Practical Solutions to Common Website Accessibility Issues

A companion to a presentation (almost) given on October 28, 2017 at the
Hawai‘i Library Association / Hawai‘i Association of School Librarians Joint Conference
by Ralph Toyama, Systems Librarian, Leeward Community College
rtoyama@hawaii.edu

Session Description: Making a website accessible to people with disabilities requires thoughtful reflection, awareness of adaptive technology, and often the manipulation of a web page's source code. This session will address topics such as providing alternative access to the content of visual images, adding captions to YouTube videos, and addressing specific issues related to the use of scripts. Basic knowledge of HTML and familiarity of CSS is necessary for this presentation, as codes and tags will be discussed.

There are things you can do to address problems that make it difficult or impossible for people with sensory or physical disabilities to access the content of a web page. While web authoring tools and content management platforms often have features that enhance accessibility, they cannot fully automate it -- it still takes an awareness of the problems and the exercise of judgement to make a web page truly accessible.

Organizations that fail to make its website accessible may face legal action. Organizations may address the threat by centralizing or regulating website production, at the expense of the autonomy of their constituent units. For libraries, our websites are service delivery platforms and teaching tools; we need to control how our resources and services are presented. Therefore, it is imperative that we have the expertise to address their own accessibility issues to retain control of our own websites.

Resources and Guidance

http://webaim.org
WebAIM (Web Accessibility in Mind) is a non-profit organization that is a great source of guidance for accessibility issues.

https://www.w3.org/WAI
The Web Accessibility Initiative of the World Wide Web Consortium (the main standards organization for the Internet) has published a set of web content accessibility guidelines (WCAG). These guidelines have criteria that address issues with differing levels of severity. Priority 1 criteria address issues that make it impossible to access content, Priority 2 criteria address issues that make it difficult to access content, Priority 3 criteria make content easier to access. There are three levels of website conformance: Level A (meets Priority 1 criteria), Level AA (Priority 1 and Priority 2), and Level AAA (Priority 1, 2, and 3). Many organizations strive for WCAG 2.0 Level AA conformance.

They also publish the WAI-ARIA technical standard for Accessible Rich Internet Applications. Adding ARIA codes to your web page can help you address issues with dynamic content and user interface components.

Accessibility Standards

U.S. statutory law has no specific accessibility standard for non-federal websites, although legal settlements for accessibility compaints usually specify WCAG 2.0 Level AA. Websites for federal government agencies must comply with the accessibility standards of Section 508 of the Rehabilitation Act. A revision of Section 508 that will take effect in January 2018 adopts WCAG 2.0 Level AA as the basis for its standard.

Checking Your Web Pages

There are websites can analyze your web page for accessibility problems. The WAVE tool at http://wave.webaim.org identifies both problems and features, and offers suggestions for improving accessibility. You can enter a URL at their site, or

you can install a Chrome extension that puts a button on your browser that lets you quickly check a web page. A more stringent check is at http://cynthiasays.com.

JAWS is a powerful and popular screen reading program that reads out text in a computer-generated voice. Unfortunately you have to pay a hefty license fee to use it, even if only to check to see how your web pages sound to a blind person. NVDA from https://www.nvaccess.org is screen reader you can download for free (donations welcome). It doesn't fully support all ARIA tags.

One test to see how understandable your page is to someone who can't see the formatting is to view it without the CSS. There are browser extensions that can do this. You can also do this my modifying your page code.

Marking Content

Use HTML tags to indicate the function of the different parts of your document. This gives semantic context that helps adaptive technology to effectively present the context. Screen readers will announce things like headings and paragraphs to help the user understand the content.

Mark your headings appropriately – **`<h1>`** for your page's main heading, **`<h2>`** for your subheadings, and **`<h3>`** through **`<h6>`** if necessary to further subdivide your page into deeper layers. It's a common mistake to rely on visual styling to make words look like a heading, because things like bolding or using a larger font are visual cues not available to someone who is listening to a screen reader.

Use HTML lists when appropriate – don't just use the **`<br />`** tag to break lines to make something look like a list.

Create tables only when you need to present tabular data. Don't use tables just to control layout, because the screen reader is going to read out the table structure. Header cells in a table should be tagged with **`<th>`**, rather than the **`<td>`** tag that marks regular data cells.

Ordering Content

You can arrange the content on your page in such a way that the arrangement itself conveys some meaning. Ideally, you should try to order your content in such a way that it still makes sense even if you couldn't see the visual arrangement.

