CSS: Cascading Style Sheets

The Web, With Style!

Introduction

In this module you will learn about CSS—cascading style sheets—the web standard for specifying style information for web pages. You will learn:

- how style sheets support good development habits
- how to link style sheets to your web pages
- basic CSS syntax
- a working set of style properties for dressing up your web pages

Why Style Sheets?

Take a look at any modern web site; if you are paying attention, you are bound to notice that professional sites have a consistent look and feel. Each page of the site will use color, fonts, and content placement in a consistent, organized way.

In the early days of web development, style information was included in the web pages themselves, directly attached to the content the style information modified:

```
<p><font face="cursive,serif" color="#ff9900" size="4">The HTML font
tag is now deprecated.</font></p>
```

While this might seem like a good idea, it fails badly in practice:

- it is very difficult to be consistent, especially for sites larger than a handful of pages
- making changes is difficult, tedious, and error-prone

These problems stem from a violation of two key principles of software development: separation of concerns and DRY: don't repeat yourself. *Separation of concerns* means keeping separate things separated: the content of a web page—defined in the XHTML—should be separated from the style of the page. The DRY

principle states that good software developers don't repeat things unnecessarily: if I want all the headings from each webpage to be highlighted in blue, I should say so once, not each time I create a heading. These principles lead to cleaner sites that are easier to maintain.

CSS is the web standard that supports these principles. XHTML and CSS work together: in your web site, XHTML defines the content of the site while CSS defines the style, i.e. what the content should look like. Since style sheets can be associated with many web pages, they provide a simple way to create a consistent look and feel for a site. Changing styles is also very easy with CSS: update the stylesheet, and every page that uses it will update as well.

Exercise 1: CSS In Action

Let's see a concrete example of XHTML and CSS working together:

1. Open the w3schools CSS demo: http://www.w3schools.com/css/demo_default.htm

   When you open the page, you will see a simple XHTML document with no style information. It will look much like the web pages you created in the XHTML module.

2. Click the link in the upper left to view the first style. The display of the page should change: you will see the same content but with different fonts, colors, list styles, and even a different background color.

3. Try changing between the three style sheets and the style-less option to see how the look of a page can be adjusted without touching the XHTML content.

4. Take a peek at the style sheets themselves to see the CSS responsible for changing how the content is displayed

This exercise gives a brief look at the basic formatting and style features of CSS. For a glimpse at the full expressive power of CSS, take a look at the CSS Zen Garden, which showcases some very sophisticated web designs implemented completely using CSS:

http://www.csszengarden.com/

Connecting A Stylesheet

Style sheets, like XHTML documents, are plain text files with a specific format. Since the stylesheet is a separate text file, pages that are supposed to use it should link to it in their header section. For example, this XHTML header specifies a document title and a link to an external stylesheet file called `style.css`:

```
<head>
  <title>My first CSS page</title>
  <link rel="stylesheet" type="text/css" href="style.css" />
</head>
```

Exercise 2: Styled XHTML

1. In your assignments folder for this class, create a new folder called `css_basics`
2. Create a new XHTML document using the template as your starting point. Insert some headings, some paragraphs, and a list or two. Don't forget to add a title
3. Save your document and load it in the browser to be sure it works
4. Create a new text file in the same folder called `style.css` and insert the following content:

```
body {
    background-color: khaki;
}
```

   Don't forget to save
5. Back in the XHTML file, insert the CSS link shown above into the header of your document. You may want to use copy-and-paste to make sure you get it right. Again, don't forget to save
6. Go back to your browser and reload the page. If you typed everything correctly, the background color of the entire document should change to a light khaki color. If it doesn't, carefully double-check both the CSS and XHTML files to be sure that you typed everything in correctly
7. Try out some other background colors: http://www.w3schools.com/css/css_colornames.asp

CSS Syntax

CSS files have a simple syntax made of three parts: selectors, properties, and values:

```
selector {
    property: value;
    another-property: other-value;
}
```

The *selector* indicates what part of the XHTML document the style information should be applied to. *Properties* are the style elements you can adjust like color, font, and alignment. Each property takes one or more *values* that set the new style.

CSS selectors can be any of the XHTML tag names in your web page, like `p` or `h2`. For example, to set the text color of second level headings to green, you could insert the following into your stylesheet:

```
h2 {
    color: green;
}
```

Neat And Tidy CSS

Just as you did for XHTML documents, it is important that you make the effort to keep your CSS files neat and organized. For example, to insert the previous style block into your stylesheet:

1. Type in the selector and the open curly brace:

   ```
   h2 {
   ```

2. Press `Enter` twice, then type the closing curly brace:

   ```
   h2 {

   }
   ```

   Be sure that the closing brace lines up with the selector so you know where the style block ends.

3. Move up to the line between the braces and press `Tab`. Type in your property and value followed by a semicolon:

   ```
   h2 {
       color: green;
   }
   ```

4. For each additional property you want to add, move to the end of the previous line, press `Enter`, and type the property on the new line:

   ```
   h2 {
       color: green;
       text-align: center;
   }
   ```

   Be sure that the new property lines up with the previous ones.

Exercise 3: Working With Selectors And Properties

1. Edit the stylesheet from the previous exercise, adding selectors for the headings and paragraphs in your document. Set the text color for each of these elements following the examples above.

   Remember to work incrementally:

   a. Add the selector for one element, say paragraphs.
   b. Set the color property for the element. Save your work.
   c. Load or refresh the webpage in the browser. Make sure your changes work correctly.
   d. Repeat the steps for each change you want to make.

