

Web Standards with Vision:

An introduction to Web Standards and its benefits if adopted by Orange Vision Ltd.

Introduction

Background

The web is now over a decade old. During that time it has seen many changes, both in its use, and in the way it is built. In the early years of its development, there was the infamous browser wars – take your pick – Netscape Navigator, or Microsoft's Internet Explorer. At Orange Vision, IE was chosen, a reasoned choice given its installed user base and the declining popularity of Netscape 4.

Skip forward to today, and the web is everywhere and on everything. Try taking your pick from an increasing list – not just of desktop browsers (the two old soldiers with a backlog of version numbers), but also an increasing range of portable devices with built in web browsing abilities. These devices also have a number of important differences between them – screen size, and input method being two examples.

With age, comes responsibility, and with the web a greater part of day-to-day life, so it needs to be able to be used by everybody, regardless of disability. Before disability rights legislation, whether you wanted to support a minority, but still important section of users was a personal choice. Now with acts such as the Disability Discrimination Act having been passed, alongside similar acts in other countries (the US equivalent is Section 508), the choice is now not whether to support accessibility or not, but a case of by how much.

With all this in mind, how is it possible to cater for such a range of devices and type of user? Content needs to be scalable and manageable on small devices, and make no assumption about how the user will view (or hear) it. Introducing Web Standards.

What are Web Standards?

Firstly we should acknowledge that 'Web Standards' are not actually standards at all. Strictly speaking they are 'recommendations' put forward by the W3C whose members then chose to support them. However, when its members include companies such as Microsoft and Netscape, as well as other technology firms such as Apple, Mozilla, Opera, BT, Sun Microsystems, IBM... and these companies have chosen to support these recommendations, then in real terms they can be regarded as standards.

HTML is one such standard, however its history has been clouded by its involvement in the browser wars, where it was adapted and modified to meet the needs of two companies, each trying to get a competitive edge over the other. This was attempted by allowing more sophisticated websites to be created that could only be supported by their particular browser.

This was futile, as no sooner had the one browser implemented new mark-up, the other would support it. However there were still differences in the way in which each browser rendered HTML (and later CSS), making it increasingly more difficult to support both. This was never going to last, especially with the growth of the Internet and the greater requirements of the Web.

At the beginning of this decade, control of HTML, and other web technologies was given back to its inventor, with both parties agreeing to support them. HTML 4.01, XHTML 1.0 (the successor to HTML – a reformulation of HTML in XML), CSS 2.1 and the W3C DOM, are four such recommended technologies, and for simplicities sake, we will call them 'Web Standards'

In order to meet the requirement of saleability, and accessibility, the ultimate aim of web standards is to separate content from its presentation. This makes sense when you understand that someone using a mobile phone does not want a heavily designed 800 pixel wide site on their 216 colour 200 pixel wide phone display. The same is true for a blind user that uses a screen reader to browse the web and cannot see the design anyway.

With this in mind, XHTML is designed to give the content *meaning*, whilst CSS alone is used to add *style* for those that require it.

However these standards have been designed as a continuum, in that current methods of design can still be used whilst supporting standards, whilst perhaps not achieving true separation. An example of this can be found in (X)HTML

HTML 4.01 – Whilst elements such as , <u> and other presentational elements have been striped out as they can be provided by CSS, as well as much propriety code such as <blink>, <marque>, many other presentational attributes still exist.

XHTML 1.0 (Transitional) – HTML 4.01 reformulated as XML, but still allowing some presentational code.

XHTML 1.0 Strict – All presentational code removed, as well as other features that can be provided by the DOM, one such example being target="blank" in links.

The same is true for WAI (Web Accessibility Initiative)

Priority 1 – Still allows the use of tables for layout, although these tables should provide a summary of their content.

Priority 2 – Tables should not be used for layout.

What are 'Standards Compliant Browsers'

Since the introduction of web standards, all desktop browsers that have been released since have aimed to support them. Whilst there are glitches and misinterpretations with some earlier versions, in essence they all support standards to an extent that we can in turn also support these standards.

In theory, we are no longer supporting a number of browsers, but one standard.

IE 6.0 = Gecko (Mozilla, Firefox...) = Apple WebCore (Safari, OmniWeb...) = Future Browsers

In practice this is not so true, as earlier versions of Internet Explorer misinterpret the recommendations. However, they can still be supported with only a few 'hacks'.

IE 5.0 = IE 5.5

Gecko (Mozilla, Firefox...) = Apple WebCore (Safari, OmniWeb...) = Future Browsers

What about Non Standards Compliant Browsers

Whilst sites can be built to support browsers introduced before the adoption of Web Standards (read as IE4, and NN4), it is far more difficult and costly to do so, with CSS support usually at CSS 1.0 (which doesn't support positioning), and which is inconsistently supported across these earlier browsers.

However, all is not lost as we can use the backwards compatibility features of XHTML to still provide users of such browsers the content they require, yet without the presentational aspects. In fact, they would be viewing the same sort of site as a user of a mobile phone or screen reader would be.

Should Orange Vision continue to support Non Standards Compliant browsers?

This question should really be "Should Orange Vision continue to support presentation in Non Standards Compliant Browsers?"

Content will always be able to be viewed in outdated browsers much in the same way as prehistoric browsers don't support tables but can still output their content.

CWHomes.co.uk is one site designed by Orange Vision that has a great number of users. During the month of January 2005, only 14 users (<1.0%) visited using a version 4 browser. This is matched with statistics available from other sources (W3Schools.com) that suggests users using NN4 alone equals 0.3%.

It is recommended that Orange Vision continue to drop support for Netscape 4.0 and now in addition Internet Explorer 4.0 too.

Current methods employed by Orange Vision and their future use

Use of and other depreciated elements.