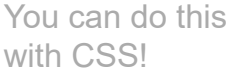
Declaring the Primary Language of the Page

This can help adaptive technology properly present the page content. Within the <html> tag near the top of the file, include this attribute – **`lang="en-US"`**

A Very Common Problem: Relying on Graphics to Convey Information

Words that are part of a graphic image are not accessible.

There are ways you can achieve the impact of a graphic image without relying on a graphic image to convey information. There are also ways to provide descriptions of graphic images to convey their content.



For an example of the first approach, if you want to emphasize a point by putting a statement in a colored box, you can use CSS to create a box with a colored background and rounded corners, and put your statement in the box as a line of text. Not only is this the best way to present information for screen reading technology, but when your information is on the screen in the form of text, it enlarges well when you use the browser's size control.

When you do use graphics, you may have to provide alternative text that either describes the image, describes its function, or provides the information given in the image.

The <img> tag is used to specify a graphic image to be displayed on a web page. All <img> tags should contain an "alt" attribute. For decorative graphics, the alt attribute may be empty – **`alt=""`**. An empty alt attribute tells a screen reader that the image can be ignored.

```
<img src="roof.jpg" alt="roof being blown off a house by hurricane winds"  />
<img src="rightarrow.gif" alt="next page"  />
<img src="squiggly.gif" alt=""  />
```

If an image can't be adequately described briefly, a "longdesc" attribute in the <img> tag may be used to provide a link to longer description, either on another web page or somewhere else on the same page. However, because longdesc support is inconsistent between browsers, some recommend that a conventional link to the description also be given near the graphic.

```
<img src= "poster.jpg" alt="Spring Festival poster" longdesc="festivaltext.html"  />
<a href="festivaltext.html">Accessible text</a>
```

A long description formatted as a web page with appropriate markup can be communicated more effectively than a long string of words.

For guidance in composing alternative text, see http://webaim.org/techniques/alttext. This article discusses assessing the image in the context of the page to determine the appropriate level of detail with which to describe the image.

Use of Colors

Relying solely on color to communicate information can make content inaccessible to people with color-blindness.

Hawaiian Diacritical Marks

There is an accepted set of codes that can be inserted into an HTML document to display diacritical marks in words and names in Hawaiian and other languages. Unfortunately, screen readers are tripped up by the ‘okina, or glottal stop (which looks like an upside-down or backwards apostrophe). To deal with this problem, we can use the **`aria-label`** attribute within a span tag:

```
<span aria-label="Hawaii">Hawai&#699;i</span>
```

Skipping Over Navigation Menus

You can put a "skip navigation" link near the top of a page that allows a user of a screen reader to skip over a long navigation menu and go directly to the content of the page. You can do this anywhere you think it might be helpful. To minimize visual impact on your page, you can make these links inconspicuous by making them small and in a color that blends in with the background (as long as they are visible when the receive "focus", as explained below.).

The current recommended method for doing this is to assign an ID name to the object you want to jump to. The link would be directed to that ID name, preceded by a pound sign.

The link would be coded as:

```
<a href="#theimportantstuff" class="skiplink">skip navigation</a>
```

The destination of this link would be coded as:

```
<p id="theimportantstuff">The actual page content starts here...</p>
```

In the example above, the link has been given a class called "skiplink". In your stylesheet, skiplink can be styled as follows to produce a link that normally blends in with the background, but becomes visible when hovered over, and displays

prominently when it has focus:

```
.skiplink a:link, .skiplink a:visited {font-size:1px;color: rgba(255, 255, 255, 0.0)}
.skiplink a:hover {color: blue}
.skiplink a:focus {font-size:16px;color:black;padding:5px;background-color:white}
```

Focus

Things that can accept user input on a web page can have a state known as "focus". These can be things like links, buttons, form fields, and other designated objects. Only one object has focus at a time. You can use the Tab key to step through them, generally in the order in which they occur in the page code (or Shift-Tab to step through them in the other direction). This is important for people who have difficulty using a mouse, and have to rely on the keyboard or other adaptive device.

You can control how something that has focus appears on the screen. By default, browsers will usually indicate focus with a faint outline around the object. Some platforms or designers disable this for aesthetic reasons, but it's important to be able to see which object has focus. If you have put hidden navigation links on your page to help screen readers, you should make sure that when these links have focus, they are displayed visibly. In your stylesheet, give a conspicuous appearance to the "a:focus" state.

