Web Accessibility Quick Guide

Ten things you can do to make your website more accessible

The OIT Disabilities & Computing Program | Spring 2010

UCLA OIT

1 Provide alternative text (“alt text”) for all images

Assistive technologies such as screen-readers (for computer users who are blind or have low vision) convert the textual elements of a webpage to electronic speech. When an image is encountered, screen-readers rely on the alt attribute content. This alt text should convey information that is equivalent to the meaning or purpose of the image. All images should therefore have an alt text description provided in the source code. The alt text should be as succinct as possible. For instance, if an image of an institute’s logo is provided on the website, the alt text should read simply the name of the institute, as this is the information that is critical to the user; not “logo,” “(institute)’s logo,” or a description of what’s happening in the logo (i.e. “logo showing bird flying over planet”). The alt text might require a lengthier description if the image is used to illustrate a point. Alt text should also be included for images that are purely decorative. Here, the code should read: <alt=””>. The empty alt attribute cues screen-readers to skip the image when reading the website.

Avoid using images of text. Screen-reading software cannot interpret these images unless alt text is included. Low vision users may wish to enlarge text, change the text color, etc. If textual information is fixed in an image, users cannot manipulate the text to suit their needs. Text should only be used in an image if a particular style of text is integral to the information conveyed on the website and it cannot be achieved using normal text elements (i.e. logo/branded elements). As with all images, be sure to include an alt text description. Generally this should read the same as the embedded text. Context is king in choosing the most appropriate alt text.

2 Provide organized and structured content

Well-organized and clearly-structured content is a critical element of any website, as it ensures all users will find the information they need. Pages should be clearly titled with each header and sub-header visually distinguishable from the main text, and this method of organization should be consistent between all pages of the site. Headers and sub-headers should be clearly defined as such in the source code (h1, h2, h3, etc), as screen-readers rely on this identification in order to scan through the major sections and contents of each page. Thinking of the site as being organized in an outline format often helps clarify the structure of the site. It is also important that users understand where they are in relation to the site hierarchy at all times. Breadcrumbs and extendable navigation menus can be helpful in ensuring that users understand their location within the site.

3 Provide text alternatives for non-text content

Many assistive technologies rely on electronic text to provide information to hearing-impaired users, in addition to those that are visually-impaired. If audio and/or video components are included on the website, electronic text should be provided as a substitute or companion to this type of content. For time-based media, the document should provide correctly-sequenced textual descriptions of content. The goal is to provide users with text-based information that is equivalent to that which is presented in any audio/visual component of the site. Captions should also be provided for all pre-recorded audio content.

4 Associate labels with all form fields

When screen-readers encounter a form input field, they look for a corresponding label to cue the user what to enter in the box. If the label is missing, the user cannot tell what information should be entered. Several unlabeled buttons can leave keyboard users struggling to guess what to put in the box or what option to select. Use the “for” attribute with the label tag to create associations between textual labels and the fields they correspond to. The value of the “for” attribute should match the input or text area element’s id.

Example code: <label for=”name”>Name</label><input type=”text” id=”name” name=”name” />

5 Allow users to turn on/off audio and visual content

Audio and visual activity on a website can be distracting for many users. If the site includes visual action (such as flash animation or rotating images) or audio content that starts automatically and last longer than five seconds, users should be given the option to pause/turn off this action.

6 Optimize text legibility

It is imperative that viewers can easily distinguish text from the background. Make sure you choose a foreground/background color combination that has a high contrast ratio, and avoid putting text on busy background images. For all text under 18-point font (including links and visited links), the contrast ratio between text color and background color should be at least 4.5. For text that appears in 18-point font and above, the minimum contrast ratio is 3. There are several online resources for establishing contrast ratio, including this website: http://snook.ca/technical/colour_contrast/colour.html. Note: Although it is recommended that they do so, logos do not need to meet color contrast minimums.

In addition to choosing a high-contrast foreground/background color combination, optimize text legibility by avoiding the use of justified text and very small text (under 9-point). Paragraph text should have line spacing of 1.5 and should not exceed 80 characters in width. Sans-serif fonts such as Verdana, Trebuchet, Tahoma and Arial are commonly used web-legible fonts, although some serif fonts including Georgia and Times New Roman are also effective on-screen. Avoid extensive use of all-caps and italicized text.

7 Do not use color as the only indicator of possible action

Visually-impaired users may rely on indicators besides color to signal that an action is possible. Links, etc. should be presented using a secondary indicator (such as an underline) in addition to having a color that differs from the main text.

8 Provide users enough time to read and use content

For elements of the website that include a timed component, users must be provided with a way of manipulating the amount of time provided, such as turning on/off the time limit before encountering it, or extending the time period when time is up.

9 Allow for various methods of navigation

Many users with disabilities have access to only a single method of input/content access, most often a keyboard or a keyboard like device. Navigating a website should be device-independent; that is, a user should not be forced to use a specific device to navigate a website. The most common form of this is navigation menus that popup submenus when a mouse hovers over a certain item. Users who cannot use a mouse will not be able to then access specific pages in those submenus. Websites should accommodate device-independent navigation by allowing users to access content through various methods of input. In particular, javascript 'hover' or 'mouse-over' effects should have a corresponding 'focus' effect and vice-versa.

10 Use caution when developing dynamic content

Use of Javascript introduces several accessibility problems. Dynamic content (via Javascript, Flash, AJAX or other dynamic behaviors) cannot always be detected by assistive technologies. Some assistive technologies will pick up and report the changes, while others may only partially report changes or fail to report changed content altogether. Technologies that are able to identify changing content may have difficulty in intelligently determining which of these changes should be reported to the user; many changes occur in elements that are invisible to typical users, etc. In short, sites that incorporate dynamic content are less likely to provide users relying on adaptive technology with the same information as those who use a typical web-browser set-up. Javascript-related accessibility issues can be avoided through the careful use of HTML DOM-compliant solutions or CSS.

For more accessibility tips and examples, please visit:

Web Accessibility at the University of California | http://www.ucop.edu/irc/itaccessibility/resources/welcome.html
OIT Disabilities and Computing Program Resources | http://dcp.ucla.edu/resources