COMP335
Web Application Development

CSS

- Assignment #1
- CSS
  - 2/12(F), 2/17 (W)
  - **2/17 Wednesday: Monday schedule**
  - 2/15 Monday: Presidents' day
- JavaScript (4 classes)
- Project prototyping
  - 3/4(F)
- In-class programming exam (tentative)
  - **3/14 (M) Exam**

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: 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>
```

- 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

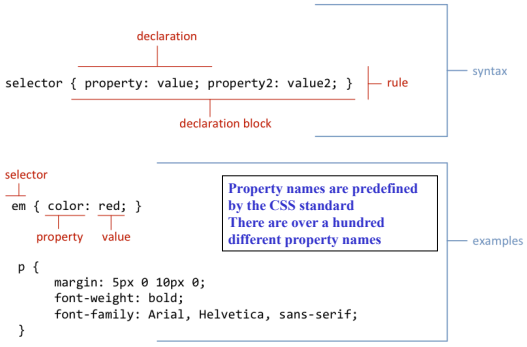
- This is by far the most common place to locate style rules because it provides the best maintainability.
- When you make a change to an external style sheet, all HTML documents that reference that style sheet will automatically use the updated version.
- The browser is able to cache the external style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

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

Selectors

1. Element Selectors
2. Class Selectors
3. ID Selectors

1. Element Selectors



Properties

Property Type	Property
Fonts	font font-family font-size font-style font-weight @font-face
Text	letter-spacing line-height text-align text-decoration text-indent
Color and background	background background-color background-image background-position background-repeat color
Borders	border border-color border-width border-style border-top border-top-color border-top-width etc

Color Values

The unit of any given value is dependent upon the property
Some property values are from a predefined list of keywords

Method	Description	Example
Name	Use one of 17 standard color names. CSS3 has 140 standard names.	color: red; color: hotpink; /* CSS3 only */
RGB	Uses three different numbers between 0 and 255 to describe the Red, Green, and Blue values for the color.	color: rgb(255,0,0); color: rgb(255,105,180);
Hexadecimal	Uses a six-digit hexadecimal number to describe the red, green, and blue value of the color; each of the three RGB values is between 0 and FF (which is 255 in decimal). Notice that the hexadecimal number is preceded by a hash or pound symbol (#).	color: #FF0000; color: #FF69B4;
RGBa	Allows you to add an alpha, or transparency, value. This allows a background color or image to “show through” the color. Transparency is a value between 0.0 (fully transparent) and 1.0 (fully opaque).	color: rgb(255,0,0, 0.5);
HSL	Allows you to specify a color using Hue Saturation and Light values. This is available only in CSS3. HSLA is also available as well.	color: hsl(0,100%,100%); color: hsl(330,59%,100%);

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

Example: 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.
  - Font, color, list, and text properties are inheritable.
  - Layout, sizing, border, background and spacing properties are not.

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

inheritance.html

2. Specificity

Element Selectors < Class Selectors < Id Selectors

specificity.html

- 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).

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

```
<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 div 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 Richardo on September 15, 2012

Easy on the HDR (High Dynamic Range) buddy.

This text is not within a div element

By Susan on October 1, 2012

I love Central Park.

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: magenta;
  }
  </style>
</head>
<body>
<!--  <p id="example">-->
  <p id="example" style="color:red;">
    sample test </p>
</body>
```

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

stylesA.css

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

stylesB.css

In-class Exercise

- Download **css-ex1.zip** from BB
- Make changes to have your page similar to the expected result
- Work together with your team member

Hint:

```
background-image
url(background.png)
background-repeat
repeat-x
<img src="hammer.png”>
```