



Basic Accessibility Checklist

Accessibility should be considered in degrees of user-friendliness for all audiences. The goal is to provide a similar degree of ease-of-use for everyone. This can be achieved for most web pages by paying attention to some basics. Naturally accessibility becomes more of a challenge the more complex your web pages are but with forethought and observing some general guidelines, most web sites can be greatly improved.

The Evaluation Checklist provides a basic structure for manual testing. While some of the list items overlap issues discovered by automated accessibility checkers, this checklist offers a greater focus on qualitative issues. Paying attention to these details will help you build functionally accessible web pages.

Evaluation Checklist

*Will require examination of the code or use of a tool. See the Tools list at the end of this document.

1. Using *only* the keyboard, not the mouse, can you navigate, operate and interact with all the elements on a web page, (menus, navigation and dynamic objects like slide-shows)? You can use the tab key, space bar, arrow keys and enter/return key. You should not rely on keyboard shortcuts since these can vary based on operating system and may require prior knowledge.
2. When you use just the keyboard, can you easily tell where you are at on the page (visual focus), similar to when you mouse over a link and it changes?
3. The reading and tab order for a web page or application must be visually logical. (You can detect the tab order by watching the focus as you tab through the page. Tab order should not jump around the page.)
4. Are videos captioned? Are the captions accurate? Is a text transcript available so that deaf audiences can understand what's going on?
5. Do form fields and buttons have valid "programmable" labels? This can be determined by clicking on the visible text label, which should move the focus to the corresponding input field.
6. *Do images have descriptive alternative text (ALT). Charts and graphs may also require additional detailed descriptions so that users can understand their purpose.
7. Does the design use good "link text"? Links like "Click here" or URLs are bad. Links should be human-readable and descriptive apart from their context.
8. *Are header tags (h1, h2, h3...) present and being used in sequential hierarchical order? (Header tags should not be skipped over.)
9. *Have tables been used for layout? (Layout tables are always a bad idea, especially if nested.)
10. *Are data tables using accessible markup? (TH, captions, headers/id should be used)
11. *Is the contrast of text adequate for visually challenged audiences to read? Does the text stand out from the background? Low vision and color-blind users may not be able to see important content.
12. Can you still understand and read the page if you switch to high-contrast color mode. (This is primarily for Windows clients.)
13. Does the web page rely on color alone to convey information. At least two design components (text, shape, patterns, color) should be used to convey meaning.

Accessibility Best Practices

The following list provides suggestions for accessibility practices that will make your web sites/applications much easier for all users. These items often overlap with usability and search engine optimization techniques.

- Keep content (text), presentation (style), and function (scripting) separate. Keeping each layer distinct from the other allows the content to always come through in the absence of design or dynamic scripting. This is especially helpful in terms of mobile design, and it also makes maintenance of a site easier.
- Always use ARIA landmark roles. When you use ARIA to mark off regions of the page, be sure to encompass everything on the page, don't leave anything outside an ARIA landmark role. See the Resources section below for helpful reference pages on ARIA.
- Use HTML elements in a way that matches their semantic meaning. For instance, don't use header markup (h1, h2, h3) simply because it's large, bold text. Assistive technology relies upon the default semantic meaning that is associated with HTML elements.
- The h1 tag is reserved for denoting the page's purpose. It should match all or part of the page's title tag. Subsequent headers like h2, h3 and so on should be used for headlines or to mark a section.
- Use lists to group similar information, such as links in navigation menus. Lists promote "chunking" of information, which is easier for users to consume, and they preserve semantic associations for blind users. Lists are well supported with assistive technology.
- Use fieldsets and legends when organizing forms. Fieldsets, similar to Lists, are a great way to group similar information, like demographics or preferences. Legend text provides a label for the fieldset and should be unique.
- Link text must make sense when read by itself. Do not use link text like "Read more..." or "Here" or a URL.
- Do not use tables for layout. If it's impossible to avoid them, then make sure the information within the table will linearize logically and is marked, `role="presentation"`. Do not nest tables (placing a table within a table).
- Avoid using CSS sprites and background images for controls, such as buttons, where a visible text alternative is not present. High-contrast modes on some platforms turns these images off.

Resources

- Tools**
- Functional Accessibility Evaluator (FAE) 2.0 - (fae.disability.illinois.edu/)
 - WebAIM Wave - (wave.webaim.org/)
 - Accessibility Bookmarklets (accessibility-bookmarklets.org)

- Plugins**
- AInspector WCAG - (addons.mozilla.org/en-US/firefox/addon/ainspector-wcag/)
 - Web Developer 2.0.1 (chrispederick.com/work/web-developer/)

- References**
- Technology Services Accessibility Examples (techservicesillinois.github.io/accessibility/)
 - WAI-ARIA Authoring Practices 1.1 - (www.w3.org/TR/wai-aria-practices/)
 - Illinois Information Technology Accessibility Act - (www.dhs.state.il.us/IITAA)
 - W3C Web Accessibility Initiative - (www.w3.org/WAI)
 - Section 508 - (www.section508.gov)