

The Fall of Tables



**Name of your
College or
University:**
**Name of the
course:**

British Columbia Institute of Technology

MDIA 2216

Date:

January 10th, 2008

INTRODUCTION

In the process of designing and developing captivating websites, over the years web specialists have gone through many different techniques and tricks to build websites in different ways. Each of these ways becoming more efficient as time goes on. At first there was tables – which captured the web developers attention with the ability to structure html pages in just a way they wanted. Then came frames, a way to show the page of websites and move around the site while still having static navigation, headers, and footers. The hype of frames was short lived as people realized book marking the pages was impossible and the layout themselves proved to be impractical (but we won't be discussing frames this time). And then, came the latest of layout strategies. Cascading Style Sheets they're called, or CSS for short. Cascading Style Sheets break through all the problems of tables and frames and they bring the web to an entire new level. So let us continue and see why the much-loved strategy of tables fell and how CSS came to dominate.

MAIN BODY

What can tables be used for? Well, these days tables can be used to display any kind of tabular data, it can align elements in forms, and it can provide entire website layout structures. They are said to be one of the most versatile elements in xhtml [1]. However along with this, tables can cause a lot frustrating problems. The code can get very complex and missing one single character can mess up the entire structure of the table, sending it toppling down like a *Jenga* tower without a critical piece. Another serious problem with tables is that different browsers interpret tables differently. While a table that is created and works perfectly in (as an example) Firefox and Safari, the same table will be crunched up like never before and look nothing like the original design in (as an example) Internet Explorer.

So why were tables so popular if they were a frustrating mess across platforms? The answer is designers “discovered that they could use tables to gain significant control over the presentation of content to users. By carefully laying out tables within tables within tables, they could position quite precisely any design element that could be contained within a table cell. And that encompassed almost everything” [2]. Tables are not intended as layout tools and they are very inefficient. Tables also have long load times. Web browsers are designed to load tables as a single entity, therefore if any entire webpage is made of only tables then the browser will load the page all at once, meaning a long wait time, rather than loading piece by piece as different elements finish loading.

Long loading times are also contributed to by the use of transparent images on the website. Sometimes a designer may want a bit more blank space on one part of a table cell and with tables, “this kind of precision was unachievable” [3]. So in order to create this extra space a designer came up with idea of creating GIF images - an image with no visible content, or

transparent area. Other than slowing down load time, these transparent images also restrict the page to a fixed pixel size and clutters the page with useless (to the user) images. The third reason tables are bad is that they can be a nightmare to maintain, especially if tables are nested within each other multiple times. It can be hard to organize the code in any readable and comprehensible way and if one uses an HTML editor (like Dreamweaver) rather than a text editor (like *TextMate*) then the code is not guaranteed to be properly and cleanly indented. In true web standards the only time it should be okay to use tables would be for their original purpose; to display tabular data.

If tables can only be used for tabular data, then how do we manage a website layout you ask? With the almighty and powerful Cascading Style Sheets, or CSS. The most important word of this acronym is *style*. This is because styles are defined by rules - and these rules are what dictate how the website will look under certain rules that the designer specifies. The cascading part refers to the fact that if styles are placed in an external *sheet* then they are called from that external sheet to the html file and are then displayed in the internet browser. However if styles are placed, or embedded, within the html, in the head area of the document then those styles are taken before those of the external style sheet. By using CSS positioning, websites can be laid out better, more efficiently, and much cleaner than using tables.

CSS is also better for search engine optimization, which is a big thing these days for websites to have as many people get to websites by finding them through search engines such as *Google*. Simply put, "Search engines spiders are actually lethargic. They don't go through the bundles of HTML code to get to the indexed codes. Font tags and tables make HTML code cumbersome, and thus reduce the accuracy of the results. If you use external CSS files to determine the design attributes, the HTML code will be clean and will create better search engine rankings. With some knowledge of CSS you can change

the code without destroying the visual layout.”[4] What makes CSS so ingenious and the code so clean is the fact that CSS separates structure from style. In this way all the content and strict structure of a website can be placed in the html and all the extra style that makes the actual design of the website can be placed in the CSS. In short, this makes the code of both the HTML and the CSS easier to scan, read, and organize.

CONCLUSION

So tables fell from the heavens because of their long load times, they didn't have good cross-platform capabilities, they created messy and hard to read, hard to organize code, and they weren't so great on the search engine optimization either. And then along came CSS, with the ability to keep clean and concise code, shorter loading times, allow better design, and best of all, pump up the search engine optimization. What I'm trying to say is. The fall of tables was necessary. Tables are for tabular data. Period. Finished. Finito. Fin. Goodbye.

BIBLIOGRAPHY

Books:

[1] Schafer, Steven M. *Web Standards*. Wiley Publishing Inc. Indianapolis, Indiana., 2005. P. 91

[2] Shafer, Dan. *HTML Utopia: Designing Without Tables Using CSS*. SitePoint Pty. Ltd. Collingwood, Australia., 2003. P. 4

[3] Shafer, Dan. *HTML Utopia: Designing Without Tables Using CSS*. SitePoint Pty. Ltd. Collingwood, Australia., 2003. P. 7

Articles:

[4] Tuknov, Mikhail, "How to Use CSS for Search Engine Optimization". http://webreference.com/authoring/css_seo/1