

## Welcome to Overture!

This is the first issue of Overture, the monthly treatise that examines web trends, techniques, and technologies at depth from a business perspective. Each issue will arm you with information that will enable you to discuss the topic with your site designers and together determine how to most effectively increase your site's contribution to the bottom line.

This paper is also available in HTML format at the following URL:

<http://www.trivectus.com/Overture/volume1/issue1>

---

## The Elements of Style Welcome More Visitors While Lowering Design Costs

*by Robert E. Williams, Jr.*

### Executive Summary

In this issue of Overture, we discuss a web technology that has been around for a few years now, but which is just starting to really catch on: cascading style sheets (CSS). By converting your site to using CSS for layout and formatting, you can drastically lower site maintenance costs and future design costs. Additionally, because a site based on CSS is typically comprised of relatively simple HTML, your site will become more accessible to hundreds of thousands, or even millions, of people, thus avoiding potentially substantial opportunity costs.

---

From its inception by Tim Berners-Lee at CERN in 1989 as a method of distributing high-energy physics papers in electronic format, HTML, the language of the web, has been focused on a document's structure, not its presentation. That is to say, when you mark up a document, you indicate such things as which text is a header, which text is body text, and so forth, and you let the browser take care of the details of drawing everything. By doing this, you enable the browser, which may be running on any platform and may be rendering onto a wide variety of display types and sizes (or no display at all!), to present your document in a way that makes sense for its structure yet is within the browser's capability. Thus, headers will always be recognizable as headers, for example, regardless of whether the document is rendered in a text-based terminal window or on a high-end Mac or Windows workstation with advanced video capabilities.

Unfortunately, as Microsoft and Netscape fought the Great Browser War in the 1990s, they each defined numerous extensions to the HTML standard with the goal of luring webmasters into creating browser-specific web sites in order to lock their competitor out of the market. Aside from being proprietary, the big problem with many of these extensions was that they were presentational by nature, enabling designers to do things like use tables for layout purposes (tables were originally designed only for presenting, well, tabular data). Regardless of this and other shortcomings, the extensions gained widespread use, so much so that the World Wide Web Consortium (W3C), the standards body behind the web, attempted to reel in the number of browser-specific sites by adding many of the extensions to the HTML standards of the era. The W3C accompanied these extensions with admonishments not to use them, but their formal acceptance led to just the opposite. The result is that most web sites came to rely on the extensions, and still do.

Fast-forward to 2003. Nowadays, people can access the web from a wide variety of devices with an even wider variety of rendering capabilities. They may be using a cell phone with a 2 inch (5 cm) black-and-white screen, they may be using a set-top box attached to their low-resolution television set, or they may even be using their refrigerator with a built-in computer. Also, people with a variety of disabilities are using

highly specialized browsers to peruse sites. Some of these browsers lack any display device at all, but instead read a site aloud or convert it to Braille. Even those folks accessing your site with a common, modern browser like Safari, Internet Explorer, or Mozilla, may see your site in very different ways depending on such things as their browser's preference settings, their screen size and window size, and their default text size. Given such a wide variety of capabilities and configurations, you don't want to rely on people recognizing your all-important headlines as headlines by virtue of their being output as 24-point bold, red Helvetica, because in many cases, one or more of those attributes won't get displayed. The solution? Go back to structural markup by specifically telling the browser that your headline is a headline, and letting the browser do its job. (For example, the speech synthesizer might read the headline with a bit more punch to it.)

There is a down side to using purely structural markup, however, and it's a big one: you lose the ability to create the fancy layouts which web surfers have come to expect, since you only specify structure and not appearance. This isn't really a problem, though, since there's a technology designed specifically to fill the void between structural HTML and the fancy web sites that we all expect to see: CSS.