Linking to Other Pages

It is recommended that you make the wording of your links unique and descriptive, so somebody skimming the page for links with a screen reader get useful guidance as each link is read out. Avoid links that just say "click here".

Your links should be styled so that they don't just rely on color to indicate that they are links. The default use of underlining might not be pretty, but it does help people spot them.

It is also recommended that when you link to other pages within your website, that you open the new page in the same tab. Having pages open in new tabs makes it harder to trace your steps backward if you can't see the tabs or use a mouse.

Scripts: Useful, and Potentially Problematic

A scripting language like JavaScript can add functionality to a web page. But there are accessibility concerns to be aware of:

Scripts That Cause Something on the Screen to Change
: Use the aria-live attribute, which can tell the browser that a particular part of the screen might change, and specify what to do when it does.

Controls That Are Only Workable with a Mouse
: Just about anything on the screen can be made to trigger some action when you click on it, but many of these objects cannot be activated with the keyboard. Adding the **`tabindex="0"`** attribute can allow the object to receive keyboard focus.

Scripts That Do Something on a Timed Basis
: Ideally, the user should be able to pause or alter the timing of an action that occurs on a timed basis. This can be a difficult thing to deal with, especially if you're trying to put a web widget on your page that you have little or no control over.

Forms

Web forms are often laid out in a manner in which the labels are to the right of the thing being labelled, which is understandable visually, but confusing when a screen reader reads the form from left to right. Therefore, form elements should be linked to their labels using the **`<label>`** tag or the **`aria-label`** and **`aria-labelledby`** attributes.

Using JavaScript to manage form data can cause accessibility problems. You should test forms to make sure you can fill them out using only the keyboard.

For more information, see http://webaim.org/techniques/forms

Flashing , Flickering, or Strobing Images

Graphic images or videos that flash prominently at a rate of 2 or 3 times per second or faster can cause seizures in those with photosensitive epilepsy, especially if the flashing contains the color red. Also be aware that some moving images can induce nausea and dizziness.

PDF Files

Like an HTML document, an accessible PDF file has its content invisibly marked to indicate the function of the different parts of the document. This marking can be generated within Adobe Acrobat Pro. You can also save a PDF file in a program that allows you to mark up a document, such as Microsoft Word using its Styles function to define and apply heading styles and its Format Picture function to apply alternative text to pictures.

Older PDF files and PDFs made from scanned images that don't have their words encoded as text are not accessible. A quick test is to click and drag your mouse across the text to see if you can highlight it; if you can't, it's not encoded as text.

For more information, see this document on PDF accessibility from Adobe.

YouTube Closed Captioning

YouTube now attempts automatic captioning on all uploaded videos. Once you upload a video, allow time for processing (it could take hours for a longer video). The captions will be listed in the Subtitles and CC interface if YouTube was able to generate captions. You can then edit them to correct errors, adjust timing, and make other refinements.

You may find that the "Transcribe and Auto-Sync" option is easier to use. You can prepare the caption text in a word processor and copy and paste it into the text entry box; the timing of the captions will be set automatically. Unpublish the automatic captions so that they are not presented to viewers as an option.

You can add additional text to describe other sounds (example – [phone ringing]) or to identify who is speaking.

Providing Alternate Web Pages

Sometimes the best solution to an accessibility problem is to provide an alternative web page that provides the same information or function in a different manner. Sometimes even when you solve the problems, the solution is not as user friendly as having something presented in a more straightforward layout.

Sample Web Page Code

An example of HTML markup, language declaration, a link to an external stylesheet, a skip navigation link, a list used for website navigation, and a graphic image with alt and longdesc attributes.

```
<!DOCTYPE html>
<html lang="en-US">
<head>
<link rel="stylesheet" type="text/css" href="mysite.css" media="screen" />
</head>
<body>
```

```
<h1>Title of My Document</h1>
<a href="#pagecontent">skip navigation</a>

<ul>
<li><a href="page1.html ">Menu Item 1</a></li>
<li><a href="page2.html ">Menu Item 2</a></li>
<li><a href="page3.html ">Menu Item 3</a></li>
</ul>

<h2 id="pagecontent">The First Section</h2>
<p>This is the content of this section.</p>
<p>This is more content of this section.</p>

<h2>The Second Section</h2>
<p>This is the content of this section.</p>
<p>This is more content of this section.</p>

<img src="infographic.jpg" alt="accessibility infographic" longdesc="infographictext.html"  /><br />
<a href="infographictext.html">Accessible text</a>

</body>
</html>
```

Rev. Nov. 20, 2017