



# Website Accessibility Testing

Todd M. Weissenberger, University of Iowa

The World Wide Web and related technologies have changed the way people learn, communicate, and engage with the world around them. To ensure that the web benefits all people, it is important to maintain websites that are usable and accessible to all.

Here are a few tips and tricks to help you to test your site’s accessibility, so that you can create and maintain more accessible websites for your students and others.

Overview .....	3
Site Testing Tools .....	4
Landmark Testing.....	5
ALT Text/Text Alternative Testing.....	6
Heading Level Testing .....	7
Keyboard Accessibility Testing.....	8
Color Contrast Testing .....	9
Forms Testing.....	10
Data Tables Testing.....	11
Document Level Accessibility Testing .....	12
Hyperlink Testing .....	13
Notes.....	15



## Overview

---

A website accessibility test is a means of evaluating the accessibility of your website. Because many accessibility principles reflect standard design processes and practices, site accessibility assessments can provide value beyond WCAG compliance.

A site accessibility assessment can take place at any point in a website's lifecycle, but may be most useful in the early phases of the design process, when it is relatively easy to develop and manage site features at a global level. Early implementation of accessible development principles can significantly increase access and improve usability for all users. After deployment, site assessment can play a role in identifying maintenance and management tasks, and determining when the site is due for tweaks, high-level maintenance, or a complete reboot.

This document offers a list of tools and techniques for evaluating and improving the accessibility of your website. These suggestions are derived from a variety of well-known sources and describe practices commonly associated with accessibility efforts. Designers, developers, and site owners are encouraged to explore other accessibility resources, as well.

For more information on I.T. accessibility efforts at Iowa, please visit <http://itaccessibility.uiowa.edu>.

### *Testing Methodology*

Testing methodology is determined by the unit webmaster and site owners. Recommended testing issues include, but are not limited to:

- ALT text: alternate text for images and other non-text elements
- Document structure: properly nested heading elements, page title, human language
- Tables: proper table structure and relationships between table components
- List structure: lists are created using semantic lists structures
- Forms labeling: label or title elements for all interactive form fields
- Hyperlinks: link text is unique and descriptive; user is informed when links open in new windows
- Keyboard accessibility testing: all content and interactivity are available via keyboard-only input
- Color contrast: sufficient color contrast for low-vision and color-blind users

An automated site assessment, using the Functional Accessibility Evaluator or a similar tool, can also expose opportunities to improve accessibility prior to a University-sponsored assessment.

The techniques described here are not exhaustive, but should help you to understand key accessibility issues that may affect your site.

### *Conformance*

- The University of Iowa standard for web accessibility is Web Content Accessibility Guidelines, version 2.0, Level AA (WCAG 2.0).

## Site Testing Tools

---

A variety of tools are available to assist with accessibility evaluation efforts. The IT Accessibility group at Iowa maintains this list as a service to developers and other stakeholders; please contact [ITAccessibility@uiowa.edu](mailto:ITAccessibility@uiowa.edu) to make suggestions or report changes to items listed here.

### *Online Assessment Tools*

- Functional Accessibility Evaluator  
<http://fae.illinois.edu>
- WAVE Online Tool  
<http://wave.webaim.org>

### *Accessibility Tools and Extensions*

- AInspector WCAG (Firefox)  
<https://addons.mozilla.org/en-US/firefox/addon/ainspector-wcag/>
- WAVE Browser Extensions (Chrome, Firefox)  
<http://wave.webaim.org/extension/>
- Web Developer Extension (Chrome, Firefox, Opera)  
<https://chrispederick.com/work/web-developer/>
- aXe Accessibility Extension (Chrome, Firefox)  
<https://www.deque.com/axe/axe-for-web/>
- Visual ARIA Bookmarklet  
<https://whatsock.com/training/matrices/visual-aria.htm>
- Web Accessibility Toolbar for Internet Explorer (Internet Explorer)  
<http://www.paciellogroup.com/resources/wat/ie>
- Colour Contrast Analyser (Stand-alone tool)  
<http://www.paciellogroup.com/node/18?q=node/20>

### *Assistive Technology/Screen Readers*

- NVDA (Non-Visual Desktop Access)  
<http://www.nvaccess.org>
- JAWS (Job Access With Speech)  
<http://freedomscientific.com>
- VoiceOver (iOS)  
<https://support.apple.com/guide/iphone/turn-on-and-practice-voiceover-iph3e2e415f/ios>

While these tools are a useful part of an accessibility testing strategy, automated tools are limited in their capacity and scope. Testers should also consider manual and user testing along with a range of heuristics to assess the functionality of their sites.

