers:

1. Development Lifecycle and Style Document (Not to be confused with CSS)
2. Site Maps
3. Hex Color
4. Pseudo Selectors
5. Backgrounds
6. Box Model
   1. Margins
   2. Padding
   3. Borders

## Sitemap Practice

Sitemaps are part of organizing the information on a website. In this section we will build a site map for a fictional art gallery called “263 Modern Art”. Part of organizing information is researching the problem. We are going to go through an exercise to do that research and then you will draw your site map.

1. Site Map Questions
   1. Research at least 3 museum websites. Write down their URLs here.
   2. What pages do the museum sites you researched have in common?
   3. What pages are different or unique on the three sites?
   4. What are some good things about these sites and what don’t you like?
   5. Create your own museum site map with a home page and at least 8 sub pages. Make sure you have a title for the site map, home page, and second level pages.
   6. Write a short rationale for why you chose those pages. For example, the Everson museum has an Exhibits and Events section where it groups relevant information together. All museum sites are different. Why does your museum have these pages?

## SETUP

Create a folder in your Github repository called lab07. Place a copy of your latest skills, bio and contact pages in the new folder. Do not copy the stylesheet from the last lab.

## Website development lifecycle and style document

### Lifecycle

Part of creating a website is planning. We've covered wireframes, copy documents and collecting media. How do these pieces fit into the process of creating a final website? Examine the following:

**Website Development Lifecycle**



* Define Project – Gather Information, Understand Audience and Competitors, Develop Requirements
* Planning – Create Project Timeline, Develop Budget, Gather Media and Copy
* Design – Wireframe, Sitemap, Style Document, Mockup Site Design
* Develop – Write HTML, CSS, JavaScript, SEO and Accessibility
* Test and Deploy – Test Usability and Functionality, Launch Website
* Maintain – Update Site

### Style Document

We are going to work through one more piece of the Lifecycle today in the lab. The website style document is a place to write down all your plans for colors, fonts, etc. Don't confuse this with CSS stylesheets. This is just a way of formalizing our thoughts on color and style before writing CSS.

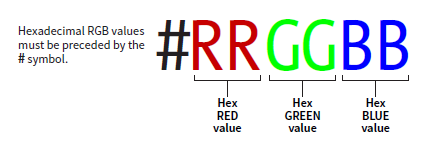
If you’d like to take a look at a real style document, head on over to this link: <https://www.syracuse.edu/about/brand/> to see Syracuse University’s. Click on the Brand Guidelines link to see all the fonts, colors and logos Syracuse University uses.

1. For this part, choose a color scheme for your portfolio website. Do you want a black page with white text and yellow links? A gray page with orange links and black text, etc. Not everyone is great at picking color schemes but there's lots of help on the web. Check out these sites for different color palette inspiration.
   1. <https://coolors.co/palettes/trending>
   2. <https://www.colourlovers.com/palettes>
   3. <https://colorhunt.co/>.
2. You will pick hex colors for the following:
   1. Body background color
   2. Text color
   3. Link color
   4. Heading color
   5. Site Navigation Background
   6. Site Navigation Highlight Color
   7. Site Navigation Link Color

## Hex Color

Hex colors use a 16 digit number system to assign colors and are the most popular way for web developers to specify a color value. Developers specify 6 digits 0-F for the color as follows.

* #000000 = black
* #FFFFFF = white
* #FF0000 = red



1. First, we are going to rename your bio HTML page. Click on it in your Github folder and name it index.html. index is a special filename that indicates the file should load automatically when you get to the directory. After this there will be no need to write the file name to enter your website instead of http://githubusername.github.io/ist263/lab07/bio-scharf.html you will just type http://githubusername.github.io/ist263/lab07/.
2. Create a new blank stylesheet file. You can use the name you used last week and save it in the lab07 folder. Or if you decide to change the name, make sure to update the style tags on all three pages.
3. Use the color palette you created in the style document above to assign the colors for the body, text, links and background. Make sure to use hex colors for this and consult last week’s lab if you have forgotten the selectors and properties.

## Pseudo selectors

### Semantic Tags Preview

I am going to preview the semantic element <nav> (more on semantic elements next week). Semantic elements define parts of the web page. They are useful to developers because they allow us to use CSS on certain sections of the page. They are useful for SEO (search engine positioning) because semantic tags help search engines understand what content means on your page. The nav element is meant to hold the main navigation on your webpage.