Sites should no longer contain depreciated elements (list to follow). Not only are these no longer required (CSS can be used instead), but support for these elements is likely to be the first to disappear in future browser releases, and almost certain to be unsupported by other devices.

Use of HTML

Where as we currently use HTML 4, we should look to use a bare–minimum of XHTML 1.0 (Transitional). This still allows us to code with very much the same mark–up that we do already, but with a few important changes:

- Documents must now begin with a declared DOCTYPE
- All tags must be written in lowercase
- Attribute values must be quoted
- All attributes require values (i.e selected="selected")
- All tags must be closed
- All 'empty' tags to be closed too (i.e.
 becomes
)
- All < and & Characters should be encoded.

This promotes stricter coding practice and is more compatible with future technology.

Where possible it is desirable to use XHTML (Strict). This should be the ultimate aim, as in the future all sites are likely to need content and presentation truly separated.

Use of CSS

The current practice of a range of styles applied to text by means of class selectors needs to change to that of apply style to elements. By this I mean instead of:

<code><p class="textMainHead">Heading</p></code>	<code>.textMainHead {font-size: 24px}</code>
<code><p class="textMainSubHead">Sub-heading</p></code>	<code>.textMainHead {font-size: 18px}</code>
<code><p class="textMain">Paragraph</p></code>	<code>.textMainHead {font-size: 12px}</code>

use the following method instead:

<code><h1>Heading</h1></code>	<code>h1 {font-size: 24px}</code>
<code><h2>Sub-heading</h2></code>	<code>h2 {font-size: 18px}</code>
<code><p>Paragraph</p></code>	<code>h3 {font-size: 12px}</code>

Not only is this noticeably cleaner and less bandwidth intensive code, but by using semantic mark–up gives the page meaning to those who are viewing the site without the defined styles.

Table based layouts

Strictly speaking tables were only designed for displaying tabular data. However to achieve some presentational effects, tables have been used to achieve not only layout, but spacing between elements and line styles. Whilst spacing and line styles can now easily be replaced with CSS, the question of tables for layout is still acceptable for 'transitional' sites, but should still be kept to a bare minimum.

One example recent example of this is homeloansdirect.net where a transitional design was used. Here tables were only used for basic page layout.

Whilst the benefits of table based transitional layouts is support for older browsers (which we are likely to discontinue support for anyway), the disadvantages are that it decreases accessibility, adds additional mark–up, and doesn't scale well on smaller devices.

I recommend therefore that tables should only be used for the layout of forms (where it is still quite difficult to layout a form with CSS alone), and of course tabular data such as list of users etc.

Of course should a client have a large installed base of version 4 browsers, a transitional layout would be required.

JavaScript.

For reasons of accessibility, where JavaScript is used, the user must also be able to achieve the same function without having JavaScript enabled. One such example of this is pop-ups (when used) where:

```
<a href="#" onclick:"javascript:popup(page.html);">
```

becomes:

```
<a href="page.html" onclick="javascript:popup(this.href)return false;">
```

JavaScript and the DOM

Currently we predominantly use the proprietary IE DOM (Document Object Model) to target elements in our JavaScript's. Whilst some other browsers support it in compatibility modes, we should change to the W3C DOM, which *all* browsers support.

My knowledge of JavaScript is limited, but essentially, instead of using the IE DOM method `document.all`, we should start using `document.getElementById()` instead. This requires very little change in current practice, just changes in the way objects are targeted.

Changes that would need to be made to project workflow

Cross browser CSS bug checking.

When designing sites that are built using CSS positioning, it is quite likely that Internet Explorer (especially earlier versions) will misinterpret the design. A number of hacks are available should they be required so that this doesn't pose a great problem.

However to ensure sites will work correctly in all current and future browsers that implement standards correctly, it is recommended that sites are initially created to work flawlessly when viewed on a Gecko (Mozilla) based browser. Not only is Gecko a far stricter rendering engine, due to its close compliance to standards, a site will then look the same on all other browsers, as well as future versions of Internet Explorer too.

It is also then easier to then view in IE adding (a few) hacks as necessary to fully support this particular browser.

Mark-up Validation

To ensure our sites meet the standard that we have declared we are using in our DOCTYPE, we must validate using a validation service such as the one provided, free, by the W3C.

It is recommended that this should be carried out once after the basic HTML has been created, and then again after the site has been made dynamic by the programmer.

As we learn how to use the correct mark-up, this process will throw up less and less errors, and so not consume much time.

Accessibility

Currently we do not support accessibility guidelines. At the bare minimum we **must** build sites that achieve Priority 1 status. It is recommended that before a site goes live, part of the process of error checking should now include completing the WAI checklist and ensuring a site meets all Priority One guidelines, and perhaps further guidelines should the time be available.

What are the benefits, and what can we sell to our Customers?

- **Accessibility**
This has got to be the number one reason for switching to standards, and easiest feature to sell onto the client. Whilst basic accessibility guidelines should always be achieved, it is also possible to offer a number of additional options to increase accessibility
- **SEO**
Sites that are built with a far greater content to XHTML syntax ratio are friendlier to search engine robots, and are likely to rank higher than other older sites. This is especially true if there is a good deal of useful content.
- **Speed**
Most sites re-launching with new (CSS/XHTML) based designs often boast that they are now faster to download. Having looked at such sites, whilst they are faster than they were to their previous table based layouts, they can be much faster still, and include even less code, by using style sheets more effectively than they have done.
- **Ease of viewing on mobile devices**
- **Print Style Sheets**
- **Style switching (e.g. contrast, text size)**
- **Greatly reduced bandwidth**
- **Site can easily be changed to meet a change in brand direction (e.g. Home Loans Direct)**
- **Everyone else is doing it!**
- **Future compatibility**