## Landmark Testing

---

### *Definition*

ARIA landmark regions label, define, and delineate primary regions of a web page, making it simpler for users to traverse a page, section by section.

### *Why is this an accessibility issue?*

Content that lies outside a landmark region may be overlooked by users of assistive technology. All content in every web page should be contained within an ARIA landmark region. This includes content that is located off-screen, content that is generated dynamically, and other content that might reside outside of the primary scope of a web page or document.

### *Testing Methodology*

1. Ensure that all content is contained within an HTML5 or ARIA landmark
2. Ensure that HTML5 or ARIA landmarks use valid roles that are not duplicated or overridden
3. Ensure that HTML5 or ARIA landmarks include unique, descriptive names
4. Ensure that HTML5 or ARIA landmarks observe all applicable rules for landmark usage

### *Tools*

- AInspector WCAG (Rule Category: Landmarks)
- Accessibility Bookmarklets (Landmarks)

### *Landmark Roles*

The following HTML5/ARIA landmark roles describe page regions, for ease of page navigation and traversal by users of assistive technology.

- main (HTML5: `<main>`) **[This landmark is required in all pages]**
- navigation (HTML5: `<nav>`) **[This landmark is required in all pages]**
- banner (HTML5: `<header>`) **[No more than one is permitted per page]**
- contentinfo (HTML5: `<footer>`) **[No more than one is permitted per page]**
- complementary (HTML5: `<aside>`) **[Restrictions on child regions]**
- form (HTML5: `<form>`) **[Restrictions on child regions]**
- search (no HTML5 equivalent) **[Restrictions on child regions]**
- region (HTML5: `<section>`) **[Must have accessible name]**

### *Conformance*

HTML5 and ARIA landmarks are likely to meet accessibility criteria if:

- All content in a page is contained in an appropriate landmark;
- Landmarks include unique and descriptive accessible names;
- Landmarks conform to level, content, and nesting requirements;
- No other landmark-specific barriers are present

## ALT Text/Text Alternative Testing

---

### *Definition*

Alternative text accessibility means that when non-text elements (e.g., images, form controls, embedded objects, multimedia, players, etc.) are unavailable, they are replaced by alternative text that conveys the same information and provides the same functionality as the non-text element.

### *Why is this an accessibility issue?*

Many users are unable to perceive graphic images that are embedded in a web page. For these users, the information and function of those images is lost unless an alternate representation is provided.

### *Testing Methodology*

1. Ensure that all <img> elements include an ALT attribute
2. Ensure that ALT attributes provide content and functionality (i.e., linking, focusability) equivalent to the <img>element
3. Ensure that background images are not used for content or functionality
4. Ensure that images used for decoration only include empty ALT text (alt="")
5. Ensure that all embedded objects (e.g., multimedia, applets, Flash components, etc.) include appropriate alternative text when those objects are not available
6. Ensure that long text descriptions are provided for non-text elements that require detailed explanation
7. Ensure that pages with visual CAPTCHAs include CAPTCHAs that use a different modality

### *Tools*

- AInspector Sidebar (Rule Categories: Images)
- WAVE Toolbar (Errors, Features, and Alerts)
  - Places overlay on current page, revealing images without ALT text
- Web Developer Toolbar (Images/Disable Images/All Images)
  - Hides embedded images, revealing ALT text
  - Hides background images

### *Conformance*

Non-text elements are likely to meet accessibility criteria if:

- All information contained in the non-text element is perceivable by the user
- All functionality contained in the non-text element is operable by the user
- Decorative images contain empty ALT text or role="presentation"

## Heading Level Testing

---

### *Definition*

Headings 1-6 should be used to provide a hierarchical structure to web documents. Each major section or topic within a document should be introduced with an HTML heading, and headings should be properly nested. No levels should be skipped in a heading sequence: for example, <h2> is an appropriate child of <h1>.

### *Why is this an accessibility issue?*

Users of screen reader technology often rely on heading levels to determine the outline of a page. A well-developed page outline helps screen reader users to navigate quickly through sections of a site to reach the content they seek.

### *Testing Methodology*

1. Ensure that pages include a top-level heading (h1) in the main landmark region;
2. Examine pages for logical and hierarchical heading structure;
3. Determine whether any logical heading levels are skipped;
4. Determine whether headings are used for stylistic, rather than structural purposes

### *Tools*

- AInspector Sidebar (Rule Categories: Headings)
- WAVE Toolbar (Outline)
  - Reveals document outline; highlights missing headings
- Web Developer Toolbar
  - Reveals document outline; highlights missing headings

### *Conformance*

A website or application is likely to meet structured heading standards if:

- All page sections begin with a heading;
- All headings are nested appropriately, according to heading level;
- Headings are not used for formatting, sizing, or other stylistic purposes

## Keyboard Accessibility Testing

---

### *Definition*

Keyboard-only access means that all site navigation and functionality are available using only the standard keyboard, and that the user can move freely through the page using only the standard keyboard without becoming caught in a "keyboard trap" (in which the cursor is "trapped" in one section, widget, or functional region of the document, and cannot move to another section).