Cascading style sheets, or CSS for short, is a formal W3C standard created specifically to combat the growing problem of presentational HTML. In fact, the newest versions of the HTML standard have deprecated most of the presentational tags added in the 1990s, and designers are now directed to use CSS instead. CSS, which works much like style sheets in such programs as Microsoft Word and QuarkXPress, provides a way to isolate formatting and layout from content by placing the formatting and layout rules into their own document separate from the content they act on.

The first advantage of this is that by having the formatting instructions in a separate file, you avoid the risk of confusing older browsers with complex layout code and causing them to badly mis-render; the worst that will happen in most of these browsers is that your content will display with minimal formatting. The result may not be very attractive, but at least all your content will be displayed and it will be readable. For newer browsers, the use of CSS enables the browser to render the page at the maximal ability of the display device.

For example, say you mark a block of text on page as a side bar, and in the corresponding CSS document, you say to make that side bar 1 inch (2.5 cm) wide and to move it over to the right side of the screen and let the remaining text naturally wrap around it. While a full-featured browser drawing to a large monitor will render the side bar exactly as instructed, a cell phone browser, which must work with a very narrow screen, might simply keep the block in line with the rest of the text (and retain other ornamentation, such as a border). In this case, CSS has let the cell phone's browser simplify a page designed for a much larger screen in a way that it's still readable on the phone's tiny screen. In contrast, the cell phone might render a traditional, table-based layout such that the side bar appears before the entire article and its headline, appears far to one side such that user must constantly scroll back-and-forth, or appears in some other awkward position.

In this way, CSS greatly increases the accessibility of your site to hundreds of thousands of visitors or more, depending on your site's nature and what devices and browsers your potential audience is using. By not tying yourself to a particular browser, configured a particular way and running on a particular platform, you enable visitors to access your site no matter what they're using. Many designers think such constraints are fine on the Internet, but think about it in real-world terms.... If you're a real estate agent, and a potential client walks into your office wanting to buy a \$3 million home, are you going to turn them away because they happen to be driving a British car, or their Japanese car is more than six months old? Of course not! Yet, companies do the equivalent of just that every day by having web sites that are completely dependent on specific, proprietary technologies, and which don't separate presentation from structure. If you want your site to be successful, you need to make it as accessible as you reasonably can, since people can't give you their money if they can't access your site, no matter how eager they may otherwise be.

Along with making your site more accessible, converting it from presentational HTML to HTML with CSS provides several other substantial benefits to both your company and to your site's visitors. (Benefits to the visitor, of course, end up also being benefits to your company because a happy visitor is more likely to give you their money than an unhappy visitor.)

The biggest benefit to the visitor, besides accessibility, is speed. A faster site is always more pleasant to use than a slower one with the same content. There are two reasons that a CSS-formatted web site is faster than a site using presentational HTML:

- 1) The HTML markup for your pages becomes considerably simpler, since the formatting details are in a separate document. This means your pages are smaller, often by 50% or more.
- 2) Since multiple pages on your site can share the same CSS document, browsers can cache that CSS document once and re-use it. This means that your formatting information only needs to be downloaded once, thus increasing navigation speed by a constant amount on every page the visitor loads after the first.

Another benefit to the visitor is that you can make pages the visitor might want to print out automatically adjust their images, layout, and even content to be more appropriate for printing. For instance, you might hide all your navigation graphics, add a footer or header with your site's URL, and switch the text to a different font. All this happens automatically when the visitor prints; you don't need to create a separate page and then make the visitor load it just to have a good printout.

The benefits of a CSS-based site to your company can be staggering. To start, the increased navigation speed that the visitor experiences helps to conquer the well-known impatience that web surfers suffer from, which lessens the chances that they'll give up on you and click over to a competitor's site. A faster site further increases the probability of making the sale by enabling you to present to the visitor more information in the same period of time. Using the above visitor benefits, 50% smaller pages can directly translate to 50% faster download times, which means you have the opportunity to potentially present twice as much information to your visitors per period of time, depending on content. Similarly, putting formatting rules into cacheable CSS files gives you additional slices of time that add up: shaving 15 seconds of download time off each page after the first in a 13 page visit means an extra three minutes of time to convince visitors to give you their money, and three less minutes of the visitor contemplating going to a competitor's site.

