Creating a baseline HTML 5 document

Posted At : 14 August 2011 09:11 | Posted By : Shaun McCran
Related Categories: Best practices, Browsers, HTML

Browser standards and cutting edge web design are not great bedfellows. I've heard the arguments from creative designers that you have to write specific browser CSS styles to accommodate the multitude of browsers and their own unique way of rendering CSS.

I disagree. I firmly believe that with a good understanding of the structure of CSS elements and how they interact with each other you can develop a completely cross-browser non JavaScript baseline template. This article explains how I have approached creating a HTML 5 layout file, that works across legacy browsers just as well as the more modern interpreters.

Firstly I'd like to point out that the demo included here has no actual styling attached to it. I am aiming to creating a baseline template that anyone can drop into a directory to kick off a new project. I'm not too bothered about the look and feel at the moment.

Goal 1 - The same output in all supported browsers

To clarify 'supported browsers' I mean almost everything post Internet Explorer 6. I totally agree with the 'kill IE 6' campaign and unless someone makes a serious case as to why I should, I do not support it anymore. I've come around to thinking that it is ok now to point people towards upgrading their browser as they will have a poor experience using legacy browsers.

If you take a look at the Demo here: http://www.mccran.co.uk/examples/html5 in any of your browsers you should see exactly the same thing. It should be identical, I mean all of it. The margins, spacing and fonts etc are all the same.

The key to starting any project, is to have a clean slate. Try and keep your layout and styling separate, and as I have done here I really, really, recommend using a reset Style sheet. In this way all of the elements a browser does not recognise are handled like Div elements.

This is what I'm using as a reset CSS at the moment.

```css
a, abbr, acronym, address, applet, article, aside, audio,
b, blockquote, big, body,
center, canvas, caption, cite, code, command,
datalist, dd, del, details, dfn, dl, div, dt,
em, embed,
fieldset, figcaption, figure, font, footer, form,
h1, h2, h3, h4, h5, h6, header, hgroup, html,
i, iframe, img, ins,
kbd, keygen,
label, legend, li,
meter,
nav,
object, ol, output,
p, pre, progress,
q,
s, samp, section, small, span, source, strike, strong, sub, sup,
table, tbody, tfoot, thead, th, tr, tdvideo, tt,
u, ul,
var {
background:transparent;
border:0 none;
font-size:100%;
margin:0;
padding:0;
border:0;
outline:0;
vertical-align:top;
}

ol,
ul {
list-style:none;
}
blockquote,
q {
quotes:none;
}
table,
table td {
padding:0;
border:none;
border-collapse:collapse;
}
```

Goal 2 - No optional technology

This is a simple principle really. Do not create elements of a site /application that rely on option technology, such as JavaScript. If it is possible for users to block or disable the technology that you are using then do not rely on it for a critical part of your site.

There must be a guaranteed baseline user experience for everyone. By using the Reset CSS file above rather than a JavaScript HTML 5 workaround you are ensuring this.

In this way all Browsers will work in the same way, without the need for user buy in to specific technology. This is the code version of my HTML 5 baseline template, you can see it rendered here ([demo of a HTML 5 template](#))

This template has been successfully validated using the W3C markup service as valid HTML 5: http://validator.w3.org/check?uri=http%3A%2F%2Fwww.mccran.co.uk%2Fexamples%2Fhtml5%2F

```
<!DOCTYPE HTML>
<html lang="en-GB">
    <head>
<meta charset="UTF-8">
<link rel="stylesheet" href="reset.css" type="text/css" media="all">
        <title>HTML 5 template</title>
    </head>

    <body>

<!-- header -->
    <header>

    <!-- navigation -->
        <nav>
            <ul>
        <li>nav item 1</li>
        <li>nav item 2</li>
        <li>nav item 3</li>
        <li>nav item 4</li>
            </ul>
        </nav>
```

```
                              <!-- end nav -->

                              </header>
                         <!-- end of header -->

                         <section id="content">
                              <h1>Header 1</h1>
sit amet, consectetur adipiscing elit. Maecenas auctor dolor in erat venenatis quis
na porta odio accumsan convallis. Curabitur ut orci orci, eget posuere enim. Etiam
                         malesuada id, bibendum sed purus. Sed arcu magna,

                              <article>
                              <h2>Header 2</h2>
or sit amet, consectetur adipiscing elit. Maecenas auctor dolor in erat venenatis
ec urna porta odio accumsan convallis. Curabitur ut orci orci, eget posuere enim.
                         id malesuada id, bibendum sed purus. Sed arcu magna,
                              </article>

                              </section>

                         <!-- bottom aside area -->
                           <aside class="wrapper">

                              </aside>

                              <!-- footer -->
                               <footer>

                               </footer>
                          <!-- end of footer -->


                              <!-- GA tracking -->
                      <script type="text/javascript">
                            var _gaq = _gaq || [];
                      _gaq.push(['_setAccount', 'UA-']);
                      _gaq.push(['_setDomainName', '']);
                        _gaq.push(['_trackPageview']);

                            (function() {
ar ga = document.createElement('script'); ga.type = 'text/javascript'; ga.async =
ps:' == document.location.protocol ? 'https://ssl' : 'http://www') + '.google-anal
 var s = document.getElementsByTagName('script')[0]; s.parentNode.insertBefore(ga,
                              })();

                              </script>

                              </body>
                              </html>
```