COMP335
Web Application Development

CSS

- Today
  - CSS
- Next classes
  - Responsive Design
  - Bootstrap
  - Assignment#1

What is CSS?

- CSS is a W3C standard for describing the presentation (or appearance) of HTML elements.
- With CSS, we can assign
  - font properties, colors, sizes, borders, background images
  - even the position of elements
- CSS is a language in that it has its own syntax rules.
- CSS has a reputation for being a somewhat frustrating language

Example: P2-CSS/index.html

Style Locations

- CSS style rules can be located in three different locations.
  1. Inline
  2. Embedded: internal
  3. External: separate file
- You can combine all 3!

1. Inline Styles

```
<h1>Share Your Travels</h1>
<h2 style="font-size:24pt">Description<h2>
<h2 style="font-size:24pt;font-weight:bold;">Reviews</h2>
```

- An inline style only affects the element it is defined within and will **override** any other style definitions for the properties used in the inline style.
  - h2 default size: 150% (1.5em) than normal
- Using inline styles is generally **discouraged** since they increase bandwidth and decrease maintainability.

2. Embedded Style Sheet

```
<head>
  <meta charset="utf-8" >
  <title>New York - Central Park</title>
  <style>
      h1 { font-size: 24pt; }
  </style>
</head>
<body>
  <!-- This is a comment -->
  <h1> Share your Travels</h1>
</body>
```

- Since each HTML document has its own `<style>` element, it is more **difficult to consistently style multiple documents** when using embedded styles.

3. External Style Sheet

- This is by far the most common place to locate style rules because it provides the best maintainability.
- When you make a change to an external style sheet, all HTML documents that reference that style sheet will automatically use the updated version.
- The browser is able to cache the external style sheet which can improve the performance of the site

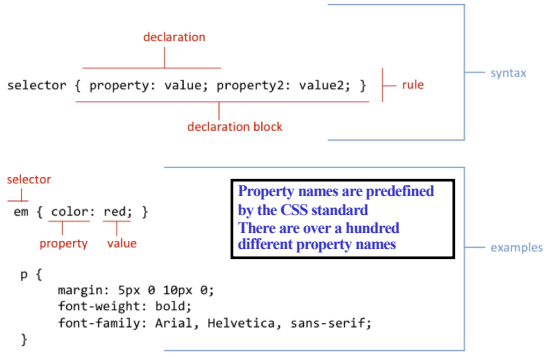
```
<head>
      <link rel="stylesheet" href="css/main.css">
</head>
```

`href` stands for Hypertext Reference

CSS Syntax

- Selectors
  - In CSS, selectors are patterns used to select the element(s) you want to style.
- Element Selectors
- Class Selectors
- ID Selectors

1. Element Selectors



Property names are predefined by the CSS standard
There are over a hundred different property names

```
p {
      margin: 5px 0 10px 0;
      font-weight: bold;
      font-family: Arial, Helvetica, sans-serif;
}
```

https://www.w3schools.com/cssref/default.asp

Grouped Selectors

```
/* commas allow you to group selectors */
p, div, aside {
   margin: 0;
   padding: 0;
}
/* the above single grouped selector is equivalent to the
   following: */
p {
   margin: 0;
   padding: 0;
}
div {
   margin: 0;
   padding: 0;
}
aside {
   margin: 0;
   padding: 0;
}
```

2. Class Selectors

```
<head>
  <meta charset="utf-8">
  <title>CSS Class Selectors</title>
  <style>
  .first{
      font-style:italic;
      color:brown;
  }
  </style>
</head>
<body>
  <h1 class="first">Review</h1>
  <div>
    <p class="first">By Richardo on September 15, 2012</p>
    <p>Easy on the HDR (High Dynamic Range) buddy.</p>
  </div>
</body>
```

A class selector allows you to simultaneously target different HTML elements regardless of their position in the document tree

3. ID Selectors

```
<head>
  <meta charset="utf-8">
  <title>CSS Class Selectors</title>
  <style>
  #first{
      font-style:italic;
      color:brown;
  }
  </style>
</head>
<body>
  <h1 id="first">Review</h1>
  <div>
    <p id="first">By Ricardo on September 15, 2012</p>
    <p>Easy on the HDR (High Dynamic Range) buddy.</p>
  </div>
</body>
```

An id selector allows you to target a specific element by its id attribute regardless of its type or position

you should only be using an id once per page.
So this will have an error

P2-CSS/selectors.html

Id vs. Class Selectors

- Id selectors should only be used when referencing a single HTML element since an id attribute can only be assigned to a single HTML element.
- Class selectors should be used when (potentially) referencing several related elements.

Cascade

- CSS has a system to help the browser determine how to display elements when different style rules conflict.
- The "Cascade" in CSS refers to how conflicting rules are handled.
- CSS uses the following cascade principles to help it deal with conflicts:
  1. inheritance
  2. specificity
  3. location

1. Inheritance

- Many (but not all) CSS properties affect not only themselves but their descendants as well.
  - The `inherit` keyword specifies that a property should inherit its value from its parent element.

```
<style>
  div{
    font-weight: bold;
    margin:50px;
    border: 1pt solid green;
  }
  p{
    border:inherit;
    margin:inherit;
  }
 </style>
```

P2-CSS/inheritance.html

2. Specificity

- Specificity is how the browser determines which style rule takes precedence when more than one style rule could be applied to the same element.
- The more *specific* the selector, the more it takes precedence (i.e., overrides the previous definition).

Element Selectors < Class Selectors < Id Selectors

P2-CSS/specificity.html

```
body{
  font-weight: bold;
  color: red;
}
div{
  font-weight: normal;
  color: magenta;
}
p {
  color: green;
}
.last{
  color: blue;
}
#verylast{
  color: orange;
  font-size: 16pt;
  font-weight: bold;
}
```

```
<body>
  This text is not within a p element

  <div>
    <p>By Richardo on <time>..</time></p>
    <p>Easy on the HDR....</p>
    This text is not within a <strong> p
</strong> element
  </div>

  <div>
    <p class="last"> By Susan on ...</p>
    <p id="verylast">I love Central..</p>
  </div>

</body>
```

This text is not within a p element

By Ricardo on September 15, 2012

Easy on the HDR (High Dynamic Range) buddy.

This text is not within a p element

By Susan on October 1, 2012

I love Central Park.

P2-CSS/specificity.html

3. Location

- When inheritance and specificity cannot determine style precedence, the principle of location will be used.
- The principle of location is that when rules have the same specificity, then the latest are given more weight.
- There is one exception to the principle of location.
  - If a property is marked with **!important** in an author-created style rule, then it will override any other author-created style regardless of its location.

location.html

```
<head>
  <link rel="stylesheet" href="css/styleA.css">
  <link rel="stylesheet" href="css/styleB.css">
  <style>
  #example{
    color: orange; /* color: orange !important*/
    color: magenta;
  }
  </style>
</head>
<body>
<!--  <p ckass="example">-->
  <p id="example" style="color:red;">
    sample test </p>
</body>
```

stylesA.css

```
.example{
  color:green !important;
}
```

stylesB.css

```
.example{
  color:blue;
}
```

Exercise

- Download **P2-CSS.zip** from BB
- Make changes (files under `/css-ex1` folder) to have your page similar to the expected result

Hint:

`background-image`

`url(background.png)`

`background-repeat`

`<img src="hammer.png”>`