KM COLUMN

Accessibility tips for website construction

We all know accessibility is important, but precisely how does one make a website or intranet more accessible?

There is a great deal of hype on this topic and a lot of discussion too, yet vagueness and confusion persist. Web teams face a considerable amount of political pressure to 'be compliant', but often don't know where to start. This can result in aggravation, misdirection of effort and ultimately a failure to make the website any more accessible.

Often what is needed is a pragmatic view based on real experience, to reveal what is really important and what should be tackled first.

This paper provides ten key tips to help improve the accessibility of any website, or intranet. It's *not* intended to be an introduction to web accessibility (instead see the earlier article *Introduction to web accessibility* for a brief overview of the topic).

Nor is this intended to be an exhaustive manual covering every detail of every accessibility technique. Besides being counterproductive and far from helpful as a starting point, it is just not possible to do this. Best practice changes constantly and one must focus on the underlying approach rather than specific details.

These tips cover the basics, getting code quality up to scratch, for instance. Higher levels of accessibility can be explored once these sanity factors are addressed.

Readers are encouraged to make use of the many links to further information sources, provided throughout this paper, from which more detail can be obtained.

Patrick Kennedy is a user experience specialist at Step Two Designs, an intranet and content management consultancy based in Sydney, Australia. Patrick specialises in website development, accessibility, information architecture and usability.

1. Don't panic

It's not the end of the world. It's not necessary to scale the pinnacle of accessibility in a single bound. It's unlikely anyone will take legal action tomorrow.

Whilst it's not as simple as inserting some 'alt text' or 'checking it with Bobby', making a website more accessible is not that hard. Relatively small changes to a website can make a big improvement for users.

> # Accessibility is not simply about 'disabled people'

Special skills are not necessary, the chances are the team who built, or are currently building, the website can do it. They just need a helping hand to get started.

Similarly, there may be little or no requirement for extra software, hardware or other expenses. But the one cost that needs to be incorporated into project plans is time; crafting an accessible website can take longer, especially if current practices need substantial adjustment.

2. Know your users

Keep in mind that the aim is to design for users. As with all user-centred design activity, it's imperative to understand the users' needs, through any means possible. For instance: direct observation, log analysis or even surveys.

Remember, accessibility is not simply about 'disabled people'. There are a huge variety of circumstances in which users may find a website inaccessible, many unrelated to disabilities.

As such, it's quite enlightening to step into someone elses' shoes and experience the web as they do. For example, try not using the mouse, without your glasses, with low resolution settings, on a black and white screen, zoomed in, over a slow connection,

using a text only browser, using a 'screen reader' or 'talking browser', or on a mobile phone. The potential problems will quickly become apparent, as will the resulting frustration these users probably feel.

This should be supplemented by thorough testing, see tip 8 for more on testing.

3. Start with the official guidelines

A great deal of effort has gone into formulating guidelines for creating accessible websites. Whilst they can be vague and confusing at times, the Web Content Accessibility Guidelines (WCAG) published by the World Wide Web Consortium (W3C) are the accepted basis of accessibility best-practice. To start with, read the following:

www.w3.org/WAI/intro/wcag.php

Adding further weight to their authority, most legislative policy relevant to accessibility (including Australia's own Disability Discrimination Act) make direct reference to the W3C's guidelines, in effect making them the law. For more information see:

- Australian Disability Rights policy
  www.hreoc.gov.au/disability_rights/

- UK Disability Discrimination Act
  www.uiaccess.com/dda-uk.html

- US Rehabilitation Act, section 508
  www.section508.gov

In the USA, more emphasis is given to 'Section 508' guidelines, but these are very similar to the W3C guidelines.

The W3C's guidelines form the basis of best-practice

In addition to, and often in partnership with, the W3C are the organisations representing groups within the community, such as people with physical or mental impairments. In the interests of their members, some of these organisations have become experts in website accessibility and are a great source of information on the subject. For example:

- Vision Australia
  www.visionaustralia.org.au

- Royal National Institute for the Blind
  www.rnib.org.uk

- Epilepsy Action
  www.epilepsy.org.uk/info/photoweb.html

4. Use standard code

Technical standards have also been published by the W3C concerning how pages are constructed, namely XHTML and CSS.

