Cascading Style Sheets

October 3, Unit 4

What are Cascading Style Sheets?

- Abbreviated CSS
- Same principle as design templates in MS PowerPoint
- Allows you to specify the style for your webpage
  - Can change fonts, colors, sizes, etc.
- 1 style sheet can be used for multiple pages
  - Adds consistency to your site
  - Saves a lot of work!

Why were CSS Created?

- Originally HTML was designed only for logical markup
  - <p>, <h1>, etc.
  - The browser was supposed to take care of the layout and look of the page
- But, IE, Netscape, and other browsers started adding tags and attributes so people could specify how the document would also look
  - Physical markup
- CSS allow developers to separate the content of the page from the style

Beginning CSS

- *External* CSS are text files
  - Have the file extension .css
- Create style sheets just like an html document
- Can use any text editor
  - Better with syntax highlighting
- Can also be embedded in your html document using the style attribute
  - These are *Internal Style Sheets*
- Better to create a separate file so you can apply the style sheet to many pages

How CSS Work

- Browsers have a default style for each tag
  - You've overridden some of the defaults (probably) while writing assignment 1
- CSS allows you to override the default style
- Perform the same function as the physical markup in your html documents from assignment 1
- Because you can use an external file, multiple html documents can link to a single css file
  - Saves a lot of tedious writing

Cascading?

- Style sheets can be specified in a number of ways:
  - In an html element (a tag)
  - In the head tag
  - As an external css file
  - Even have multiple css files for a single document
- These multiple definitions *cascade* into a new, virtual style sheet

Cascading Priority

- Multiple styles can be declared for a single element
- Have to have a priority
- From lowest to highest:
  1. Browser default
  2. External Style Sheet
  3. Internal Style Sheet
  4. Inline style

Where Best to Define Style Sheets

- External Style Sheets:
  - Great when there are multiple pages for the same site
  - Allow you to change look of entire site by editing a single file
  - Best for general layout and overall look of all pages
    - Colors
    - Font family
    - Font size
    - Tables, like color, cellspacing, etc.
  - Any of the elements you want to be consistent across the site

Where Best to Define Style Sheets

- Internal Style Sheets:
  - Great if you only have a single page
    - Or pages which are wildly different
  - Also useful if you want most of your page to be the same as others on your site, but change a few elements
    - For example, changing colors
- Inline Style
  - Good for quick markup of a page
  - Should only be used in the published version when you want to change the look of a single item once or twice

Writing Style Sheets

- Style Sheets are made up of rules which change the default style for the browser
- Each rule has three parts:
  - Selector
  - Properties
  - Values
- Looks like:

```
selector{
        property: value;
        property: value;
}
```

Parts of a Rule

- Selector:
  - “selects” what will be modified
  - Simplest selector would be a tag
    - <p>, <h1>, etc.
- Property:
  - What is going to be changed
  - You already know things like color, font-size, etc.
  - Can have any number of properties
- Value:
  - Specifies what the value of the property should be changed to
  - For instance changing the color to “red”
- The property: value parts are very similar to the attribute: “value” specified inline

Simple CSS Rule

- Lets change our default font size and color for paragraphs

```
p {
    color: orange;
    font-size: 150%;
}
```

Changing Heading Defaults

- What if we want all of our headings to be a different color and a different font?

```
h1 {
        color: #FFFF33;
        font-family: Verdana, serif;
}

h2 {
        color: #FFFF33;
        font-family: Verdana, serif;
}

h3 {
        color: #FFFF33;
        font-family: Verdana, serif;
}

etc.
```

Grouping

- Instead of listing all the headings in the Style Sheet we can group them to cut down on how much is written and help with readability

```
h1, h2, h3, h4, h5, h6 {
        color: #FFFF33;
        font-family: Verdana, serif;
    }
```

Grouping, cont.

- Grouping is more efficient than listing each selector separately
- Big help for readability
- Can group any selectors

```
h1, p, {
  text-align: center;
 color: #FF3366;
}
```

- You should use grouping when writing your style sheets

Applying an External Style Sheet

- To use an external style sheet we're going to use the <link> tag
- The <link> tag goes inside of the <head> tag
- Example:

```
<head>
<link rel = "stylesheet" href ="style.css" type = "text/css" />
</head>
```

Parts of the <link> Tag

<link rel = “stylesheet” href =“style.css” type = “text/css” />

- rel: the type of relation
  - In this case “stylesheet”
  - To see the types of relations available check out the <link> tag in the XHTML1.0 reference
- href: the file
  - Just like a hyperlink
  - In this case its style.css and it’s in the same directory
- type: specifies the MIME type
  - Here it is text/css
  - Though servers usually supply the MIME type, sometimes it’s incorrect for css files
  - Specify the MIME type when linking style sheets

Applying an Internal Style Sheet

- If a single page is different from the rest, use an Internal Style Sheet
- Same syntax as an external style sheet
- Insert an internal style sheet using the <style> tag
- Goes inside of the <head> tag

<Style> Tag

```
<head>
   <style type = “text/css”>
   h1{ color: red;}
   p { color: orange; font-family: sans-serif; }
  </style>
</head>
```

Multiple Style Sheets

- Lets say all of our pages use the same style sheet
- But, we want to change a few items on one of the pages
- Can override the external style sheet using an internal style sheet

Multiple Style Sheets, cont

External Style Sheet

```
h1 {
    color: red;
    font-size : 18pt;
   font-family: serif;
}
```

Internal Style Sheet

```
h1 {
    font-size : 20pt;
   font-family: sans-serif;
}
```

Resulting “Virtual” Style Sheet

```
h1 {
   color: red;
   font-size : 20pt;
   font-family: sans-serif;
}
```

- The style sheet “cascades”
- Any properties defined remain
- As you get to higher priority style references, the old values get overwritten

Multiple Sheets Example

External

```
p {
    color: #FF123C;
    text-align: center;
}
```

Internal

```
p {
    color: red;
    font-size: 20pt;
}
```

Inline

```
p {
    color: orange;
    font-size: 18pt;
    font-family: sans-serif;
}
```

Final Sheet

```
p {
    color: orange;
    text-align: center;
    font-size: 18pt;
    font-family: sans-serif;
}
```

Simple Style Sheet

- In Class Example
- We'll go over a bit on paragraphs, headings, and backgrounds