

Benefits of Cascading Style Sheets

[Web Smart Newsletter](#) originally published August 2005 - Updated July 2006. By Eric Holter.

Benefits of Cascading Style Sheets

This month's newsletter is a short summer follow up to last month's newsletter about the current state of web browsers. It's also short because, for the first time in ten years since starting Newfangled, I've been able to take a week long vacation (two of them, in fact). Thank you, Mark O'Brien! Last month we reviewed the current crop of browsers and discussed some positive movement toward adopting browser standards, including the way Cascading Style Sheets (CSS) are implemented. Cascading Style Sheets have been around for quite a while - if you're a web designer this subject may seem archaic - yet their effective use depends a lot on how all the various browsers support CSS standards, and that's where it gets tricky. This month we'll describe what CSS is and why it's so helpful. We'll also discuss how they can be implemented in the context of an imperfect web browser environment.

Once upon a time in web design land, web pages were coded entirely in HTML. The evolution of web design has introduced new, better ways of creating web pages. HTML was a very inefficient way of telling a web browser how to display a web page because every aspect of how words and pictures were displayed (including position, size, color, and font) had to be embedded in the page using HTML tags. These tags had to surround every piece of content. If you wanted to make a sub-head green you had to insert font tags that told the browser to make the words green. If you've ever viewed the source code of a web page you'll see HTML tags that describe such settings all over the place. These tags had to be repeated every sub-head occurrence. Formatting content in HTML was a lot of work. In fact, it used to be one of the largest line items in our web development proposals.

In addition to being cumbersome, the process was also inflexible. Imagine a site with hundreds of sub-heads all tagged to be green. Now, suppose you wanted to change the sub-heads to blue. Every instance of green tags would then have to be replaced with blue tags to implement the design change.

CSS to the Rescue

Cascading Style Sheets fixed this problem. Rather than embedding tags that tell a browser how to display content, CSS tags are used to tell the browser *what the content is*. So, rather than surrounding a sub-head with green tags, you surround it with sub-head tags. The style of the sub-head is defined separately in a single text file called the CSS file.

	Internet Explorer				Netscape Navigator				Opera			
	Windows		Macintosh		(All Platforms)				(All Platforms)			
	4.0	5.0	5.5	6.0	4.0	5.0	4.x	6.0	7.0	3.5	5.0	7.0
color	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
font-weight	Y	Y	Y	Y	Y	Y	P	Y	Y	Y	Y	Y
font-family	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
font-size	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
font-variant	B	Y	Y	Y	B	Y	N	Y	Y	Y	Y	Y
font-style	Y	Y	Y	Y	Y	Y	P	Y	Y	Y	Y	Y
text-decoration	P	Y	Y	Y	P	Y	P	Y	Y	P	Y	Y
text-transform	Y	Y	Y	Y	Y	Y	B	Y	Y	B	Y	Y
letter-spacing	Y	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	Y
word-spacing	N	N	N	Y	Y	Y	N	Y	Y	Y	Y	Y
line-height	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
vertical-align	P	P	P	P	P	P	P	Y	Y	P	P	P
text-indent	B	B	Y	Y	B	Y	B	Y	Y	Y	Y	Y
text-align	Y	Y	Y	Y	P	Y	Y	Y	Y	Y	Y	Y
direction	N	N	N	N	N	N	N	N	N	N	N	N
unicode-bidi	N	N	N	N	N	N	N	N	N	N	N	N

Y=supported, P=partial, B=buggy, N=no. For details [see here](#).

Whenever the browser sees a CSS tag around some text identifying it as a sub-head, it looks to the CSS file to find out how the text should be displayed. Rather than having to fix every occurrence of sub-head text when a design specification changes, a single change to the CSS file affects every instance of sub-head text across the site. You can imagine how happy that makes a web site developer!

CSS has been around for a long time, but how aggressively it has been utilized has been a factor of both browser compatibility and browser support. If you think about a typical web page, there lots of aspects with regard to display. There is type color, size, font and leading, column widths, and inline images with borders and margins. There are links and roll-overs, navigation bars with drop downs, form widgets, graphics used for bullets, complex table layouts, images that align with text, and images that float in columns next to text. For every one of these elements there are multiple aspects to consider. Text, for example, has size, color and font as you would expect, but also leading, margins, and paragraph attributes. There can easily be hundreds of CSS settings to define the way a web page looks.

Problems with CSS Formatting

Partial CSS Support in Various Browsers

Here's where we run into trouble. For all the hundreds of possible CSS settings there are many that certain browsers do not support, or support them a little differently than others do. Additionally, there are certain unofficial CSS styles that some browsers have invented for themselves without respect to whether or not other browsers have them. Then, there are simply bugs in some browser/platform combinations that do not properly display valid CSS formatting. Check out this chart (you can find this online at www.westciv.com/style_master/academy/browser_support/text.html). It shows the CSS browser compatibility, just for text attributes. Notice the key: Y=supported, P=partial, B=buggy, and N=no.

While CSS is a great way to efficiently code a web page, it's still a tricky business. If you want to get a sense of the power of CSS visit the CSS Zen Garden (www.csszengarden.com). This site demonstrates how the exact same content can be displayed in radically different layouts by simply changing the CSS file. Granted, it takes a lot of effort to establish CSS practices that will display properly in all of your targeted browsers, but once you get things working, design adjustments (whether minor or major) can be accomplished far more easily than if display is hard coded in HTML throughout the website.

CSS for Designers

Making sense of CSS from a print design perspective

For graphic designers who design for the web but do not code their own web pages, how their pages end up displaying can sometimes be frustrating. The nice thing about print is that when you smear ink on paper it generally stays there. But web pages tend to get distorted. We've written a couple of newsletters (Web Design Tips for the Print Designer [part 1](#) and [part 2](#)) giving particular advice to print designers as they make the transition to the web. CSS is something designers should be aware of because any style that's defined via CSS may, in

some cases, or in some browsers, look a little different that it does in Photoshop. This is the cost of doing design on the web. Utilizing CSS as much as possible maximizes both efficiency and flexibility. But for every design, there is a tension between using dependable CSS attributes versus hard coding workarounds. Newfangled exists to help designers contend with these realities and to put our experience to use to find coding solutions that can faithfully represent a designer's layout, yet make the resulting website work well.

We'll always be providing some pressure toward web standards. However, we know that in many cases design trumps coding efficiency - we're cool with that. The art of the process is finding ways of doing both. This sometimes requires creative options, or innovative techniques. Making sure our design partners understand the value of CSS can help us work together more effectively toward solutions that benefit the website and champion the designer's vision.