The currently recommended standard for markup is XHTML 1.0 Strict and CSS 2.0, which have been widely accepted and used by developers for some time now. The next version of both XHTML and CSS are still in the draft stage and as such will not be widely supported for some time yet.

Separation of content from presentation is critical

The key benefits of these standards are stricter rules for use of tags and attributes, and the ability to properly validate the markup, providing a built-in quality control.

This means websites can now be constructed using a standardised language which can be translated by browsers, and other software and devices that people may be using, into a form that caters for their individual needs. In effect, making the information independent of the interface, or to use a more common phrase: separating content from presentation.

Separation of content from presentation is critical for good accessibility. XHTML marks up the content, giving it structure and functionality. CSS is applied to this markup to make it look a certain way. By separating off CSS (ideally in external CSS files) in this way, several benefits are gained:

- styles can be switched (to accommodate different user circumstances)

- users can take this one step further and replace or supplement the author's styles with their own

- smaller amount of markup code

- CSS files are cached between pages, improving performance

- content can be structured logically, without necessarily affecting appearance

An important aspect of adopting web-standards is the practice of semantic markup, whereby tags are used for their intended purpose, and as succinctly as possible. This way the meaning of the content can be known based on what markup is used. Whilst this is still an area under develop-

ment, the general approach is a good tool in building accessible markup.

Superfluous markup, that is used purely for presentation, should be avoided. A good example of this is the once common practice of using HTML tables for page layout.

It's worth noting that claims of standards based markup are often made by software vendors. Always check the resulting code and fix it if possible, or choose other software if necessary.

For further information on standards see:

- XHTML 1.0 standard
  www.w3.org/TR/xhtml1/

- W3C markup validation tools
  validator.w3.org
  jigsaw.w3.org/css-validator

- Web Standards Project (WaSP)
  www.webstandards.org

- Web Standards Group (WSG)
  www.webstandardsgroup.org

- A List Apart
  www.alistapart.com

5. Accessify your markup

Simply adopting web-standards is a good start, but not quite enough to make a website highly accessible. There are a number of relatively simple code enhancements which make it more accessible. Here are a few crucial points:

- use `alt` and `longdesc` attributes for non-decorative images

- leave `alt` attributes empty for decorative images

- use the correct tags for the content

- ensure semantic source code order

- 'skip links' to take users directly to content, skipping the navigation

- use text that gives more information but is hidden by default using CSS

- ensure link text is unique per page, use hidden text if need be

- use the `title` attribute for hyperlinks to clarify purpose of the link

- avoid fixed size text to allow users to increase or decrease size as required

- use background images for decorative effects without the need for extra markup

- use the `<label>` tag and `id` attributes in forms

- use `<fieldset>` and `<legend>` tags in forms

- correctly label form fields; `<label>` first for all but radio buttons and checkboxes

- use `<thead>`, `<tbody>` and `<tfoot>` tags in tables

- use `id` and `headers` attributes in tables

- use the `<caption>` tag to describe tables

- use correct character entity encoding to avoid garbled characters

- use `tabindex` attributes for keyboard shortcuts

Always check "accessible" markup output by authoring and content management software, and fix it if possible, or choose other software.

> What users actually 'see' will in all likelihood be quite different to what you designed

6. Adjust your design and layout

The visual design given to a website by authors and designers is only the *default* design. User circumstances, differences between software, assistive technologies etc all mean that what users actually 'see' will in all likelihood be quite different.

With this in mind, there are some things that will make the default design more accessible:

- choose colours for sufficient contrast and brightness difference, check with tools such as the Colour Contrast Analyser:
  www.visionaustralia.org.au/
  info.aspx?page=628

- don't rely solely on colour for user feedback (such as 'you are here' indicators in navigation)

- choose fonts (size, face, style) to optimise readability, and allow user to adjust to suit their needs

- design navigation that is suitable for the amount and depth of links

- design to avoid use of popup windows, unecessary DHTML and scripting, automated reloading of pages etc

- reorder or simplify forms and tables to make their markup more accessible

In general animation and rich media, such as Flash, should be avoided unless they serve some useful purpose, as opposed to decoration or gimmick. In these cases ensure appropriate steps are taken, see tip 7.

