Detailed Tips on Web Accessibility

By Rhiannon Dixon and Celeste Ramos
Important Accessibility Resources

• The Web AIM Color Contrast Checker
  • https://webaim.org/resources/contrastchecker/

• WAVE Accessibility Tool
  • http://wave.webaim.org/

• Adobe Acrobat DC Accessibility Tool for PDFs
  • https://helpx.adobe.com/acrobat/using/create-verify-pdf-accessibility.html#check_accessibility_of_PDFs

• Technology Accessibility
  • http://accessibility.ua.edu/
The Basics

• Link Dos and Don’ts
• Phrases to Avoid
• iFrames
• Images
• Color Contrast
• Headers
• Table Structure
• Menu Structure

• Creating “Skip Nav”
• ARIA
• PDFs
• Forms
• Justified Text
• WebAIM
• Apps to Avoid
Links

• Don’t do this:
  • Visit the library website-- https://www.lib.ua.edu/#/home

• Do this whenever possible:
  • Visit the library website

• Why?
  • Because most screen readers only read the anchor text (the part highlighted here in blue and underlined). It is more helpful to know where the link will take me then to hear the exact URL of the link (i.e. h-t-t-p-s:-/-/-w-w-w-w-.-, etc)
  • Short action phrases are the best!
Links Don’t Need Titles

• `<li><a href="/about/libraries-policies/photography-policy-for-university-libraries/" title="Photography Policy for University Libraries">Photography Policy for University Libraries</a></li>`

• Redundant and unnecessary
More Link Tips

• Absolute No’s:
  • Click Here
  • Read More
  • More
  • More Information Here
  • Read More Here

• Instead, use a description of the page or information that you are linking to! Examples:
  • PDF on How to Surf.
  • More information can be found using Google’s Search Engine.
The Pains of “Read More Here”

• Why is “Read More Here” and its ilk an absolute no?
  • Because when someone with a screen reader is navigating through a web page, all that the screen reader reads is “Here.” This is unhelpful because “Here” could mean anything!
iFrame/Videos

• Always add a Title attribute to iFrames!

• Example:
  • `<iframe width="320" height="240" align="left"
    src="https://www.youtube.com/embed/ChKwlUhx_ic" frameborder="0"
    allowfullscreen="allowfullscreen" title="Ted Talk on 3D Printing" ></iframe>`
Images

• Always add alt text! Example:
  • `<img src="https://blablahblah.jpg" alt="Da-Vinci" width="140" class="aligncenter" style="margin-top: 20px;"/>

• If your image is just a picture of words, like this...

![Creating Sentences](image)

• Change it to actual text!
Why are images of text bad?

• Image alt text is meant to be concise and succinct. With most images of text, the text is too much to all be placed in alt text.
• Most screen readers do not have the capability to read text that is an image.
• However, all screen readers can read normal text.
• Image with text = Building without an elevator
Decorative Images

Does not need alt text!

E.X = alt="" (Do not add a space inside the quotation marks)
Making Images Links

• Make sure to have Alt text with images that are links that explain what the link is
  • Ex. Alt= “University Libraries Home”
Headers and Images

• DO NOT have headers and images with the same link.
  • Creates a redundant link
  • Confuses the user
• DO connect the link with the header.
  • Screen reader will not reader both the text and image
Header/Image Example

<a href="http://alabamamaps.ua.edu/contemporarymaps/campus/3D%20Campus%20Map%202006-18-14.jpg">
  <h3>Alabama Campus Map</h3>
  <img src="http://alabamamaps.ua.edu/contemporarymaps/campus/3D%20Campus%20Map%202006-18-14.jpg" alt=""/>
</a>
Color Contrast

- This is bad. Can you even read this?
- This is still pretty bad.

Yay for UA, this is a pass!
If you’re unsure, use https://webaim.org/resources/contrastchecker/ to check!

- Level AA requires a contrast ratio of 4.5:1 for normal text and 3:1 for large text. Large text is 18 point (24px) or larger.
- A good rule of thumb is to avoid layering a dark color on top of the same color in a lighter shade, or vice versa. Dark gray on top of a light gray background is always iffy.
Headers

• Different Heading Elements:
  • <h1>Types of Dog</h1>
  • <h2>Golden Retrievers</h2>
  • <h3>Favorite Foods of the Golden Retriever</h3>
• The first header on a page should always be H1!
• Because the WordPress title shows up as H1, on library web pages your content should start with H2
• Although sighted users only see that different heading levels are different sizes, screen readers announce heading levels. If a web page has one <h1> header and three <h3> headings, non-sighted users are led to believe that their screen reader is failing to them h2 content.
Header Tricks

If you want a header to be a certain size while following accessibility requirements, follow the example below:

```
<h1 class="h3">Lorem ipsum</h1>
```

In this example, the “h1” is what the screen reader will notice and read out. However, the header will take on the appearance and size of an “h3.”
Table Structure

• In order for an HTML table to be accessible, scope must be defined and tables must be complete.
• Including <th> is not enough, an accessible table header must also have scope=’col’ for each table header and scope=’row’ at the beginning of each row of <td> information.
### Table Structure Example

- `<table>`
  - `<tr>`
    - `<th scope="col">Name</th>`
    - `<th scope="col">Birthday</th>`
    - `<th scope="col">Favorite City</th>`
  - `</tr>`
  - `<tr>`
    - `<td scope="row">Marvin Gardens</td>`
    - `<td>July 25<sup>th</sup></td>`
    - `<td>New York</td>`
  - `</tr>`
- `</table>`
Accessible Rich Internet Applications (ARIA)