### *Why is this an accessibility issue?*

Some users find it difficult or impossible to use a mouse, often due to visual or motor impairments. For these users, the keyboard provides a reliable means of interacting with the computer. If the user can't operate a site with the keyboard, they can't operate it at all.

### *Testing Methodology*

1. Access all site navigation, form controls and other functionality on a device with no mouse;
2. Determine that all content and function can be keyboard controlled;
3. Visually confirm that objects indicate focus with a visual cue;
4. Ensure that no keyboard traps exist to isolate the user in one area of the interface;
5. Review any custom key commands for conflict with the browser or operating system

### *Tools*

- AInspector Sidebar (Rule Category: Keyboard)
  - Reveals machine-readable issues with keyboard access
- Keyboard (View page with mouse disconnected)
  - Allows tester to tab through page to determine accessibility
- Web Developer toolbar (CSS/View Style Information)
  - Tester can select an element and check CSS (e.g., for `:focus` pseudo-class) in Style Information tab
  - Reveals skip links that may be positioned off-screen

### *Conformance*

A web site or application is likely to meet keyboard accessible criteria if:

- All navigation and interface elements are operable through keyboard controls only;
- Headings and focusable elements permit interface navigation for screen readers;
- Elements change visually to indicate focus;
- No keyboard traps are present;
- Accessibility key commands do not conflict with key commands in the native application;
- No other keyboard-specific barriers are present



## Color Contrast Testing

---

### *Definition*

Sufficient color contrast means that for text content, all foreground/background color combinations provide sufficient contrast for low-vision users or users with color blindness.

### *Why is this an accessibility issue?*

Some people are unable to distinguish between colors at certain parts of the color spectrum. If content doesn't have sufficient contrast, it can be difficult or impossible for the user to tell the foreground from the background, rendering the content illegible.

### *Testing Methodology*

1. Document background/foreground color combinations in the application;
2. Using a color contrast tool, evaluate the color combinations for compliance, OR
3. Calculate color contrast

### *Tools*

- AInspector Sidebar (Rule Category: Styles/Content)
  - Identifies insufficient contrast between foreground and background elements
- Colour Contrast Analyser
  - Evaluates contrast between two colors (eyedropper or HEX values)

### *Conformance*

A website or application is likely to meet color accessibility criteria if:

- All background and foreground color combinations contrast at a 4.5:1 ratio for 14 point text or smaller;
- All background and foreground color combinations contrast at a 3:1 ratio for text larger than 14 points
- Color is not the only visual feature used to identify or distinguish objects or text

## Forms Testing

---

### *Definition*

Accessible forms are fully keyboard-operable, with well-labeled fields that change visually when they receive focus. Forms also should include detection and recovery mechanisms that alert the user of invalid input, required fields, and other changes in context.

### *Why is this an accessibility issue?*

Some users are unable to see the text that accompanies most form fields. Without the visual text, it is impossible to tell the purpose of a form field unless there is a label that a screen reader can associate with the field.

### *Testing Methodology*

1. Keyboard-operability: complete and submit the form without using a mouse or stylus;
2. Labeled fields
  - Click or send focus to a field's corresponding <label> element; the field should take focus, OR;
  - Evaluate form <label> items using the Forms tool in the Web Developer toolbar; OR
  - Fields without <label> tags use the TITLE attribute to provide contextual information
3. Review error detection and recovery mechanisms to ensure that:
  - Cursor focuses to the message in the alert mechanism;
  - The user can navigate by keyboard within the alert interface;
  - When the user closes the alert mechanism, focus returns to the point of the error, or another appropriate location within the document

### *Tools*

- AInspector WCAG (Rule Category: Forms)
  - Lists unlabeled or mislabeled form controls
- Web Developer Toolbar (Forms >> View Form Information)
  - Generates a grid of fields and labels
- WAVE (Errors, Features and Alerts)
  - Reveals unlabeled form controls
- WAVE (Text-Only)
  - Exposes labels within form controls

### *Conformance*

Forms are likely to meet accessibility guidelines if:

- They are fully keyboard-operable
- All fields are accompanied by meaningful <label> elements, or use the TITLE attribute
- Error messages are interactive and navigable for users of screen readers and other AT
- No other barriers exist in the form

## Data Tables Testing

---

### *Definition*

Accessible tables are semantically sound, using <th> and <td> to distinguish between headings and data. Heading and data cells should include SCOPE, and ID or HEADERS to associate data with the correct headings. Tables should also include a meaningful SUMMARY attribute, and a <caption> element to aid in describing the purpose of the table. Tables should be used for data presentation only. If a table layout is necessary, the layout table must be identified as such using the ARIA presentation role.