7. Consider your content

Content itself can be inaccessible. Most importantly all content, other than plain markup, should have an alternative. For example graphics, animation, audio, video, and other rich media formats, should all have alternative versions. Where possible the substitution of these alternatives should be automatic and seamless, but users should also be given the choice to switch between formats. The aim to allow the content to degrade gracefully to a level suitable for the user.

> ## Content itself can be inaccessible

Due to the popularity of using Flash for functionality, as a first line of defence, Macromedia's accessibility features should be employed to make that functionality more accessible. Of course alternatives should still be provided.

In addition to the technological assessment of whether the content *format* is accessible, there is the question of appropriateness and quality of the content itself. Content can be inaccessible due to excessive length, verbosity, complexity or through being poorly written.

In general, content for websites should be treated quite differently from other kinds of publications. It should be shorter and 'scannable' by the reader, for instance making use of the benefits of a hypermedia environment; links, emphasis of key words, headings to give structure etc. A popular writing style suitable for the web is known as the 'inverted pyramid' format, as used in newspaper journalism. For more information on this and other tips for writing for the web, see this article by Jakob Nielsen:

www.useit.com/alertbox/9606.html

An important, yet often overlooked, aspect of website content is the 'microcontent'; the small pieces of text found on a website, such as page titles, link names, `alt` text etc.

These all need to be accurate and informative, yet concise. The best example would be the text used for links, which should be informative, enticing and as short as possible, yet also meaningful when read out of context.

8. Test, test, test

There's no substitute for realistic testing, using a wide variety of devices, software and scenarios, to simulate real user's experiences. But there's usually no need to hire expensive consultants or have special facilities.

There are several useful tools for testing websites; valid markup, errors, broken links etc. Other tools, such as browser extensions, allow you to inspect many properties of a page, helping with debugging.

But don't rely solely on automated testing tools. They can only perform a relatively superficial check of code, and whilst this is important, there's much more to accessibility than that. By all means use tools to verify progress and check for the bleeding obvious, but ultimately guidelines need to be interpreted, techniques need to be applied intelligently, all according to the specific problems of each website and it's audience.

Quite simply it takes a person not a machine to judge what is accessible. For example, a tool can check that markup is valid, but can't determine if the page makes sense.

Web teams may be surprised to find that the staff in their company are quite diverse, making a great test audience. Remember, most users will benefit in some way from making a website more accessible and thus can give useful feedback. Later on, of course, when aiming to refine the website further, specialised test facilities and users may be required.

> ## There's no substitute for realistic testing

Lastly, in terms of testing procedures, a good rule of thumb used in many areas of web development is "test early and test often". It's crucial to build time into your process, and not leave testing till the end. The old adage is quite true; it's much more expensive to fix something late in the project than early on.

For more information on things to help test:

- WebXACT accessibility checking tool
  webxact.watchfire.com

- pwWebSpeak talking browser
  www.soundlinks.com/pwgen.htm

- JAWS screen reader
  www.freedomscientific.com/fs_products/
  software_jaws.asp

- IBM Home Page Reader
  www-306.ibm.com/able/dwnlds/
  hpr4trial.html

- Lynx text-only browser
  lynx.browser.org

- Web developer toolbar for Firefox
  chrispederick.com/work/webdeveloper/

- Accessibility toolbar for IE
  www.visionaustralia.org.au/
  info.aspx?page=614

The best approach is to make a start!

9. There are no shortcuts

There is no easy answer to accessibility, there's no magic button that will instantly fix a website.

It might seem like an insurmountable task, but start somewhere. Every little bit will make the website more accessible. Don't try and change everything at once. Start by updating templates to be web-standards compliant, for instance. Then move onto ensuring alternative content formats, then perhaps move on to tweaking forms and tables.

