

## WEB TEXT: Understanding Cascading Style Sheets

Cascading Style Sheets (CSS) are a collection of formatting rules that control the appearance of content in a web page. With CSS styles you have great flexibility and control of the exact page appearance, from precise positioning of layout to specific fonts and styles.

CSS styles let you control many properties that cannot be controlled using HTML alone. For example, you can assign custom list bullets and specify different font sizes and units (pixels, points, and so on). By using CSS styles and setting font sizes in pixels, you can ensure a more consistent treatment of your page layout and appearance in multiple browsers. In addition to text formatting, you can control the format and positioning of a block-level elements in a web page. For example, you can set margins, borders, float text around other text, and so on. The term cascading refers to your ability to apply multiple style sheets to the same web page. For example, you can create one style sheet to apply color and another to apply margins, and apply them both to the same page to create the design you want.

CSS tags can reside in the following locations:

**External (or Linked) CSS style sheets** are collections of CSS styles stored in a separate, external file .css file (not an HTML file). This file is linked to a one or more page pages in a website using a link in the head section of a document. If you change the code of this style type all web pages referencing that style will automatically change (very efficient and powerful!).

**Internal (or Embedded) CSS style sheets** are collections of CSS styles included in a style tag in the head portion of an HTML document.

**Inline styles** are defined with specific instances of tags throughout an HTML document. The disadvantage of using inline CSS is that you have to add them to each tag you want to modify.

**Imported:** Imported styles allow you to use one style sheet that you can override using inline, embedded, and linked styles.

When multiple style sheets are used, the order of importance is as follows:

1. Inline styles
2. Embedded styles
3. External/Linked styles
4. Imported styles
5. Default browser styles

Each style overrides the next style.

### CSS CODE: The Basics

As mentioned before a *style* is a group of attributes called by a single name and a *style sheet* is a group of styles.

A style can be one of two types: a redefined tag or a style class.

A *redefined tag* uses a traditional HTML tag and redefines it. For instance, the following code changes the H1 tag (a header) so that every time you use the H1 tag it will include the attributes you assign to it.

```
h1 { font-family: Arial, sans-serif; font-size: 16px; font-style: italic; font-weight: bold; color: #00FF00; }
```

This code will make the H1 tag use an Arial , 16 point size, italic, bold and green. The tag being defined (h1) is called the *selector* and the attributes between the {curly brackets} are called the *style definition (a.k.a. declaration)*. The declaration has two components, the *property* and its *value*. In the example above, one of the properties is “font-weight” and its value is bold.

When you name and define a new style (and not redefining an old HTML tag) it is called a *style class*. A style class can be applied to blocks or spans of text. In the *heading* portion of your document (above the body tag) you define your style. For example, below I named a style, “mystyle” and then listed its attributes between the curly brackets.

```
<style type="text/css">

.mystyle {font-family: Courier; font-size: 14px;color: #FF0000; text-decoration:
underline; border: medium solid #0000FF;}

</style>
```

Then in the actual body of my document when I want to apply “mystyle” you attach it as an attribute to block of text such as <p>, <div> or <span> tags: For instance the code, <p class="mystyle"> will apply all the properties of my “mystyle” style (as listed in the header of the document-see above) inside the <p> tags.

One way to make using CSS more efficient is to save all your styles in one or several documents separate from your HTML document. When you decide to use a certain style you can just link it to the appropriate CSS document, also known as a *External or Linked Style Sheet*. A style sheet is just a document that lists all your various styles you are using (using the same code syntax shown above) and saved with the extension “.css”. You can even organize your styles into separate documents. If you want a particular web page to reference those styles then simply link the style sheet to that page in the header portion of your web page.

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

The “allmystyles” in the above code is simply the name I chose to call my style sheet but you may call it whatever you choose.

Using a linked style sheet is particularly powerful. If for example, you have a hundred web pages all using a particular style that you would like to change, you don’t need to change the code on each and every one of those 100 web pages. You just simply need to change the code on that one style sheet!

## **Web Fonts**

In CSS3, the @font-face CSS rule now allows us to use fonts that are not necessarily stored in the end-users operating system. In order to do this, you must first inform the web browser where the fonts are that you which to use (with the src property). These fonts can be placed in your root folder of your web site or linked to another server on the web. The fonts should be saved in multiple formats so they can be read by all web browsers. The various font formats available should be shown in your code in the following order; eot, woff, ttf/otf, and svg. Once the @font-face rule establishes the fonts to be used, you can then use those fonts as you would normally with the font-family property. Fontsquirrel.com is an excellent site to download free fonts and includes other tools such as a generator to create additional font-formats.

Sample code:

*Below the font SansumiRegular was placed in the folder myfonts inside the root folder.*

```
@font-face {
  font-family: 'SansumiRegular';
  src: url('myfonts/Sansumi-Bold-webfont.eot') format('embedded-opentype'),
       url('myfonts/Sansumi-Bold-webfont.woff') format('woff'),
       url('myfonts/Sansumi-Bold-webfont.ttf') format('truetype'),
       url('myfonts/Sansumi-Bold-webfont.svg#SansumiRegular') format('svg');
}
```