



## Lab 07: Making A Web Page Inviting with CSS

**Goal:** In this lab you learn the different ways to add style to your Web page.

“Style” information refers to all of the features that make a Web page look attractive: colors, fonts, positioning, decoration – like blinking – and many others. These can be controlled for a specific tag, or uniformly over the whole page. Style is given by CSS specifications.

**Resources:** The main resource you will need is the CSS (Cascading Style Sheets) home page: [http://www.w3schools.com/css/css\\_reference.asp](http://www.w3schools.com/css/css_reference.asp) This contains all the styling information you need, so just choose from the topic list at left.

The concepts covered in this lab are

1. Structure of style information
2. Specifying style in the tag
3. Specifying style in the <head> section
4. Specifying style in a file

Specific details follow.

**Prepare:** Find your earlier Web page from Lab 06 (or create a similar one quickly). Open it in your text editor, and display it in Firefox.

### ***Structure of Style Information***

Style information has the following form:

```
<style property> : <property value> ;
```

Notice the two punctuation symbols, and that the property is always on the left, and the value is always on the right. For example, the color of text property (**COLOR**) is changed to **red** using any of these forms

```
color:red;  
color:#FF0000;  
color:rgb(255,0,0);
```

Spaces are not needed and usually not given. Some colors have standard names (see CSS Home “ColorNames”). Or, the RGB can be given in hexadecimal (this is most common) or use the `rgb( )` function if you want to use decimal numbers. Other examples of property/style pairs are

```
font-family:helvetica;  
font-size:x-large;  
text-align:center;
```

Check out the properties at the CSS Home.

### ***Specifying Style In A Tag***

The easiest way to add style to your page is within a tag using the style attribute. If you have a paragraph, for example, you can change the color of the text to red by including the style attribute in the start tag:

```
<p style="color:red;">This paragraph is printed in red.</p>
```

Notice that you give the attribute name (style) followed by the equal sign (=) followed by the property/value pair in quotes. In this case, the semicolon isn't really needed, but it's always OK to put it in. Most tags can be styled in this way.

Some times you want to change the style, but there is no handy tag around. For that you use the `<span>` tag as in

```
<p>Paragraph with only 1 <span style="color:red;">red </span> word.</p>
```

which prints as

Paragraph with only 1 **red** word.

Span just encloses content so that it can be styled using the style attribute.

**Part 1.** Improve the style of your Web page by using the style attribute in three different places in the HTML. In addition, make one word of the page blink.

## Specifying Style in the `<head>` Section

The style information we put in the tags with the style attribute can be put in the `<head>` section, where it applies to the whole page. To do this, we need `<style>` tags (yes, tags, not attributes) and we put them after the `<title>` tag in the `<head>` section. For example

```
<html>
  <head><title>Sample Page</title>
  <style>
    p {text-indent:30px;}
    body {background-color:pink; font-family:arial;}
    ul {list-style-type:square;}
  </style>
</head> ...
```

which specifies that the paragraphs should always be styled as indented, the whole page should have a pink background and arial font, and the unordered lists should have square bullets.

Notice the structure of the `<style>` tag information. The tag word or letter is given (e.g. `p`) and the property/value pairs are given after it inside of braces `{ }`, separated by semicolons, as usual.

**Part 2:** Create the Web page at right. You should use `<style>` tags in the `<head>` to set the style for level 2 heads and ordered lists. The heads must be green with an overline. The ordered lists must be preceded by lowercase roman numerals.

Show that the style specification in the head can be "over ridden" in any tag by (a) making exactly one of the list items be red, and (b) making a heading that has no overline. [Hint: `none` is a text decoration choice.]



**Key Idea:** If the same tag (e.g. `p`) is styled both in `<style>` tags, and as a `style` attribute when the tag is used (`<p style= ...>`), **the closer specification applies! Always.**

## **Specifying A Styling File**

If you develop your own Web page style, you'll usually have a lot of property/value pairs specified inside of `<style>` tags in the `<head>`. If you have a lot of Web pages with the same style, these would have to be repeated on each page. Then, when you change your mind ("I now hate those green `<h2>` headings") you have to go back to each `.html` file and change them. Yuck!

The better solution is to have a "style file" with all of those styling specifications in it. Then on each page, say what the file's name is. When you want to change the style, you only change one file and the change shows up everywhere! (You could still have `<style>` tags in the head if you need them for other tags, and you can still put style attributes inside specific locations ... they just 'cascade' down to style the text, **nearest applies**.)

Create a new file, titled `mystyle.css` and copy into it everything that appeared between the `<style>` tags in your `.html` file (but don't include the `<style>` tags themselves!). Next, add a `<link>` tag in your `.html` file in the head section so the browser knows how to find the style file. Place it after the `</title>` closing tag. The link has the form

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

where the attributes are

- `rel` defines the relationship of the linked page; use **stylesheet** as the value
- `type` defines the content of the file, namely CSS text; use `text/css` as the value
- `href` defines how to find the file; use the standard file path naming conventions

**Part 3:** As the final change, move the global styling specifications out of the page from Part 2, and into a file with a `.css` extension. Then put a link in the `<head>` section of the `.html` page and see that it works as before. Finally, add one more styling specification, perhaps to the background, and put it in the style file, too.

## **To Turn In**

Submit the final page and style file of Part 3 to the course drop box.