Web Accessibility Guidelines

If you want to reach your total potential audience, your company's website needs to be accessible to users with disabilities. More and more, companies recognize that good web design is accessible design. Just as curb cuts help people with strollers as well as wheelchairs, an accessible website works better for everyone. The following document offers standards to build accessibility into your websites. As these are technical instructions, managers may prefer to share and review the guidelines with their web team.

ICI Philosophy

The Institute for Community Inclusion (ICI) strives to make all of its web products accessible and usable to all people, regardless of ability or disability. The overall goal is for our web products to be accessible and usable to anyone connecting in any way. This means that they work for people using screen readers, text-only browsers, keyboards only (no mouse), wireless (phone, PDA), slow connections (dial-up), etc.

The following are the suggested guidelines to follow when building a web product. ICI web developers are available for consultation on accessibility issues.

Guidelines

Section 508

- These are the guidelines the U.S. government has created to evaluate accessibility in both real and virtual places (i.e., the internet). www.section508.gov/index.cfm?FuseAction=Content&ID=12#Web

Web Content Accessibility Guidelines 1.0

- These are the guidelines the World Wide Web Consortium has created in an effort to have standards for developers of online content. Follow Level 2 at the minimum. www.w3.org/TR/WAI-WEBCONTENT

Standards

xHTML (high priority)

- All templates/modules will pass validation before content is added. Strive to make content valid, although the realities of multiple coders and old pages at times make this impossible. www.w3.org/TR/xhtml11

CSS 2.0 (low priority)

- CSS strives to achieve both validation (compliant to CSS standards) and browser support, which unfortunately are two different things. www.w3.org/TR/CSS21

Mark-Up Guidelines

- Use xHTML and CSS to separate content from style.
- Use proper mark-up.
- Organize pages in a logical manner, meaning they will read correctly, top to bottom, without CSS styling.
- Use headings (<h1>, <h2>, etc.) to organize content.
- Use a logical tabbing order.
- Use list tags (<ul>, <ol>, <dl>) for lists and menus. (Note: Definition lists work very well for CSS layouts).
- Do not use deprecated tags like <font>, <b>, <i>.
- Use tables only for displaying tabular data (with rare exceptions). General page layouts will be done with CSS.
- All pages will have unique and meaningful titles. Standard format is "page name [or article title], name of site."
- Site navigation and menus come after the content on pages other than the homepage. This prevents users of assistive technologies from having to tab through menus on every new page to get to the content. Alternatively, a "skip to content" link can be used as a secondary option.
  - > Navigation and menus should be coded with xHTML markup, not rendered on the fly with a scripting language like JavaScript/ECMAscript.
- Use text in place of images when possible. No graphical menus will be used.
- Use <abbr> and <acronym> elements when appropriate. Use the title attribute to define the abbreviations and acronyms.
- Use <label> and <fieldset> elements for forms.
- Use rel attributes for <link> and <a> elements: www.w3.org/TR/2002/WD-xhtml2-20020805/abstraction.html-dt_LinkTypes

CSS Guidelines

- Try to not rely on "classes". Rather, write your stylesheet to format tags within a specified <div>.
- Avoid using <span> tags.
- Avoid "div soup," which is when you overuse divs or have many nested divs.
- Name ids and classes with semantics in mind. Good: "content", "primary-navigation", "search-input", "search-results". Bad: "bluelinediv" or "255pixelbox".
- Do not rely on browser-specific hacks. A few might be necessary, such as the @import hack to prevent NS4 from using CSS, but don't write separate styles for separate browsers.
- The majority of measurements within pages will be done with "em" or "%". Control fonts with CSS keywords (small, x-small, etc.). Use pixels only with images and related items.

Design Guidelines

- Design pages to use a "liquid design" technique. This means a technique that does not rely on fixed sizes. Users can resize windows, view the site on small screens, and increase or decrease fonts without breaking the design.
- A design doesn't need to look perfect in an unsupported browser, but it does have to degrade gracefully and still look logical and usable.
- Use contrast between foreground and background colors.
- Use client-side scripting (JavaScript, ECMAscript) sparingly. No page should require client-side scripting to be usable.

Content Guidelines

- Avoid contextually meaningless link text, like "click here", "learn more", or "go".
- Links should describe the destination page/resource.
- Use title attributes to provide additional contextual meaning for links.
- For the sake of screen readers, avoid combining two words into one, such as "homepage". Use "home page" instead.

Supported Browsers

- Support anything newer than Netscape 4.x and early versions of Internet Explorer. You may also wish to support Lynx.

Usability Tests

The best option is to do usability tests with a group of people who have varied disabilities. If that is not possible, make sure the site works under these conditions:

- With style sheets disabled
- With images disabled
- With javascript
- Without mouse (keyboard only)
- With text very large
- With screen size very small or very large (within reason)
- With an alternative stylesheet (high contrast, large text)
- With JAWS (demo versions are available)

Note: ICI recommends using the Mozilla (or Firefox) browser with the Web Developer extension by Chris Pederick. Mozilla is a standards-based browser, and the Web Developer plug-in allows the designer to replicate the above conditions. The browser is available at www.mozilla.org and the Web Developer extension at www.chrispederick.com/work/firefox/webdeveloper.

Written by Jeff Coburn, Web Specialist, ICI/UMass Boston. Special thanks to Danielle Dreilinger and Dan Mozgai.

www.communityinclusion.org