# Unit 7 – Cascade Style Sheets

| | |
|---|---|
| Objectives | - Learn how to create embedded style sheets.<br>- Learn how to use tag, class, and id selectors for binding styles to the html source code |
| Required Reading | - Lesson 6 - Formatting Text with HTML and CSS, Lemay, pp. 132-135 and 151-153.<br>- Lesson 9 – Creating Layouts with CSS, Lemay, pp. 270-284, 296-300<br>- Chapter 6 – Introducing Cascading Style Sheets (Sklar, p. 169)<br>- Introduction to CSS: http://www.peachpit.com/articles/article.asp?p=24102&rl=1<br>- Elements of CSS: http://www.peachpit.com/articles/article.asp?p=24110&rl=1<br>- Should you use ID or class? http://css.maxdesign.com.au/selectutorial/advanced_idclass.htm |
| Resources | - CSS Properties Reference: http://www.htmlhelp.com/reference/css/properties.html |
| Assignment | - Create a new webpage and named it as **mini_project4.html**. You need to add the new page more styles using embedded SCC. **Refer to mini project 4 instruction for details.**<br>- **Mini Project 4 will be due on July 17th at 11:45 pm** |

## I. Important Concepts

Review the article of Introduction to CSS (http://www.peachpit.com/articles/article.asp?p=24102&rl=1 ) to understand the purpose of using CSS.

### 1. Inline, Embedded, and External

You have learned how to modify font properties using inline style CSS in unit 3 (refer to pages 132-135 and 151-153 in the Lemay's book). However, you may not experience the power of CSS yet since the way of writing inline CSS is similar to use font tags. There are three ways to write CSS rules: inline, embedded (page-level), and external (sitewide) CSS. In this unit, you will learn how to write embedded CSS.

#### 1.1 Inline

In Unit 3, you have learned how to write CSS using inline style. Basically, you add a style attribute to an html tag. Here is a generic CSS rule for inline style.

<tag style="property: value">…..</tag>

You need to replace <tag> with html tags, such as , , ,…, etc.

You can also combine multiple CSS rules for one html tag by adding a semicolon at the end of the first rule and typing the second rule starting with a property name. For example,

….

#### 1.2 Embedded

On page 270-272 in the Lemay's book, she discusses the page-level (embedded) styles and sitewide style sheets. In the embedded style, you are not writing a style attribute to an html tag anymore, instead you will be adding the style tags within the <head> section in your html and then writing CSS rules within the style tags. Here is a generic CSS rule for embedded styles in <head> in html source.

```
<style type="text/css">
  selector {property1:value1;
            property2:value2; ..}
</style>
```

Below is an example using the embedded CSS.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
 <head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <title>Unit7 CSS Example</title>
  <style type="text/css">
1  body {background-color:#FFFFCC}
   h1 {text-align:center}
2  .subtitle {font-family:Arial, Helvetica, sans-serif; font-size:18px}
3  #footer {text-align:center; font-style:italic; background-color:#669966; color:#FFFFFF}
   a:link {color:#FF3300}
   a:active {color:#CC6600; text-decoration:none}
   a:visited {color:#339966}
   a:hover {color:#CC6600; text-decoration:none}
  </style>
 </head>
 <body>
  <h1>Unit7 CSS Example</h1>
  <p class="subtitle">Inline, Embedded, and External CSS</p>
  <p>I am dummy text. I am dummy text. I am dummy text. I am dummy text. I am dummy text.</p>
  <p class="subtitle">Contextual, Class, and ID Selectors</p>
  <p>I am dummy text. I am dummy text. I am dummy text. I am dummy text. I am dummy text.</p>
  <p class="subtitle">Go to <a href="http://sakai.missouri.edu/portal" target="_blank">Sakai</a></p>
  <p id="footer">Contact Information: 573-882-1653 Digital Media Zone</p>
 </body>
</html>
```

4

In the above example, 5 selectors are used, including body, h1, .subtitle, #footer, as well as a:link, a:active, a:visited, and a:hover (the selectors for different states of a link **are counted as one in this class**).

1. **body** and h1 are Tag selectors
2. **.subtitle** is a Class selector. You prepend . to the class name. You can give a class any name you want but it's better to give a descriptive name than using .style1, .style2…and so on.
3. **#footer** is an ID selector. You prepend # to the ID name. You can give an ID any name you want but it's better to give a descriptive name than using #style1, #style2…and so on.
4. **a:link, a:active, a:visited, a:hover** are Pseudo-class selectors for links

In the next section, we will discuss more about selectors, which is used to bind CSS rules to the html source.

### 1.3 External

An external CSS file basically is the content within <style> tags in the head section of a page using the embedded style. In the above example, the external CSS file will look like the following in Notepad or TextWangler. None of the html code goes into external CSS files. The file should be saved with **.css** as the file extension.