1. We are going to add site navigation to your lab HTML pages. Let's start with the bio page which should be named index.html. Add a nav element Just under the body tag. The top of your bio page should look like this.  
   
2. Inside the nav tag we are going to add links for all the pages in the portfolio, so links for:
   1. Home (index.html) – old bio page
   2. Skills
   3. Contact

You might be wondering why we would add a link for the index.html page when we are already on it. It's customary that the links in the navigation include all pages, even the one the user is currently viewing. This means the nav will be uniform on all pages helping users easily locate nav items.  
  
Let's also add a separator between page names appearing in the navigation as well. The pipe character | is a great separator. When you're done, your navigation should work to take you to the other pages and look like this:  
  


1. Add the background color for the nav element to your CSS stylesheet. The selector is nav, the property is background-color and the value is the hex color you choose above.

### Descendant Selector

A descendant selector specifies the element that is the parent of the element and then specifies the element you are styling. The purpose is to only style an element that is inside another. For example, I could only style links that are inside a paragraph element. The syntax is as follows:

Parent\_selector selector {property: value}

1. Using the descendant selector, assign a color to your navigation links by adding a rule to your CSS stylesheet. Add the following to your CSS stylesheet substituting the hex highlight color you picked above for ??????.  
     
   nav a {color: #??????;}

### Pseudo Selector

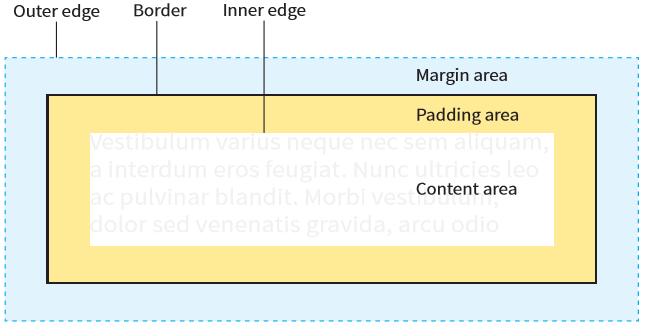
A pseudo selector uses the name of an element combined with the state of the element. The styling applies to an element in a certain state; hover, visited, link, etc. and looks like this:

selector:state {property: value}

1. Add a highlight to the navigation links to show which one the user is rolling over. Because we are doing this only to the links in the nav area (not all the links on the page) we must specify a descendant selector with our pseudo selector. Add the following to your CSS stylesheet substituting a hex value color that you would like to highlight text with for ??????.  
     
   nav a:hover {background-color: #??????;}  
     
   Here's what my navigation looks like currently when the mouse is hovering over Skills:  
     
   

## Box Model

The CSS box model is a box that wraps around every HTML element. It consists of different properties you can style: margins, borders, padding, and the content.



1. Let's continue our path of making the navigation look a little better and add some padding. Everything inside the border will get the background color, so when we change the padding you will see the background color box get larger as well. Add the padding to the nav element so:  
   selector = nav  
   property = padding  
   value = 10px
2. Let's also add a margin around the navigation so the nav has some space before the content starts. This is often called white space and adding it to your web pages is important. It helps to give a visual separation, make a site look more organized and less busy. In the lecture this week, I showed you how to put a margin all the way around an element. Now we are going to change the margin just on the bottom of the element.   
     
   nav {margin-bottom: 30px}
3. Finally, let's add the navigation to all your pages. Copy the content from the beginning of the <nav> tag to the close </nav> tag and paste it to the skills and contact pages.

## What will You hand IN?

Create a word document, pdf or use the "write submission" option in blackboard to provide the following:

1. Submit the answers to the questions about site maps here. (4 pts)
2. Submit the hex colors you choose for Question #3. (1 pt)  
   1. Body background color
   2. Text color
   3. Link color
   4. Heading color
   5. Site Navigation Background
   6. Site Navigation Highlight Color
   7. Site Navigation Link Color
3. Submit the GitHub URL your index.html file (4 pts - Colors displaying on page, margin on nav links, and padding on bottom)
4. Validate the index.html file and submit the validation link. (1 pt)