Introduction to Cascading Style Sheets

Cascading Style Sheets (CSS) are an increasingly common way for website developers to control the look and feel of their web pages. A style sheet allows the website developer to specify the size, font and color of text for an entire website without having to code the information in each page. Style sheets separate the style (or look and feel) of the document from the structure (or the content and ordering of information).

CSS is a powerful way for content developers to preserve the desired fonts, colors, and text sizes of HTML documents. However, students with disabilities can also benefit from sites designed using CSS. Individual users can choose to apply their own style sheets to documents, overriding any other style sheets. For example, a person with low vision could modify every CSS-enabled website to display text in yellow on a black background, if those colors enable her to read the text.

However, be aware that CSS does not solve all accessibility problems. In fact, both the WC3 and Section 508 accessibility guidelines stipulate that, for a page to be accessible, the content must be easily understood with the style sheets turned off (i.e. if the user is utilizing an earlier Web browser that doesn't support style sheets, such as Lynx).

Defining External Styles

In earlier versions of HTML, the only way to display text in a given font style, color and size was to use the tag. For example, the following HTML coding would make the text appear as "Futura Extra Bold" in 24-point size red typeface:

Code	Result
This is some sample text.	This is some sample text.

Additionally, developers once used the <b> and <i> tags to display text as bold or italic. For example:

```
<font face="Arial" size="12">This text is in <b>bold</b> and <i>italic</I>
fonts.</font>
```

However, in HTML 4.0, the <font>, <b> and <i> tags have been deprecated, meaning that they should not be used on newly created/designed web pages. Also, the <font> tag was very difficult to use, because it had to be placed around every formatted piece of text. This made it difficult to change the look and feel of a website, because each <font> tag had to be changed individually.

Instead of these deprecated tags, developers should use CSS to define an "external" style sheet for setting font information across multiple pages, instead of resorting to the deprecated tags.

To build a style sheet, start with a text file with a ".css" extension (for example, "style.css". Styles are defined within the ".css" file by specifying the tag ("p", "h1", etc.) and then placing all of the style preferences within curly-brackets ("{ }").

GRA+DE
Georgia Tech Research on
Accessible Distance Education

factsheet
factsheet
factsheet 10

www.catea.org/grade

■ Introduction to Cascading Style Sheets - *continued*

To build a style sheet, start with a text file with a ".css" extension (for example, "style.css". Styles are defined within the ".css" file by specifying the tag ("p", "h1", etc.) and then placing all of the style preferences within curly-brackets ("{ }").

The following examples demonstrate how to achieve the same look and feel of the above example using correctly formatted style sheet definitions:

Style Sheet	HTML Coding	Result
H1 {font-family:"Arial"; font-size:"150%"; color:"black"} p {font-family:"Comic Sans "; font-size:"100%"; color:"green"}	<h1>This is the title of the page.</h1> <p>This is the content of the paragraphs in the page.</p>	This is the title of the page. This is the content of the paragraphs in the page.

Each style preference is separated using a semicolon (;), so developers can define many different types of styles. The above example establishes the look and feel for all text that appears within the "<p>" tag. Additionally, these examples use relative font sizing rather than fixed sizing; relative sizing is more accessible because it allows the user to dictate the size of the text more easily.

■ Using Colors in CSS

In the example above, the colors are described using words such as "black" and "green". This is not the best way to describe colors in CSS files because the use of "built in" colors like "red", "blue", "black", etc., is deprecated.

The new standard defines each color with respect to its RGB (red, green, and blue) components. As a result, the first value represents the amount of red to use (0-255), the second value represents the green (0-255), and the third value represents the blue (0-255). The numeral 0 represents none of the constituent color, whereas 255 represents 100% of the constituent color.

Thus if we say RGB(255,0,0) we're creating pure red; if we say RGB(0,255,0) we're creating pure green; if we say RGB(0,0,255) we're creating pure blue. We can mix and match values too, so RGB(128,0,128) would be equal half tones of red and blue, creating a light purple color.

■ Resources

This document only summarizes the creation of Cascading Style Sheets. For much more information, visit the following websites:

- World Wide Web Consortium: Cascading Style Sheets: http://www.w3c.org/Style/CSS/

- W3C's CSS Validation Service: http://jigsaw.w3.org/css-validator/

- WebMonkey Stylesheet Guide: http://hotwired.lycos.com/webmonkey/reference/stylesheet_guide/