```
body {background-color:#FFFFCC}
h1 {text-align:center}
.subtitle {font-family:Arial, Helvetica, sans-serif; font-size:18px}
#footer {text-align:center; font-style:italic; background-color:#669966; color:#FFFFFF}
a:link {color:#FF3300}
a:active {color:#CC6600; text-decoration:none}
a:visited {color:#339966}a:hover {color:#CC6600; text-decoration:none}
```

In order to associate the external CSS with the html page, a link tag needs to be specified in <head> to refer to the .css file you just created.

<link rel="stylesheet" href="**filename.css**" type="text/css">

The .css file needs to be uploaded to server as well. We will not go into details about external CSS in this class. More advanced CSS will be introduced in the intermediate web development class.

## 2. Selectors (Lemay, p. 272-274)

Another key concept covered in this unit is selectors, which are used to bind the CSS rules to the html source code so browsers would know what CSS rules should apply to what tags. The common use ones are Tag selector, Class selector, ID selector, and Pseudy-classes (styling links). Review the article of Elements of CSS (http://www.peachpit.com/articles/article.asp?p=24110&rl=1) to know the methods of binding styles with HTML elements. Make sure that you understand the differences between them and the conditions of using each of them. Here is an article providing a couple of rules of using ID or class, Should you use ID or class? (http://css.maxdesign.com.au/selectutorial/advanced_idclass.htm)

### 2.1 Tag

Any html tag can serve as a CSS selector, and the CSS rules for that selector will be applied to all instances of the tag on the page. You may refer to the body and h1 selectors in the above example on page 2.

### 2.2 Class

The class attribute is for assigning identifiers to groups of tags. The class rules can be applied multiple times to multiple html tags by adding a class attribute to each tag and assigning the selector name as the value. In this way, you only need to write the rules once but apply it to multiple places. You may refer to the subtitle selector in the above example on page 2. You prep end " **.**" to the class name when writing the rules.

If you have the html source code like this,

<p **class**="**subtitile**">I am dummy text</p>

then you write the CSS rule like this,

**.subtitle** {font-family:Arial, Helvetica, sans-serif; font-size:18px}

### 2.3 ID

The id attribute is for assigning identifiers to specific elements. Unlike class selectors, IDs are supposed to be unique and can only be used once in one html page. You may refer to the footer selector in the above example on page 2. You prepend # to the id name when writing the rules.

If you have the html source code like this,

<p **id**="**footer**">Contact Information: 573-882-1653 Digital Media Zone</p>

then you write the CSS rule like this,

**#footer** {text-align:center; font-style:italic; background-color:#669966; color:#FFFFFF}**2.4 Pseudo-class selector**

The most commonly used Pseudo-class selector is for styling links. The selector for links is a bit different than other tags since links can have multiple states:

- an unvisited link
- a link that user currently is clicking on it
- a visited link
- a link that user currently has the mouse over

Here is an example for styling links using pseudo-class selector:

a:link {color:#FF3300}

a:active {color:#CC6600; text-decoration:none}

a:visited {color:#339966}

a:hover {color:#CC6600; text-decoration:none}

One more thing you need to pay attention to is that **CSS is case-sensitive**, so if you write class or ID selectors in a given case, you must use that case in the HTML file. For example, a class of .leftNav must be added to the document as class="topNav". Otherwise, it will NOT work. However, we know that XHTML is case-specific (lowercase) in terms of elements and attribute names. If you wish to comply with the XHTML specifications, any element selectors in the XHTML files must be in lowercase.

## 3. More about CSS Rules

### 3.1 Units of Measure

On page 275 in the Lemay's book, units of measure are discussed. CSS is picky about using space. Please be sure to follow the generic form of CSS rules. No space is allowed between the value and the units of measure. For example,

p {font-size:16px} and

p {font-size:16 px}

are different. The first one works but not the second one since no space is allowed between the value and the units of measure.

### 3.2 Box Properties

On pages 276-284, several box properties are introduced, including controlling size, borders, padding and margin.

### 3.3 Modifying the appearance of tables

On page 296-298, Lemay discusses how to use CSS to modify the appearance of tables. You may refer to the example on page 297 where Lemay sets the default styles for td and th using tag selectors, and she also uses a left class selector multiple times to the cells that need to be left aligned.

### 3.4 Override

Different types of CSS can be used together. They are not mutual exclusive. You can have a page-level CSS rule to style most of text on the page between <style> in <head> and write an inline style CSS rule to modify the appearance of certain text. For example, if you set the text color to black (#000000) for all the p tags using embedded CSS and you want to change one sentence to be red, you may have a rule between <style> in <head>, like

**p {color: #000000}**

and in the html code between <body>, you will use the **span tag** to do override locally,

I am dummy text. I am dummy text.

I am dummy text. **I want to be red.</style>** I am dummy text.

I am dummy text. I am dummy text.

## 4. Useful Tools

Here is a **CSS Properties Reference web site** (http://www.htmlhelp.com/reference/css/properties.html) that you may refer to while working on CSS. It provides a list of CSS properties, the values that a property can have, and the elements that a property can apply to (inline elements or block-level elements or both). Remember that in unit 3 we have learned about inline and block-level elements? Some CSS properties only have effect on inline or block-level elements. Therefore, it is important to organize your page in boxes with the correct hierarchy. For example, it will not work if you have a pair of span tags outside a pair of div tags and assign properties to the span tags.

Similar to HTML validation, you can check your page using the CSS validation services at http://jigsaw.w3.org/css-validator/. If you have any questions in interpreting the error messages you get from the CSS validator, post it to the "**Mini Project 4**" discussion board.