

CIS 3.5, Spring 2010, Lab I.3 / Meyer

1. Introduction:

- 1.1. Cascading Style Sheets or CSS files provide a way to control the look and feel of your web page that is more convenient, more flexible and more comprehensive than adding parameters to XHTML tags (like `<td bgcolor="blue">`).
- 1.2. Style sheets let you define a wide range of settings such as colors, font types, font sizes, boxes around regions on your web page and spacing between various types of elements on the page.
- 1.3. Style sheets are especially useful because you can create one CSS file that defines the look and feel of multiple web pages.

2. Getting started

- 2.1. A CSS file is a separate file from an XHTML file. You create the CSS file in a text editor, just like you created your XHTML file.
- 2.2. First you need to add a tag to your existing XHTML file to tell it that it should use the CSS file. There are several ways to do this. One common way is shown below, using the link tag:

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <link rel="stylesheet" type="text/css" href="meyer-style.css">
  <title>The Greatest Webpage Ever Made</title>
</head>
```

- 2.3. Note that the `<link...>` tag is contained inside the `<head>...</head>` tags.
- 2.4. Add the link tag (as above, but use your name) to your XHTML file.
- 2.5. NOTE: Other ways for including CSS in your documents were covered in the lecture on CSS.

3. Create your CSS file.

- 3.1. Open a new file in your text editor (Notepad on a PC). This new file will be your separate CSS file that will contain the style sheet rules.
- 3.2. Save this file with an extension of .css
 - The full filename should be the value of the href parameter in the link tag you created above.
 - Example: meyer-style.css.

4. The Basics

4.1. Adding a Comment:

- Inside the file, that you just created create a comment at the very top, which identifies that this file belongs to you:

```
/* Mr. Meyer's CSS document */
```

- 4.2. Enter a body rule to set the properties of the body element in the HTML file. The example below sets the background and text colors. Note that these are the same properties that we set earlier, by using the `<body>` tag and its parameters `bgcolor` and `text`. You can set more properties using a style sheet. Note that the property settings in the style sheet take precedence over any set directly in the HTML tags.

```
body {
    background: white;
    color: brown;
}
```

4.3. Note the use of punctuation:

- the open-curly-bracket { at the beginning of the rule,
- the close-curly-bracket } at the end of the rule,
- the colon : after the name of each property being set by the rule, and
- the semi-colon ; after the value of each property.

4.4. Enter the rule, above, using your text editor. Make sure that you specify different colors here than the ones in your XHTML <body> tag, so that you can be sure that the CSS file is being picked up by the browser.

4.5. When you are ready to save the file, make sure it is named with your last name followed by -style.css. For example, my file is called: meyer-style.css Remember, this file name should match the value of the href parameter in the <link> tag that you entered in the step 2.2.

4.6. After you save both files (HTML and CSS), reload the HTML page in your browser. You should see the new colors for the background and text that you specified in your CSS file.

4.7. **NOTE:** There are dozens of other standard tags you might be using in your XHTML file, including: h1, h2, h3, h4, h5, hr, and p. Each of these can be redefined using the technique above.

Example:

```
p {
    color:orange;
    text-align:center;
    font-family:"Times New Roman";
    font-size:20px;
}
```

5. Setting link properties

5.1. You created several links in your XHTML page. It is possible to change the color and behavior of those links. Links recognize different “states” or conditions:

- a:link /* unvisited link */
- a:visited /* visited link */
- a:hover /* mouse over link */
- a:active /* selected link */

5.2. Add the following code to your CSS file. Then save it, and refresh your XHTML file. Within your XHTML file move your mouse over the links to see the effects.

```
a:link, a:visited, a:active {
    color: red;
}
a:hover {
    font-size:150%;
    background: green;
    font-family:monospace;
    color: yellow;
    text-decoration:none;
}
```

6. Dividing up your web page

6.1. One of the most powerful aspects of style sheets is the ability to divide your web page into sections. You can then set the properties within each section, regardless of whether you have lists, tables or whatever inside the section.

6.2. Using sections requires two things:

(1) Specifying the extent of the section in the XHTML file using the `<div class=...NAME...> ... </div>` tags or the ` .. `.

(2) Specifying properties for the section in your CSS file.

6.3. Bring up your XHTML file in the text editor.

- Find the section that looks like this:

*3.This is how a <div> division tag </div> behaves.
*

*4.This is how a span tag behaves.
*

- And Change it so that it looks like this:

*3.This is how a <div class="caution"> division tag </div> behaves.
*

*4.This is how a span tag behaves.
*

6.4. Now go to your CSS file and add the following lines:

```
.caution {  
    font-size:200%;  
    color: Red  
}
```

6.5. Save both of your files and then reload the XHTML file. Now you should see a difference!

6.6. A few more things about divisions:

- You can name the divisions whatever you want, but the names should not contain any spaces. Use the underscore character () if you want to have a multi-word name, e.g., my fun division.
- Notice that in the CSS file, the rule begins with `div.` followed by the name you specified as the value of the class parameter in the `<div>` tag.
- NOTE: You can “nest” divisions, by defining new divisions inside other ones. For example:

```
<div class="main">  
  <div class="part_one">  
    /* this is the first part of the page. */  
  </div>  
  <div class="part_two">  
    /* this is the second part of the page. */  
  </div>  
</div>  
  
/* In this example, the divisions part_one and  
part_two are nested inside the division main. */
```

7. Setting table properties

7.1. You can use a style sheet to set the properties of a table. For example, the syntax for changing the colors of all the table cells and text is:

```
table {
    background: magenta;
    color: pink;
}
```

7.2. Table properties that can be set include:

Name	Description	Possible Values
color	color of text in table cells	any valid color name or hex value
background	background color for table cells	any valid color name or hex value
font-family	font specification for text in table cells	examples: serif, sans-serif, monospace, courier, verdana
font-size	size of font in table cells	relative value (e.g., 200%) or absolute (e.g., 8pt) values

7.3. More great things that can be done with tables can be found here:

http://www.w3schools.com/css/css_table.asp

7.4. Try setting the color properties of your table by adding a table rule to your CSS file. Place it in the file after the body rule. Actually, it does not matter which order rules are placed in a CSS file. The important thing is to keep the rule together, starting with the name of the rule, followed by the open-curly-bracket, the list of property settings and then the close-curly-bracket.

8. Setting list properties

8.1. You can use a style sheet to set the properties of a list. For example, the syntax for changing the background color of the whole unordered list is:

```
ul {
    background: black;
}
```

8.2. Try adding a ul rule to your CSS file, as above. Save the file and reload your XHTML file in the browser. See how it looks.

8.3. You can also set the properties of list items:

```
li {
    font-size: 16pt;
    border: solid;
    border-width: thin;
}
```

9. Validating your CSS and XHTML files.

9.1. Once you have made your changes you will want to make sure that your XHTML file is still “valid”.

<http://validator.w3.org/>

9.2. You can also make sure your CSS file is “valid”:

<http://jigsaw.w3.org/css-validator/>

9.3. Validate both your XHTML file and your CSS file and add the appropriate tags to the bottom of your XHTML page.

Good Job! You’re Done!