### *Why is this an accessibility issue?*

Some users are unable to perceive or recall multidimensional information, as in a table. For these users, associating table headers and data cells, as well as summary information, can reinforce the purpose of a table, and help to make sense of the data. Page layouts presented in table format can also present barriers to screen readers and keyboard users.

### *Testing Methodology*

1. Ensure that the <table> tag includes an appropriate SUMMARY attribute;
2. Check for the presence of a <caption> tag inside the <table>;
3. Review row and column headers for use of the <th> tag, with SCOPE attribute;
4. Review data cells for use of the <td> tag;
5. If <th> elements use ID, ensure that <td> uses corresponding HEADERS attribute;
6. Examine the table in a screen reader (or linearize the table) to ensure that table content makes sense when presented linearly;

### *Tools*

- NVDA/JAWS (Screen reader)
  - Yields text-to-speech rendering of table
- Web Developer Toolbar (Information/Display Table Information)
  - Displays table headers and summary
- WAVE Toolbar (Structure/Order)
  - Reveals scope and reading order of table

### *Conformance*

A table is likely to meet accessibility guidelines if:

- The purpose of the table is the tabular presentation of data OR if identified as a layout table;
- Headers and data cells are marked up correctly;
- All of the component parts include appropriate, descriptive attributes;
- The SUMMARY attribute and <caption> tag are present, and descriptive;
- No other barriers are presented by the use of the table

## Document Level Accessibility Testing

---

### *Definition*

Document-level accessibility testing covers a number of issues in the construction of a page at the document level. Document-level tags should be properly nested, and should include specific meta-information to provide context to the user and user agent.

### *Why is this an accessibility issue?*

Heading levels provide a natural outline structure that facilitates page navigation for some users. Headings let the user browse a page's hierarchy until they find the information they seek.

### *Testing Methodology*

1. Ensure that pages begin with the appropriate Document Type Declaration for the HTML version used to author the page
2. Check each page for an appropriate, descriptive <TITLE>
3. Make sure pages contain information required by the HTML version used to author the page
4. Ensure that the <HTML> tag includes a LANG attribute to identify the language of the page
5. Ensure that any changes in human-readable language are indicated by using the LANG attribute in the element that contains the language change
6. Ensure that pages are equally legible and functional when stylesheets are omitted

### *Tools*

- Validate site ([validator.w3.org](http://validator.w3.org))
  - Reveals absence of DTD, TITLE, LANG, CHARSET
- WAVE Toolbar (Disable Styles)
  - Yields visual inspection of page with CSS disabled
- Web Designer Toolbar (CSS/Disable Styles)
  - Yields visual inspection of page with CSS disabled

### *Conformance*

A web page is likely to meet document-level accessibility standards if:

- All of the above criteria are met
- No other barriers exist in the page construction

# Hyperlink Testing

---

## *Definition*

Hyperlinks are the backbone of web and hypertext functionality. Links should be clear and informative, and should include information that assists the user in deciding whether to follow them.

## *Why is this an accessibility issue?*

Users rely on meaningful and descriptive hyperlinks for context, orientation, and navigation. Links that lack this information are confusing and can present a barrier to access for some users. Also, links that specify a single call to action (e.g., "Click Here") may exclude users who employ other means for following hyperlinks.

## *Testing Methodology*

1. Ensure that no links are empty or use background images only for identification;
2. Ensure that all links include a unique, descriptive, accessible name to guide the user;
3. Ensure that adjacent links to the same resources are contained in a single link;
4. Ensure that all links accept keyboard focus and activation;
5. Ensure that all links display a visual indicator when they hold keyboard focus;
6. If hyperlinks open in a new window, ensure that this is communicated to the user;
7. If links are used as buttons, tabs, or for other interactions, ensure that the appropriate role is assigned

## *Tools*

- AInspector Sidebar (Rule Categories: Links)
- NVDA
  - NVDA + F7 (Displays all links in a page)

## *Conformance*

Links are likely to meet accessibility standards if:

- All links include accessible text descriptions or content
- All links are keyboard operable, including focus and activation
- Adjacent text and image ALT within a link does not repeat, or "stutter"
- No other barriers are presented by the implementation of hyperlinks



## Notes

---