Today's issue is a quick recap on style sheets that was requested by many of our readers. It presents an overview of style sheets and how they function.

Enjoy!
Gram Gibson, Editor – of DesignNewz.com

The Font Tag vs. the Style Sheet

When hypertext markup language was introduced to the world in 1990 by the W3 group, no one could have possibly known the impact it would have on the world. The language was so simple and direct that it allowed anyone with the desire to learn it to create virtual spaces for information. As the Internet took off, and the browser wars began, more and more functionality was piled into HTML, and by the last release, support for scripting languages, frames, and a host of other things had been included.

One of the major changes in HTML 4.0 is the deprecation of the font tag. What does this mean for you? Well, it means this - kiss your font tag goodbye. Future versions of HTML will not include support for the tag. So how in the world will you format your text? Well, W3 has introduced support for style sheets, which allow you to do all the same things with your text that you could do with the font tag- and more.

Cascading style sheets work on a hierarchical system, and there are three levels. The first is the external style sheet, which exists somewhere on your server, but is not actually a part of your html. Your pages call up this style sheet and apply its specifications to their components.

The next is an internal style sheet, which exists in the head of your document. It performs the same function as the external style sheet, specifying the way different portions of the page will be presented. This style sheet will override corresponding specifications set by the external style sheet. Any specifications of the external style sheet which are not overriden by the internal one remain the same.

The last type of a style sheet is an inline style. This one differs from the other two in that it will determine the properties of only one specifically marked section. It exists in the body of your page and overrides any specifications set by both the external and internal style sheets.
Now let's look at how a style sheet actually functions.

Using the font tag, a small page would appear like this:

```html
<html>
<head></head>
<body>
<p><font face="verdana" size="2">
  <b>This is some bold text.</b>
</font></p>
</body>
</html>
```

Using an external style sheet, we would format this same page as:

```html
<html>
<head>
<link rel="stylesheet" type="text/css"
href="http://www.me.com/mystyle.css">
</head>
<body>
<p class="verdbold">This is some bold text.</p>
</body>
</html>
```

The actual external style sheet would look like this:

```css
p.verdbold {
  font: bold 12px verdana
}
```

The HTML document is calling the external mystyle.css page, which is telling it to format every paragraph of the class "verdbold" using bold verdana text twelve pixels high.

An internal style sheet of the same type would look like this:

```html
<html>
<head>
<style type="text/css">
<![CDATA[

p.verdbold {
  font: bold 12px verdana
}

]]>
</style>
</head>
<body>
```

<p class="verdbold">This is some bold text.</p>
</body>
</html>

We could also use an inline style sheet to format this text:

<html>
<head></head>
<body>
<p style="font: bold 12px verdana">
This is some bold text</p>
</body>
</html>

Inline styles are very rarely used, because they defeat the purpose of having properties defined for a general class of objects.

Style sheets are always used in this hierarchical order. In the instance that an external style sheet defined "p.verdbold" as a bold arial, and an internal one defined the "p.verdbold" class as 10px verdana, any paragraphs of the class "verdbold" would show up as bold 10px verdana.