Module 3 Day 2
Can You … / Do You ... ?

- Utilize the following selectors to apply CSS styles:
    - element
    - id
    - class
    - universal
    - attribute
    - pseudo (hover, focus)
    - descendant combinator
- Understand and utilize selector specificity and cascading to apply styles
- Describe the box model (padding, margin, border)
- Understand and demonstrate the use of `block, inline` and `inline-block`
- Understand the effect of "normal flow" on the positioning of elements on a page
- Describe and use the different positioning styles: `static, fixed, absolute,` and `relative`
- Understand the difference between common units of measure (px, %, rem) and be able to choose an appropriate unit for CSS styles

CSS: Cascading Style Sheets

- CSS allows us to define how an element appears on a website through the use of style rules.

- These rules can control everything from colors and fonts, to positioning and behavior.

- For the purposes of layout and design, every element is thought of as a box, and a style rule is meant to format that box (and its contents).

CSS: Using Stylesheets

Internally:

CSS can be used by declaring a <style>...rules...</style> element within the <head></head>

Externally:

CSS files are linked, related, the a webpage (HTML) using the <link/> element within the <head></head>. Link elements should have three attributes:

href   = "*path_to_the_*stylesheet.css"
rel   = "stylesheet"
type   = "text/css"

CSS: The Box Model

When you look at a website, learn to "see" every element as a box.

This box surrounds both BLOCK level and IN-LINE level elements.

The box has four (4) parts:
>   Margin - Outside Edge
>   Border - Outline
>   Padding - Surrounds Content
>   Content - Inner HTML & Text

Block:
(Start on a new line)
\<div\>
\<h1-6\>
\<p\>

In-line:
(Flow with text)
\<span\>
\<b\> & \<i\>
\<img\>

Margin		
Border		
Padding		
	Content	

CSS: Application of Style Rules

Boxes:

      Background - Color and Image

      Border - Width, Style (Solid, Dashes, Dots), Color

      Position

Text:

      Font (Typeface)

      Color

      Size

      Format (Bold, Italic, Upper\Lower Case)

Special Element-Specific Styling Attributes Exist For:

      Forms

      Lists

      Tables

CSS: Anatomy of a Style Rule

Style rules have four parts:

Selector	**div**
Declaration	**{** border-style: solid; border-width: 2px; **}**
Properties	**{** **border-style**: solid; **border-width**: 2px; }
Values	**{** border-style: **solid**; border-width: **2px**; **}**

---

div **{** border-style: solid; border-width: 2px; **}**

CSS: Selectors - The Basics

- Universal: * {}   Targets all elements on a page
- Element: h1 {}   Targets all *h1* elements on a page
- ID: #userphone {} Targets the element whose ID="userphone"
- Class: .info   Targets any element with a class attribute with a value "info"

                                 div.info   Targets only *div* elements with a class attribute value "info"
- Descendent: div ul {}   Targets any *ul* elements inside a *div* element ***regardless*** of other elements in the *div*

- Child: li>a {}   Targets any *a* element that is the ***direct child*** of an *li* element
- Siblings:
  - Adjacent: h1+p {}   Targets the next *p* element after an *h1* element under the same Parent element

  - General: h1~p {}   Targets all *p* elements that are siblings of an *h1* without regard to its postion

CSS: Selectors - Beyond The Basics

- Attributes
  - Existence: h1[class]   Targets all h1 tags with a class attribute
  - Equality   h1[class="note"]   Targets all h1 tags with a class attribute value equal to "note"
  - Space   h1[class~="note"] Targets all h1 tags with space delimited attributes in which one value
                                                  is equal to "note"
  - Prefix   div[attr^"n"]   Targets all div tags with an attribute value beginning with "n"
  - Substring div[attr*"ot"]   Targets all div tags with attribute values containing the letters "ot"
  - Suffix   div[attr$"e"]   Targets all div tags with an attribute value ending in "e"
- Pseudo
  - :hover   input[type="text"]:hover   Applies when the cursor or pointer is over an element
  - :focus   input[type="text"]:focus   Applies when an element has the focus (is ready for and the
                                                          focus of user interaction)
  - :active   input[type="text"]:active   Applies when a user is interacting with the element