



When using hierarchical navigation, you can either roll out or roll up the top levels, depending on how many options there are

match, and I tend to use this format: Page name: Site name > section name. This puts the important information first, and the page name would be the same as the main heading.

Consistency of navigation is important for usability and accessibility, and it also happens to be great for search engine optimisation (SEO) when combined with good HTML code.

Design considerations

Although most aspects of accessibility are taken care of at the HTML level, there are a few things to check at the storyboard stage. The obvious aspect is contrast: is there enough contrast between foreground and background colours? This applies to all text on a page, but it's most important for text within images, as this is difficult for the user to change. How much contrast is enough? The easiest way to check is by using a small tool called the Colour Contrast Analyser (www.nils.org.au/info.aspx?page=628), which compares the foreground and background colours and gives you a green tick or a red cross. I tend to use the numbers suggested by Hewlett-Packard, which hasn't caused any problems when testing with people, and allows for a slightly larger range of colours.

Layout is another factor to consider at the design stage, though it depends heavily on how it's coded later. One of the least common (yet most useful) things you can do for accessibility is to use a flexible layout that scales well over a range of conditions.

At Nomensa, layout behaviour is accounted for at the design stage and is probably the hardest thing for a designer to get used to. As part of the handover from design to coding, a description is provided to describe the intended behaviour of the design, which parts should flex and how. Also, large images are provided for the background images, so that they cope with scaling well.

Front-end code

This is where the action is and where most accessibility issues are created and solved. Separating style and content forms the basis of accessibility,

Who's responsible for accessibility? By Patrick Lauke

In web accessibility, you'll often hear emphasis being placed on the duty of web authors to create accessible content. However, this is only one part of the web accessibility equation.

Right at the planning stage, clients and site owners should gain a basic understanding of what accessibility is, why it's important for their project, and how to choose the right web authors/developers to do the job. An excellent starting point is the Publicly Available Specification (PAS) 78 Guide to Good Practice in Commissioning Accessible Websites, at tinyurl.com/yufg46, endorsed by the Disability Rights Commission (DRC).

Web authors and developers need to be aware of potential accessibility issues when it comes to developing their content. The Web Content Accessibility Guidelines (WCAG) at www.w3.org/WAI/intro/wcag provide a solid foundation for development and evaluation. However, authors still need an actual understanding of the rationale underpinning the various checkpoints and current best practices that not only satisfy the guidelines, but work in the real world (through user-testing and feedback).

In web development, "authoring tools" covers software such as Dreamweaver, web-based Content Management Systems, blogging tools and admin systems. Authoring tool developers must ensure that their products are both accessible in the way they can be operated and in the output they produce, in line with the Authoring Tool Accessibility Guidelines (ATAG) at www.w3.org/WAI/intro/atag.php.

Any effort on the part of web authors to add accessibility features is rendered useless if browsers and assistive technologies don't take advantage of them. User agent developers need to ensure that their products support these features and, most crucially, make them available to users in an accessible and obvious manner, as outlined in the User Agent Accessibility Guidelines (UAAG) www.w3.org/WAI/intro/uaag.php.



Finally, there has to be an onus on users themselves. They need to learn what user agents are available to them, how to use them, and how to configure them to best suit their particular needs. For example, if they require larger text or particular colour combinations, they should strive to gain the knowledge to set their preferences as defaults. If their current browser doesn't allow for this in a satisfactory manner, they should consider switching to one that does. BBC's My Web My Way (www.bbc.co.uk/accessibility) is a good resource for users, but organisations such as the RNIB also have a mandate to make this sort of information available to their stakeholders.

Only if all of these work together can a truly accessible experience be realised. Placing the onus on just one of these groups will result in sub-optimal solutions, such as web authors having to create custom solutions to make up for missing AT support or the user's ignorance of their browser's built-in functionality.

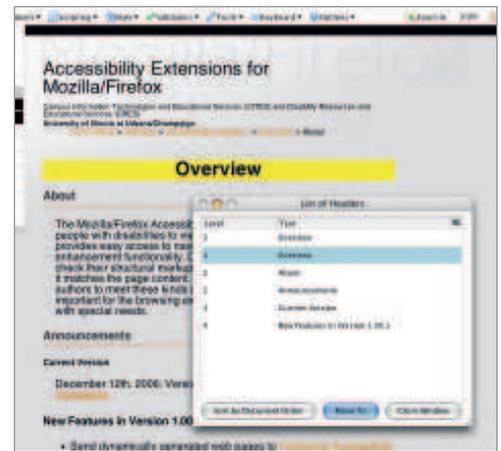
Patrick Lauke is committed to delivering websites using best practices and web standards such as table-free HTML and CSS markup. www.splintered.co.uk

and combined with valid, meaningful (X)HTML is the best foundation to start with.

As Andy Clarke covers on page 50, defining your underlying code by the meaning of the content is essential. Although HTML doesn't have elements to use for every situation, there are plenty to learn about and extend using "XHTML compounds", as Tantek Çelik described at Web Essentials 2005 (tantek.com/presentations/2005/09/elements-of-xhtml).

One of the main problems recently is that many people have "got" CSS, but not semantic (X)HTML. So instead of styling a heading, they'll add <div class="mainheading"> rather than <h1>.

When translating from design to code, it's important to know what you're trying to achieve, which is not pixel perfection. Even if it's slight differences in how browsers render a font, you'll never get pixel perfection across all circumstances. As Nate Koehley of Yahoo said at @Media 2005,



Several assistive technologies use HTML structure to help with navigation, such as headings and lists

