



## Roadmap Series

Issue 1

# CSS: The How and The Why

Prepared by Joseph Lowery

1-28-09



**WEBASSIST**<sup>®</sup>  
Build better websites, faster.

# CSS: The Why and the How

Produced by Joseph Lowery

2009 is a big 10<sup>th</sup> anniversary year. It's been 10 years since Macromedia launched Dreamweaver. WebAssist is entering its 10<sup>th</sup> year of creating world-class level products. And, the push for web standards has been on-going for 10 years. As such, I thought it'd be good to take a step back and remind folks why web standards in general—and CSS in particular—is a good idea, both from a business and artistic point of view. In the spirit of providing a useful roadmap for web designers, this article will be a mix of a broader 10,000 foot view as to why you should create CSS-based sites, and a more practical, step-by-step set of instructions as to how.

CSS-based layouts and proper use of CSS are at the root of web standards. While there are many in the community—including myself and WebAssist—constantly beating the drum for wide-spread CSS use and overall compliance to web standards, you need to understand the benefits so you can decide to follow this design path yourself. Sooner or later, you'll need to convince your boss, colleagues or client why a CSS-based approach is the right way to go for a new or redesigned site. Although it involves technology, it's not really a technological decision—it's a practical, business decision.

There are a great many advantages to the web standards design, but let's look at the top three: portability, speed and maintenance.

## Better portability

One of the primary tenets of web standards is the separation of content from presentation. A key benefit of this separation is the ability to reshape the content to fit the media. You can, for example, take a site that looks great on the computer screen and reconfigure it for hard copy printer output with a CSS style sheet. With the change of just a few style rules, you can hide areas not pertinent to the page, like a footer of links, or expand a column of text to better fit the printer margins. One of our extensions, [Eric Meyer's CSS Sculptor](#), automatically creates separate style sheets for both screen and print. Most importantly, the portability doesn't stop there. As handheld device technology proliferates, through the iPhone and other devices, cell phone browsing is much more of a reality today. Through a separate CSS style sheet, your sites can look their best no matter the media.

## Increased download speed

CSS brings with it a “real dollars and cents” cost benefit. Most developers find that the average page weight of a CSS-based layout is half that of a table-based one. Let's look at a basic code comparison. First, here's a simple header, content area with sidebar and footer layout done with tables:

```
<table width="780" border="0" cellspacing="0" cellpadding="0">
  <tr>
    <td>Header</td>
  </tr>
  <tr>
    <td><table width="100%" border="0" cellspacing="0" cellpadding="0">
      <tr>
        <td width="31%">Sidebar</td>
        <td width="69%">Main Content</td>
      </tr>
    </table></td>
  </tr>
  <tr>
    <td>Footer</td>
  </tr>
</table>
```

The nested table used for the content area and sidebar adds a bunch of code, doesn't it? Now, let's look at the same layout with CSS-based design:

```
<div id="header">Header</div>
<div id="contentWrapper">
  <div id="sidebar">Sidebar</div>
  <div id="content">Content</div>
</div>
```

## Roadmap Series

Issue 1





```
<div id="footer">Footer</div>
```

Not only is the code much more readable, and thus easier to update (a subject for another, upcoming paper), but the markup is substantially less. Compare character count of the two methods and you'll find that the CSS-based approach is 46% smaller. Naturally, the download times for these particular code snippets would be essentially the same, but if you look at completely developed pages, a notable difference is evident. With major sites—or any site experiencing high traffic—there are actual, significant savings for bandwidth costs. Moreover, designers typically find that the faster download times result in an increased number and length of page visits, which again, can result in a notable increase in revenue, especially if you consider banner and search engine ad placement.

### Easier maintenance

As noted above, the code for a CSS-based site is much easier to read and, therefore, update. Faster changes mean less time which translates into more time designers can spend on other projects, another real, bottom-line benefit. When external style sheets are used, site-wide changes can occur almost instantaneously and even, if appropriate, be user-driven. For a wildly entertaining look at just how different the same content can be, look no further than [www.csszengarden.com](http://www.csszengarden.com). Here are two examples of the same content with totally different appearances, all thanks to CSS.



### First steps

Okay, let's say you're totally sold and you're ready to move to CSS-based layouts. What do you do first? Here are three basic steps to help you on your way.

#### Plan for CSS

Whether you create a comp in Photoshop or Fireworks, or sketch it out by hand and build it directly in Dreamweaver or another code editor, it really helps to have CSS layout concepts in mind from the beginning. Take a look at your design and figure out where the major sections are—the header, the content area, and the footer, for example. This will help you decide which graphic elements need to be part of the CSS-styled background and which are standard foreground images. You'll also get a clearer sense of the margin values and padding required to place your elements in the desired position.

These concepts—and more—serve as the core of our interactive training module, CSS Designer Starter Kit. If you're brand-spanking new to CSS design, it'll give you a good leg up for just a little expense. Another approach is to use your "View Source" browser option liberally and see how designs are recreated in CSS.

#### Use simple building blocks

Or, to put it in an aphorism: "Crawl before you walk; walk before you run." Many times, I see designers extend their reach before they're ready and fail to reach their lofty goal. A CSS page layout is, at its most basic, almost literally a series of building blocks. The content "block," also known as a <div> tag, is placed on top of the footer <div> tag and the header

<div> element goes on top of the content one. When it comes time to ramp up the complexity and split the content section into a sidebar and main area, those two "blocks" are side-by-side, in-between the header and footer. In the early stages of climbing the CSS learning curve, you could do worse than playing with your blocks :-).

## Test early and reset

When do newly-minted CSS designers start to pull their hair out? When they've created the perfect design that looks exactly like they want it to on their browser and then they view it on another browser or system. The more sophisticated the design—and the greater amount of time spent on its creation—the more hair in their hands. To avoid such unnatural hair loss, your best bet is to test on the major browsers at every stage of development.

The conventional wisdom is to initially test your web pages in Firefox or Safari, two well-established web standards compliant browsers. Then, after your site meets your expectations, test in a variety of Internet Explorer versions and adjust. In my experience, the "build-it-all-first" approach leads to much frustration and longer development times. I find that if I do a quick check of my initial page design after every major design addition in Internet Explorer, I can more easily make any modifications necessary. The code is, in essence, fresher in my mind and I can more quickly see where changes might be needed.

Another essential step, is to use a reset declaration at the top of your CSS to level the playing field between browsers. All the pages generated by CSS Sculptor, for example, use a subset of a reset declaration developed by CSS expert Eric Meyer, shown here:

```
html, body, div, span, applet, object, iframe, h1, h2, h3, h4, h5, h6 {  
    margin: 0;  
    padding: 0;  
    border: 0;  
    outline: 0;  
    font-size: 100%;  
}
```

Without a reset declaration, content in an opening <h1> tag will be in an obviously different location when viewed in Firefox and Internet Explorer.

## About WebAssist

WebAssist helps you build better websites faster by offering software, solutions, and training needed to succeed on the web. WebAssist.com hosts a thriving community of over 300,000 designers, developers, and business owners. WebAssist's partners include Adobe, Microsoft, and PayPal.



**WebAssist.com Corporation**  
8899 University Center Lane  
San Diego, CA 92122  
USA  
[www.webassist.com](http://www.webassist.com)