An additional challenge is the opposition to accessibility that may be faced in some organisations. It may be necessary to cut red-tape as well as cutting code. The web team or manager may need to sell the idea that these steps are important and that real benefits will be achieved. If all else fails, mention the legal obligations, or the potential increase in revenue from users who formerly were unable to use the website let alone order something through it.

10. Don't stop now

There's a lot more to do, this is just the beginning! Much more can be done to further refine a website, for example handling tricky

issues such as multilingual content, rich media, business application interfaces, PDF documents etc.

Accessibility is a constantly moving target

Additionally, accessibility is a constantly moving target. Before too long WCAG 2.0 will be released and you'll have to catch up again. And of course best practice techniques and opinion are constantly shifting.

Fortunately, the web is the best source of information on itself, and a great deal of help is available for dealing with accessibility, including some excellent dedicated websites:

- Accessibility.com.au
  www.accessibility.com.au

- Accessify.com
  www.accessify.com

- Juicy Studio
  www.juicystudio.com

- Simply Accessible
  www.simplyaccessible.org

...in addition to those already mentioned. Most of the websites listed here are complemented by mailing lists and discussion forums, and some other good examples are:

- e-Access Bulletin (UK)
  www.headstar.com/eab/

- Accessify discussion forums
  www.accessifyforum.com

- Section 508 discussion forums
  www.accessibilityforum.org

It's now fashionable amongst the web design community to champion web-standards and accessibility, and there are many talented people putting their minds to work on finding better solutions, and happily sharing their results.

This is an invaluable source of considerable knowledge, usually distributed in the form of 'blogs'. In addition to articles on the latest techniques, there is often constructive debate over big picture issues. Some good examples are:

- Joe Clark
  www.joeclark.org/access/webaccess/

- Drew McLellan
  www.24ways.org

- Derek Featherstone
  www.boxofchocolates.ca
- Jeffery Zeldman
  www.zeldman.com
- Eric Meyer
  *www.meyerweb.com/eric/thoughts/
  category/tech/standards/*

Lastly, take a break from online and read a real book. There is a growing list of very useful books available on accessibility and web standards.

Most books on any web technology quickly become out of date, but good ones give advice on a broad approach as well as specific techniques, so even if the example code gets old the general idea will remain valid. Some good examples are:

- Constructing Accessible Web Sites
  *www.glasshaus.com/
  bookInfo.asp?bookId=44*
- Building Accessible Websites
  www.joeclark.org/book/
- Designing with Web Standards
  www.zeldman.com/dwws/
- Eric Meyer on CSS
  www.ericmeyeroncss.com

In conclusion

There is a stigma surrounding accessibility and most people write it off as being too hard. This is easy to understand considering those responsible for maintaining websites and intranets are probably not accessibility experts, nor are any of their colleagues. Yet as this article shows, some relatively simple steps will allow them to make a start at increasing the accessibility of their site.

The most important thing to keep in mind is that accessibility needs to become part of the standard operating procedure; to be considered first rather than tacked on as an after thought.

The industry is approaching a turning point in terms of the ability to deliver truly accessible solutions, and widespread improvements are just around the corner. Aspects of user-centred design have been integrated into web design leading to real benefits, in a similar way, accessibility will soon be a seamless and a ubiquitous part of website development, informed by audience needs and verified through testing.

Yet, as with any complex and growing field of expertise, there is significant ground to be covered before stepping up to this new level. To ensure they are ready for 'tomorrow's accessibility', website and intranet development teams need to be up to speed with current best practice. The above ten tips should help get them there.

More articles!