Along with loading faster, the simplicity of CSS-based pages means that those pages are also easier to create. In particular, it's not unusual that four or five lines of CSS can accomplish what it takes dozens of lines of presentational HTML to accomplish. If you're paying an average of \$100 per page to create a site containing 25 pages, and using CSS shaves 40% off that cost, you'll save a whopping \$1,500. Further, the fact that your pages will be comprised of simpler and "cleaner" code means they're more resistant to breaking in future browsers, and that when the inevitable problems do arise, they're more easily fixed through small modifications.

Ah, modifications. This is perhaps the area where CSS provides the biggest benefit to your company. Because CSS separates formatting from content, you can make site-wide formatting changes simply by modifying the single CSS file that defines site-wide formatting styles. For instance, if you decide to make all your headlines centered instead of left-justified, you can make this change site-wide by altering one line of CSS. Compare this to presentational HTML, where your designers may need to go in and edit every file individually. This extends to much larger-scale changes, as well. With careful designing, your site can be such that the basic layout of all of its pages can be completely altered just by modifying the one, global CSS file. If you wanted to, your site could even be set up so that, each time it's loaded, it presents a completely different appearance based, for example, on the seasons of the year, or whether you're open for business on a given day.

To see an example of how a site's look can be completely altered just by changing one CSS file, visit the CSS Zen Garden:

<http://www.csszengarden.com/>

The CSS Zen Garden was designed purely as a demonstration of this power of CSS's, and is perhaps the best example out there. When you click on the name of one of the dozens of designs they offer, the CSS file is swapped and the appearance of the site completely changes. It's something you have to see to believe. Of course, you probably don't want to go to that extent for your own site (and you don't need to), but being

able to make large-scale changes very quickly, rather than spending considerable time editing every page individually, will save your company large sums of money when it comes redesign time, especially if your site has more than just a handful of pages. On a very large site, this single benefit of CSS could easily save tens of thousands of dollars or more.

Another major benefit to your company, especially if you have designers doing layout and programmers doing back-end coding, is that it enables clearer separation of these two duties. Once the designers and programmers agree on a set of style names, the programmers can much more easily have their code generate HTML that the designers can style and re-style to their hearts' content just by altering the style sheets. This avoids the back-and-forth that otherwise can be a problem when programmers must implement design details in their code. The result is that your site is created faster and easier, and the designers can make more changes without the help of the programmers.

Sites based around cascading style sheets are clearly the way of the future in web design. CSS lets us return to using HTML the way it was intended, and yet provides for unparalleled design and creativity options. And it does all this while drastically cutting initial design costs and future maintenance costs and increasing revenues through increased visitor attraction, satisfaction, and retention. CSS also offers benefits beyond those discussed here, such as preparing your site for easier integration of future technologies like XML-based web services. The bottom line is that if you're not already planning your site's next revision to be based, at least in part, on CSS, you should be.

---

## About the Author

Bob Williams is the co-founder and President of TriVectus, LC, a web consulting company based in the thriving tech oasis of Phoenix, Arizona, specializing in the development of complex business web sites and web applications. Bob began doing dynamic web development in 1994, and brings to the profession extensive experience in the areas of software engineering, user interface design, systems analysis, and Internet marketing. He can be reached via TriVectus's web site at <http://www.trivectus.com/>.

---

This newsletter may be freely distributed or reprinted without permission, as long as it is so done in its entire, unedited form, including the forward, the body, the About the Author section, and this copyright and distribution section. Permission to distribute without all parts intact or with editing may be granted, but only in writing by TriVectus, LC or its authorized representatives.

All contents copyright ©2003 by TriVectus, LC. All rights reserved worldwide.  
Visit TriVectus at <http://www.trivectus.com/>.