Introduction to Best Practices for Web Accessibility

Greg Vogl
Academic Computing and Networking Services
April 28, 2017
Why is Web Accessibility Important?

• Required by many organizations, including CSU
  – federal and state laws and/or organizational policies

• Basic human right
  – everyone should be able to access information on the web

• Social inclusion, non-discrimination, equality
  – regardless of ability, age, location, language

• Business case, overlap with other best practices
  – usability, mobile-friendly, device independence, search engine optimization (SEO), performance

• W3C: Why: The Case for Web Accessibility
Laws, Policies, Standards, Guidelines

- Americans with Disabilities Act (ADA) – laws
- Section 508 – recommendations for complying with legal requirements
- Web Content Accessibility Guidelines (WCAG) 2.0 – detailed guidelines
- CSU Policy: Accessibility of Electronic Information and Technologies
## Disabilities

<table>
<thead>
<tr>
<th>Category</th>
<th>Examples of Disabilities</th>
<th>Examples of Needs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Visual</td>
<td>Blind, low vision, color blindness</td>
<td>Screenreader, keyboard navigation, enlarged content, high contrast</td>
</tr>
<tr>
<td>Audial</td>
<td>Deaf, hearing impaired</td>
<td>Video captions, audio transcripts</td>
</tr>
<tr>
<td>Mobility</td>
<td>Limited motion, coordination</td>
<td>Keyboard navigation, large controls</td>
</tr>
<tr>
<td>Cognitive</td>
<td>Learning, memory, dyslexia</td>
<td>Simple content, layout, navigation</td>
</tr>
</tbody>
</table>
Content

• Be brief and minimalist, keep only what’s necessary
• Use structure, including logical outline, introduction and summary
• Use direct instructions, positive terms, active voice, strong verbs
• Avoid jargon, acronyms and abbreviations; use familiar words and short sentences
• Write for the target audience; check the reading level
• WebAIM: Writing Clearly and Simply
Visual Design

• **Fonts**
  – Use few, easily readable font faces, with large enough, relative, easily distinguishable sizes.

• **Color**
  – Don’t rely on color or style alone to convey meaning.

• **Contrast**
  – Provide good contrast between background and text colors, and between link text and body text.

• **Coding**
  – Use templates and external CSS for consistent formatting, layout, fonts, colors, and white space.
Examples of Accessible Pages

- ACNS Accessibility Example Web Page
- WAI Before and After Demonstration
- University of Washington Accessible University
Navigation

• Navigation should be simple and consistent.

• Navigation links should come after the main content.
  – Or provide a link or method that allows users to skip repetitive navigation links.

• Display some content only to screenreaders.
  – Use absolute positioning off the screen instead of display:none or visibility:hidden.

• Links should describe the linked page and make sense if read out of context.
  – Avoid "click here" or "more"

• Buttons and tabs should be large enough for easy use.

• WebAIM: Links and Hypertext
HTML Structure

- Helps screenreaders and keyboard navigation (accessibility) and web crawlers (SEO)
- **Page title**: `<title>` - unique, descriptive
- **Headings**: `<h1>`-`<h6>` - properly nested
- **Emphasis**: `<strong>`, `<em>` - use instead of underlines or styles
- **Lists**: `<ul>`, `<ol>`, `<dl>`, `<li>` - unordered or ordered lists, navigation menus
- **Frames**: `<frameset>`, `<frame>`, `<iframe>` - only if necessary
- **HTML5**: `<header>`, `<footer>`, `<nav>`, `<article>`, `<section>`, `<aside>` - semantic elements
WAI-ARIA Landmarks

- Help users navigate rich Internet applications
  - Page structure
  - Scripted and dynamic content (JavaScript, AJAX)
  - Interactive controls and complex widgets
  - Supported by most screen readers and browsers
- Attributes (often not needed for HTML5-specific tags):
  - role: main, navigation, search, banner, article, contentinfo
  - aria-required, aria-disabled, aria-readonly
  - aria-label, aria-labelledby
- WebAIM: ARIA Landmarks
Tables

- Data tables: summary="…", <caption>, <th>, scope="col"
- Layout tables: role="presentation"
- Keep tables simple; minimize nested tables and cells spanning rows and columns
- [WebAIM: Creating Accessible Tables](https://webaim.org/techniques/tables/)

[COLORADO STATE UNIVERSITY](https://www.coloradostate.edu)
Images

- **alt** attribute: used to describe the content and function of images, image inputs, image map areas, and Java applets.
  - HTML code: `<img src="/images/morgan.jpg" alt="Morgan Library" />`
  - An image which is not a link, and merely decorative or explained in the page text, should use an empty alt attribute: `alt=""`
- Image maps: use client side rather than server side.
- Movement and animation (graphics or text): use sparingly or consider alternatives.
- [WebAIM: Accessible Images](#)
Multimedia

- When using PDF documents and Power Point presentations, ensure that they are accessible.
  - Also provide the information in HTML format. (See Adobe accessibility information.)

- When a plug-in, player, viewer or application is required to view essential content,
  - Provide a link to download the plug-in, such as Adobe Reader for PDF files.
  - If you are unsure if the plug-in meets accessibility standards, provide the information in an alternative format.

- Provide access to multimedia for users with visual or hearing disabilities
  - video captioning, audio transcripts, and audio descriptions of visuals

- Manual controls for multimedia (such as video, audio, and image carousels) should not require using a mouse.

- CSU Accessibility Guidelines: Creating Accessible Multimedia
Online Forms

• **Easy** to navigate, complete, and submit
  – **Instructions** positioned before the form elements; clearly identify required fields
  – **Keyboard** for navigation (e.g. Tab and arrows to move, Enter to submit) and logical tab **order**
  – **Label** form fields (e.g. text, dropdowns): `<label for="name">Name:</label><input type="text" id="name" name="name" />
  – **Group** related fields with `<fieldset>` and `<legend>` tags

• **Error handling**
  – **Validation**: both client and server
  – Error **messages** that clearly indicate the problem and solution
  – Error **correction** should be easy
  – **Alternative** way of providing the information requested, e.g. email

• **WebAIM: Creating Accessible Forms**
Accessibility Testing: Text to Speech

- **Screenreaders**
  - JAWS – Windows, commercial, CSU license
  - NVDA – Windows, free
  - VoiceOver – Mac, free (pre-installed)
  - ChromeVox – Chrome extension

- **Text to Speech readers**

- **WebAIM: Designing for Screen Reader Compatibility**
Accessibility Testing: Web Browsers

- Keyboard-only navigation (keyboard focus, tab order, forms, dialog boxes, multimedia controls, navigation)
- Disable images, styles, scripts (using Web Developer extension)
- Change zoom level (everything or text only)
- Different browsers, operating systems, mobile devices
Accessibility Testing: Automated Tools

- WebAIM WAVE
- Other tools
- Related automated checks
  - code validation (especially HTML), broken links, spelling, readability, mobile, performance, security
Best Practices: Process

- **Knowledge**
  - regulations, standards, code, tools, techniques

- **Holistic approach to design, development and testing**
  - content, visual design, navigation, coding standards, usability, accessibility, security, performance, analytics, SEO, etc.

- **Organizational strategy**
  - policies, plans, procedures
  - site testing at launch time, site audit periodically (see example checklist spreadsheet)

- **Human support**
  - training and resources for developers and content providers
  - accessibility information and contact for end users
  - example: CSU Libraries: Website Accessibility
More Information

- CSU ATRC: Accessibility Resources for Web Developers
- CSU ACNS: Making Accessible Websites
- Web Accessibility in Mind (WebAIM.org)