Find out when new papers are published:
www.steptwo.com.au/papers/subscribe
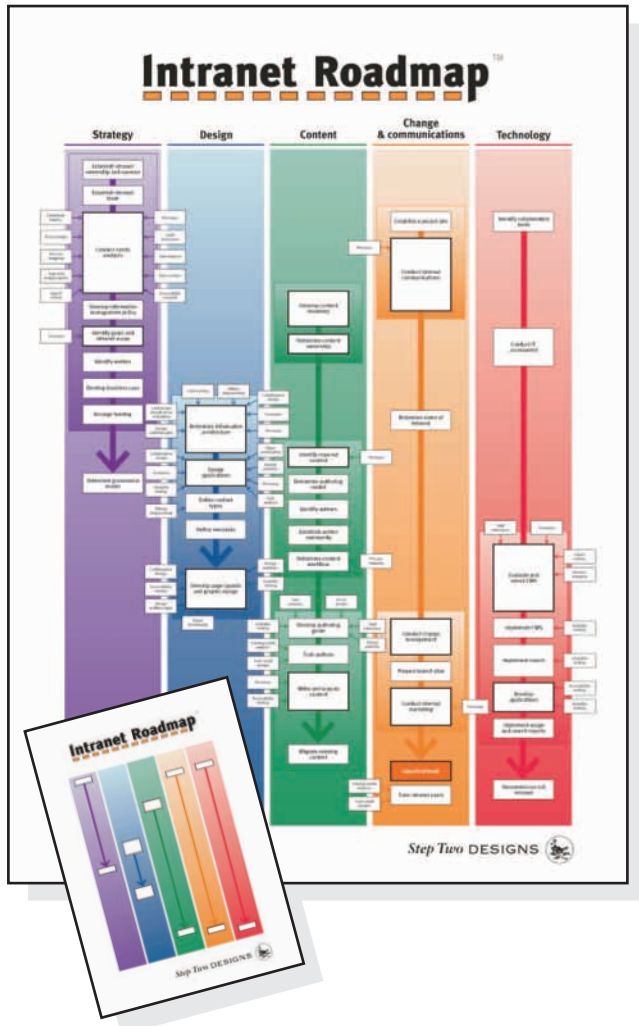
Did you like this article?

Send your thoughts and feedback to:
patrick@steptwo.com.au

Step Two DESIGNS

www.steptwo.com.au • contact@steptwo.com.au
knowledge management • content management • intranets

Intranet Roadmap™

The **Intranet Roadmap™** provides the first truly comphrehensive methodology that describes all the activities required to develop (or redevelop) an intranet.

Beyond just implementing software or redesigning the site, the Intranet Roadmap covers activities in five key streams:

- **strategy**
- **design**
- **content**
- **change & communications**
- **technology**

The Intranet Roadmap does more than just indicate what activities need to be conducted, it clearly shows the sequence of activities required, and the techniques that will help at every stage of the project.

The Intranet Roadmap is delivered in two forms:

- **full colour A1 wallchart**
- **supporting 54 page booklet**

The **wallchart** lists the key activities required in each of the project streams. It also highlights which activities (such as usability testing, affinity diagramming, personas and collaborative design) can be used to support individual activities.

The supporting **booklet** then provides an overview of every activity and technique listed on the Intranet Roadmap, as well as linking to further resources and information.

The combination of the wallchart and booklet will be invaluable for any team looking to develop or redevelop an intranet, and it will assist in both planning and reviewing the approach taken.
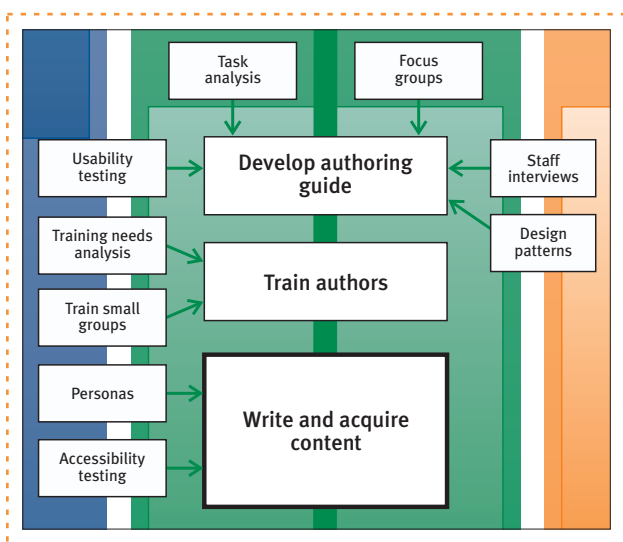
US$120 + $15 P&P
Secure online ordering:
www.steptwo.com.au

Step Two DESIGNS