2. Let's simulate the highlighted paragraphs from the XHTML assignment:

```
This is a heading

This is a paragraph.  Web development is fun!

This is another paragraph, brought to you by
the letters T, E, and X, and the number 2.
```

a. First, insert the heading and paragraphs into your web page. You can copy-and-paste the XHTML from that assignment into the XHTML for this assignment.

b. Simulate the highlighting by setting the `background-color` property to pink for the heading and yellow for the paragraphs. Save your work.

c. Reload the webpage in your browser to see your changes.

Hmmm... well, it highlighted the whole line, which is OK, but we've also highlighted *all* the paragraphs and heading two text, not just the ones we wanted to. Let's see if we can fix things up...

3. Read about class selectors on the CSS syntax page: http://www.w3schools.com/css/css_syntax.asp

4. So we need to add two classes for highlighting some text pink and other text yellow. Let's think of clever names for them... How about `highlight-pink` and `highlight-yellow`? Clever, huh?

5. Edit the XHTML document, adding class attributes for pink and yellow highlighting in the appropriate places. For example, to specify a certain paragraph should be highlighted yellow, you would say:

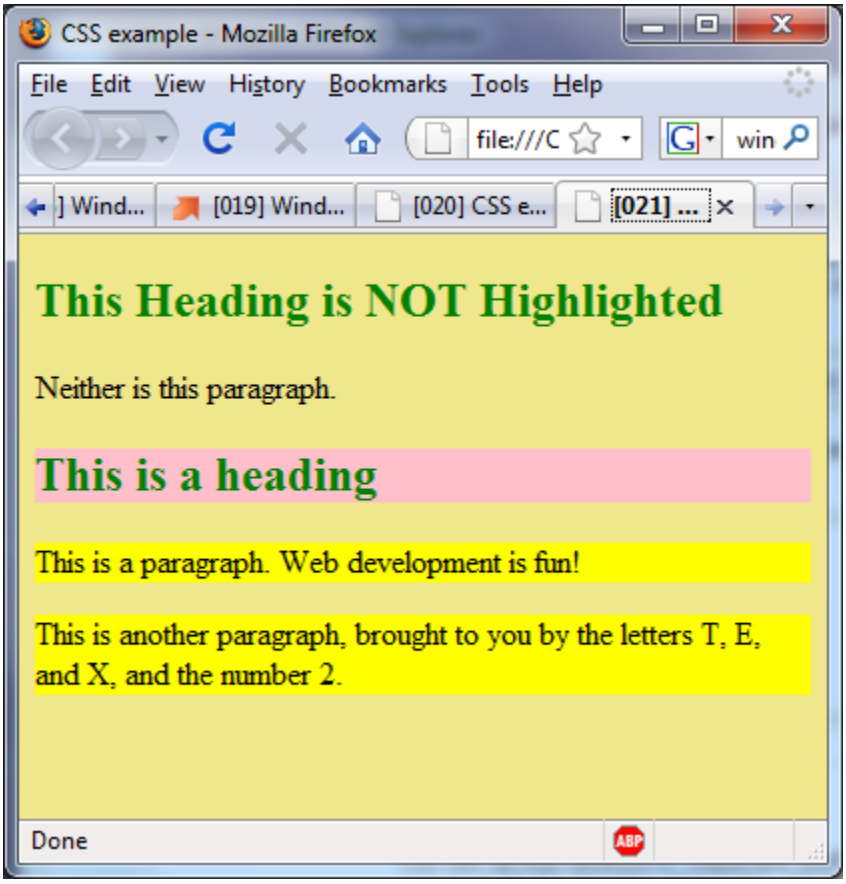
   ```
   <p class="highlight-yellow">This is...</p>
   ```

6. Now edit the CSS file and add the class style descriptions. You can select a certain class of elements in a web page using a selector with the class name preceded by a period:

   ```
   .highlight-yellow {
   ```

   Add the selectors and the appropriate `background-color` properties for both pink and yellow highlighting. (be sure to remove the document-wide highlighting commands you added in step 2)

7. Save your work and reload the document in your web browser. Check to make sure that the highlighting hits only the paragraphs and headings it is supposed to:



Additional CSS Properties

Text And Fonts

1. Read about CSS's text and font style properties
   - http://www.w3schools.com/css/css_text.asp
   - http://www.w3schools.com/css/css_font.asp
2. Try out the alignment, decoration, and indentation examples in the Try It Editor. Adjust the property values and click the button to see the results of your changes.
3. In the font examples, be sure to look at the size, style, variant, and weight examples

Borders, Margins, And Padding

1. First, read about the CSS box model to understand how content is laid out on a webpage: http://www.w3schools.com/css/css_boxmodel.asp

2. Once you understand the box model, you will have a better feel for how borders work in CSS: http://www.w3schools.com/css/css_border.asp
3. Finally, look at the margin and padding sections to get a feel for how to adjust the space around the content of your page
    - http://www.w3schools.com/css/css_margin.asp
    - http://www.w3schools.com/css/css_padding.asp

Summary

In this module you learned how CSS allows you to add style information to your web sites while encouraging clean separation of the style and content of your pages.

Key Points

- CSS selectors choose elements from a page to apply style information to
- Selectors can be XHTML tag names or user-defined classes
- Properties define the style elements that can be changed on a page

Additional Resources

- BrainJar article *Using Style Sheets*, http://brainjar.com/css/using/