Separation of Content & Design Part II: Why Won't This CD Play In My 8-Track?

Ken Westin
Pacific University

Follow this and additional works at: http://commons.pacificu.edu/inter03

Recommended Citation

This Article is brought to you for free and open access by the Interface: The Journal of Education, Community and Values at CommonKnowledge. It has been accepted for inclusion in Volume 3 (2003) by an authorized administrator of CommonKnowledge. For more information, please contact CommonKnowledge@pacificu.edu.
Separation of Content & Design Part II: Why Won't This CD Play In My 8-Track?

Rights
Terms of use for work posted in CommonKnowledge.

This article is available at CommonKnowledge: http://commons.pacificu.edu/inter03/37
Separation of Content & Design Part II:
Why Won’t This CD Play In My 8-Track?

Posted on August 1, 2003 by Editor

By Ken Westin <webmaster@pacificu.edu>

In November we showed you the basics of XHTML and CSS and its advantages over the old HTML soup you had grown to know and love. Well, if you surf the web as much as some of us, you may start to see some very popular sites converting over to XHTML/CSS based designs that follow the web standards we had discussed earlier. A few notable examples include: ESPN.com, Cingulular.com, Fastcompany.com, Wired.com and of course most importantly Pacificu.edu. The most notable advantages of this shift has been decreased file size, increased accessibility, decreased reliance on graphics for site navigation and overall cleaner code that is easier to manage and a site that is easier to alter structurally and aesthetically. In short these sites have transitioned to these standards to save time and money.

The one disadvantage to these sites occurs when someone views the site using an older browser such as Netscape 4 and below. Since these older browsers do not support CSS, or do not support it correctly, they tend to ignore rules, or even at times choke on them and make your pages a big mess. The trick is to design your pages so that they degrade gracefully to these older browsers. This means designing your pages so that newer browsers can take full advantage of CSS stylings and older browsers get a simplified version that they understand. By doing this you make your content accessible to pretty much everyone and the 97% of your audience using browsers made in this millenium get the styled version as you designed regardless of what browser/platform they are using. At this point in the web browser game asking for a a web site to be authored to work in these older browsers is a bit like asking Sony to make CD’s that you can play in your dad’s 8-track. The advantages of a business or institution authoring in XHTML/CSS far outweigh the drawbacks and if done correctly will not leave those with older browsers in the cold.

ESPN chose to leave those using old browsers out in the cold completly, this was decision made by the development and management team early on in the redesign process. I had the privilage of sitting in on a session where the ESPN developers spoke about their redesign and the primary
reason they went this route was due to the added cost of development time, if they chose to support old browsers it would have been a 30% increase in labor costs to support 3% (and dwindling) of their audience so they opted to draw a line in the sand and stopped supporting older browsers all together.

At Pacific University we cannot leave older browsers out in the cold so quickly. Our site content needs to be as accessible as possible in order to serve our diverse audience base. It was with this in mind that we decided to find a way to utilize the cost/time benefits of XHTML/CSS based layouts while at the same time supporting older browsers so that the content can be viewed. For example if you look at Pacific’s admissions section you will see that we are still using tables in our design, which is a necessary evil if we wanted to support older browsers and keep the layout of the content consistent, however the tables that we used were simplified a great deal by relying on CSS to do much of the formatting.

The entire navigation system relays solely on CSS styled text there are no graphics aside from the logo in the header of these pages. By going this route it was easier to dynamically generate the navigation using server side programming and also to keep the page size down. If you view these pages using an older browser you will see the navigation as simple blue links and also a message telling you that the page would look better in a newer browser. You can link external stylesheets by using the <link> tag like this:

```html
<link rel=”stylesheet” href=”mystyles.css” type=”text/css” />
```

By having all of your styles defined in one stylesheet and having all of your pages link to this it makes it pretty easy to change the design of your site. However, if you want your CSS to work in older browsers such as Netscape 4, you are fairly limited to the amount of CSS that you can use. But what if you want to use more advanced CSS, but don’t want to crash these older browsers? Well another way of linking external stylesheets is to use the “@import” statement. The “@import” statement is only understood by newer browsers that support more of the CSS spec. So you may wonder why would you want to use this method when the other method is more broadly supported. Well since these older browsers ignore the “@import” statement you can put all of your more advanced CSS in this sheet and the older browsers will ignore the stylesheet and will not try to apply these styles at all. For example:

```html
<link rel=”stylesheet” href=”netscape4OK.css” type=”text/css” />
```

```html
@import url("complex.css");

First I am linking in the CSS that is Netscape 4.x friendly (usually just font definitions for existing
tags and a few custom classes). So Netscape 4.x understands this statement and applies these styles located in the “netscape4OK.css” style sheet. Next we have some HTML comment tags (for the really old browsers, so they do not try to render this code to the screen). Inside of the comment tags we have the “@import” statement followed by the url of the stylesheet we want to import. Since Netscape 4.x has no idea what the heck this means it just ignores it and does not attempt to apply the styles. However, your larger audience base with standards compliant browsers will have a browser that understands what this means and will import the stylesheet and apply it to the page.

This can actually get much more complex as you try to figure out what styles you should put in what stylesheet. My suggestion is to start with a standards compliant browser such as Internet Explorer 5.5+, Mozilla 1+, Netscape 6+, Opera or Safari, get the pages looking how you would like them then take a look in older browsers and start moving your more complex CSS into the imported stylesheet. I would also suggest picking up some books on CSS if you want to get into more advanced use of CSS, here are a few that I have on my bookshelf that I have found useful:

- **Eric Meyer on CSS: Mastering the Language of Web Design** By Eric Meyer
- **Designing with Web Standards** By Jeffrey Zeldman (published later this month, however this book has been a long awaited continuation of his prior work “Taking Your Talent to the Web”) I would also recommend Zeldman’s web site at Zeldman.com for more information and inspiration.
- **Cascading Style Sheets: Designing for the Web (2nd Edition)** By Hakon Wium Lie, Bert Bos, Robert Cailliau

See Part !

This entry was posted in Uncategorized by Editor. Bookmark the [permalink](http://bcis.pacificu.edu/interface/?p=2780).