



# **The Importance of Standards Compliant XHTML and CSS Development**

**February 1, 2008**



## The problem with web design is...

Designing for the web has always been fraught with obstacles, from rendering inconsistencies to, in some cases, complete lack of feature support. Jumping these hurdles often required thinking out of the box (and instead, thinking in tables), but as time wore on, these caused their own set of problems. Problems that web designers today know all too well.

Enormous walls of markup and styling that lead to quickly ballooning file sizes and, subsequently, increased load times and potentially massive bandwidth usage; increased development times and costs; incorrect use of elements leading to even *more* rendering irregularities between browsers and possible (probable!) forward incompatibility; alienating potential clients with inaccessible sites.

## But there's a solution!

The de facto governing body of web standards, the W3C, predicted these problems, and gave us the tools to fight them: valid XHTML and CSS. The combination of valid XHTML and CSS is a powerful one that gave designers and developers unlimited power in comparison to our previous arsenal of hacked up HTML and table-based layouts.

However, many sites today are still mired with invalid HTML markup and a seeming disdain for CSS. I imagine the biggest sticking point for those that have yet to adapt is cost. Converting an existing site to pure XHTML and CSS can be an initially costly venture, depending on the complexity of the code base. However, once compared to the longer-term savings and benefits, the motivation usually becomes clear.

## The benefits of standardized XHTML & CSS:

One of the biggest checkmarks in the "XHTML/CSS" column is for reduced page size. Why is reduced page size so important? Well, because that's an aspect that not only affects your potential reach (a user on a slow internet connection may not have time to wait for a large page to load), but also has a direct financial impact. Smaller files require less disk space, so you save on storage, but they also require less bandwidth, so you end up saving on the serving side, too. Depending on how bloated the table-and-font-tag page is, you can generally end up reducing file sizes from 25%-50%. In addition to using CSS for layouts, there are more advanced techniques that can completely replace large JavaScript files for, say, rollovers in navigation menus. Bam, you've just decreased your file size even more.

Of more concern to the developer is the time and complexity that goes into maintaining a table-based design. It is much easier and quicker (and thus, less expensive) to find and edit information in a valid XHTML page than to try and track it down through a half-dozen nested tables and extraneous formatting tags, especially if you want to change the existing layout of the page. A CSS based layout can be radically transformed, generally, with only a few small edits to a CSS file, whereas a table-based layout must be completely redone.



Another benefit to adopting standards based design is that you are designing for the future. Sure, you may be able to work a table-based design into looking correct in your browser now, but what happens when the next browser version comes out? Will it still look the same? Well, last year's launch of Internet Explorer 7 answered with an emphatic "no", as thousands of websites designed around IE6 quirks broke under the new rendering engine in IE7. Designing standards compliant sites ensures that, as long as the browser vendors hold up their end of the bargain, your site will work for perpetuity. Anytime a new browser comes out, as long as they advertise that they adhere to the standards, your site continues to work seamlessly.

And what's quickly becoming another benefit to CSS designs is the proliferation of internet-enabled mobile devices. These devices have less memory than a traditional computer and smaller screens. Fortunately, we've already established that CSS designs are much lighter than their older counterparts, and are easier to adapt to smaller screens.

One of the least touted, but probably most powerful, aspects of XHTML comes from the X in its' name- it's extensible. Simply put, if an element does not exist, or an element exists that does not behave the way you wish it to, you can change/define its' behavior at your whim. This is really only of concern to developers, and only a hardcore subset of developers at that, but it is, nonetheless, a remarkable aspect of XHTML that gives developers the power to literally reshape the behavior of their documents.

## Conclusions...

In the end, migrating to XHTML/CSS designs may bring with it an upfront cost, but the savings reaped from smaller files and shortened development cycles, the promise that only minimal work may be needed to adapt to vastly different browsers, and the increased accessibility that comes with semantically correct documents more than makes up for any initial cost concerns.