

# HTML 1, Week 6 – Using CSS

## How CSS Code Works

**CSS format:**

```
selector {  
  property1: value;  
  property2: value;  
}
```

**Key:**

- selector – the tag you want to modify (without angle brackets)
- curly brackets – to enclose all properties and values
- property – the CSS version of the attribute
- colon – substitutes for the HTML attribute equal sign
- value – value of the attribute (without quotes, except when there are spaces in the value, such as a font name like “Times New Roman”)
- semi-colon – substitutes for HTML space that separates attributes

**Example:**

```
h1 {  
  font-family: Arial, "Arial Black", sans-serif ;  
  color: #ccccff;  
}
```

**Note:** Spacing does not matter in the statement of a rule, so you can have any number of spaces surrounding the various parts. Many authors use a lot of space for easier reading.

## Linking to an External Style Sheet

An external style sheet may be linked to an HTML document through HTML's link element which is placed in the document head:

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

The type attribute is used to specify a media type--text/css for a Cascading Style Sheet--allowing browsers to ignore style sheet types that they do not support.

The rel attribute is used to define the relationship between the linked file and the HTML document. rel="stylesheet" specifies a *persistent* or *preferred* style. A persistent style is one that is always applied when style sheets are enabled.

The href attribute specifies the path to the CSS file.

External style sheets should not contain any HTML tags like <head> or <style>. The style sheet should consist merely of style rules or statements. A file consisting solely of the following could be used as an external style sheet.

```
p { margin: 2em; }
```

**(note: measurement cannot be separated from value with a space, it's 2em not 2 em)**

An external style sheet is ideal when the style is applied to numerous pages. With one external style sheet, an author could change the look of an entire site by simply changing one file. As well, most browsers will cache an external style sheet, thus avoiding a delay in page presentation once the style sheet is cached.

## Using an Embedded Style

A style sheet may be embedded in a document with the style element, which is placed in the document head:

```
<style type="text/css">
  body { background-color: black; color: white; }
  p em { color: yellow; }
  .note { margin-left: 5em; margin-right: 5em; }
</style>
```

Note: The type attribute is used to specify a media type, as is its function with the link element.

An embedded style sheet should be used when a single page has a unique style. If the same style sheet is used in multiple pages of a web site, then an external style sheet would be more appropriate.

## Using an Inline Style

Style may be inlined using the style attribute. The attribute takes as its value any number of CSS declarations, where each declaration is separated by a semicolon. An example follows:

```
<p style="color: red; font-family: 'Comic Sans MS', serif"> This paragraph is
styled in red with the Comic Sans MS font, if available.</p>
```

Note: Comic Sans MS is contained within single quotes in the STYLE attribute since double quotes are used to contain the entire style value.

Inline styles are far more inflexible than the other methods. Inline styles lose many of the advantages of style sheets by mixing content with presentation.

## The CLASS Attribute

Classes are used to allow different styling to be applied to different instances of the same element. The following example would allow you to have some paragraphs with normal text and other paragraphs with a different style:

CSS:

```
p.warning { font-weight: bold; color: red; background-color: black; }
```

HTML:

```
<p>This is a paragraph that displays normal looking text.</p>
<p class="warning">This is a paragraph that displays bold red text on a black
background.</p>
```

Classes can also be used to apply the same style to different elements.

CSS:

```
.punk { color: green; background-color: yellow; }
```

HTML:

```
<h1 class="punk">My New Haircut</h1>
<p class="punk">I had my hair colored green one year, and it was great!</p>
```

A good practice is to name classes according to their function rather than their appearance. The warning class in the previous example could have been named red, but this name would become meaningless if the author decided to change the style of the class to a different color.