Enables navigation landmarks, widgets, form hints, error messages, and more

Components:
• Roles
• Properties
• States
ARIA Components

Roles - defines what an element does
  • <form role="search">

Properties - defines properties or meanings of elements
  • <input aria-required="true">

States - defines current condition of an element
  • <input aria-disabled="true">
In general, accessibility issues with rich internet applications can be characterized as:

- Inability to provide the semantic structure of page areas and functionality (e.g., navigation, main content, search, etc.)
- Inaccessibility of content that is dynamic and may change within the page (e.g., AJAX content updates)
- Inability to change keyboard focus to page elements (e.g., setting focus to an error message within the page)
- Difficulty with keyboard and screen reader accessibility with complex widgets and navigation elements (e.g., sliders, menu trees, etc.)

ARIA can help address many of these issues.

(https://webaim.org/techniques/aria/)
PDFs

• It’s hard to make a truly accessible PDF, which is why they are often flagged as accessibility warnings. An accessible PDF:
  • Is tagged
  • Does not have color contrast errors
  • Has set language and a set title
  • Includes alt text for images
PDF Accessibility Checker

- Open Adobe Acrobat DC
- Open a PDF you want to check by double-clicking the name within Adobe Acrobat DC or pull it up separately
- Click on the “Tools” bar (upper left)
- Scroll down to the “Protect and Standardize” section, and find the Accessibility icon (a purple person)
- Go through adding the Accessibility tool, and open it. Now, when looking at your PDF there should be a list of Accessibility resources on the right, including “Autotag Document” and “Accessibility Report”
• Usually the best first step is to run an Accessibility Report and make changes based on the recommendations.
• If the document is untagged, autotag it.
• If the title is not set, go to File → Properties → Description tab to add a title and author, and then go to the Initial View tab to change the “Show:” option from “File Name” to “Document Title.”
• If the language is not set, go to File → Properties → Advanced tab in the “Reading Options” Box to set the document language.
Creating “Skip Nav”

- Typically created by a web developer
- “Skip Nav” is a tool that allows keyboard users to skip directly to page content and bypass the menu.
- “Skip Nav” can be implemented in many ways, but most developers prefer to have it be out of sight unless it receives tab focus. “Skip Nav” should always be at the top left of the page and occur before the menu.
- Imagine that you have to tab through every single item on the web page menu before getting to the content. What if you visit this website a lot? Tabbing through the menu every time is redundant and frustrating.
More on “Skip Nav”

• The “Skip Nav” we’re demonstrating uses a combination of HTML and CSS to stay hidden until it receives user focus.
• HTML:
  • `<div id="skip"><a href="#content">Skip to Main Content</a></div>`
The CSS of “Skip Nav”

//CSS for the "Skip to Main Navigation" Link for accessibility purposes
//This hides the Skip Nav link to anyone using a mouse
#skip a {color:white;
    position:absolute;
    left:-1000px;
    top:auto;
    width:1px;
    height:1px;
    overflow:hidden;}

//This ensures when Skip Nav menu has tab focus, it will appear at the top of the page
#skip a:focus{
    position:static;
    width:auto;
    height:auto;}
}
Forms

• An accessible form includes:
  • Labels (carefully checking to make sure each element has a unique id and label)
  • Fieldsets (for radio buttons or groups of checkboxes)
  • Legends (like a label heading, but for fieldsets)
Forms

• When making an online form, best practice is to give all elements labels.
• Put all questions that have radio buttons, checkboxes, or multiple choice answers within a `<fieldset>`.
• The question in a fieldset is a `<legend>` element, so the question does not need to be labelled within a fieldset.
• The answers within a fieldset should still be labelled.
Form Example Beginning

<div class="row">
  <div class="col-md-12">
    <div class="form-group col-md-4">
      • Fieldset/question goes here
    </div>
  </div>
</div>
Form Example Continued

<fieldset>
  <legend>When does your class meet?</legend>
  <div id="field5">
    <div class="checkbox">
      <label><input type="checkbox" name="ClassDays" value="MWF"> MWF</label>
    </div>
    <div class="checkbox">
      <label><input type="checkbox" name="ClassDays" value="MW"> MW</label>
    </div>
    <div class="checkbox">
      <label><input type="checkbox" name="ClassDays" value="TR"> TR</label>
    </div>
  </div>
</fieldset>
Justified Text

• Justifying text is unaccessible. Why?
  • Because fully justified text makes it harder to tell where the edges end when scanning a page that is zoomed in
  • Because “kerning” (the varied white space between letters to make text justified) makes the text harder to read and understand
  • Especially a problem for users with dyslexia
Web AIM Testing

• One of the most useful accessibility tools!
• Web AIM plugin allows you to test sites that aren’t public yet
• WAVE will tell you exactly what spots on the web page need correction, which are most urgent, why to correct, and how to do it.
• The WAVE Accessibility Checker (http://wave.webaim.org/)
  • Put any web page address into the textbox and press enter, or click the arrow button
  • The “Styles” tab summary to the left of the page shows in detail any accessibility alerts and their severity.
  • The “Contrast” tab summary shows color contrast issues
Apps to be Wary of

The following apps are nearly always inaccessible (as of October 2017) and should be avoided completely:
  • Carousels (image slideshows) of any kind
  • Google Maps
  • Twitter Boxes
  • Any Social Media Boxes