

Web Accessibility Quick-Check Guide

Use this checklist to audit your web pages and components against key WCAG 2.1 Level AA principles.

Part 1: Foundational Checks (Perceivable)

1.1. Images & Non-Text Content

- **Alt Text:** Every `` tag that conveys information has descriptive alternative text (`alt="..."`).
- **Decorative Images:** Purely decorative images have empty alt attributes (`alt=""`) so screen readers ignore them.
- **Complex Images:** Charts and graphs have a long description or are explained in the adjacent text.
- **Video/Audio:** Videos have accurate closed captions and a transcript is available. Audio-only content has a transcript.

1.2. Color & Contrast

- **Color Contrast:** Text has a color contrast ratio of at least 4.5:1 against its background (3:1 for large text). Use a contrast checker tool to verify.
- **Information in Color:** Color is not the *only* way information is conveyed (e.g., don't rely on red text alone for an error; use an icon or text label as well).

1.3. Text & Readability

- **Resizable Text:** Users can zoom the page up to 200% without breaking the layout or losing content.
- **Clear Language:** The content is written as clearly and simply as possible for the intended audience.

Part 2: Interaction Checks (Operable)

2.1. Keyboard Navigation

- **Keyboard Access:** Every interactive element (links, buttons, form fields) can be reached and activated using only the `Tab` and `Enter/Space` keys.
- **Visible Focus:** The element that currently has keyboard focus has a clear, visible outline or style. This outline is never removed with `outline: none;`.
- **Logical Order:** The tab order follows a logical path through the page content (usually top-to-bottom, left-to-right).

- [] **No Keyboard Traps:** The keyboard focus never gets "stuck" inside a widget or component. The user can always tab out of it.

2.2. Forms & User Input

- [] **Labels:** Every form field (`<input>`, `<textarea>`, `<select>`) has a properly associated `<label>`.
- [] **Error Identification:** If a form submission fails, errors are clearly identified, explained in text, and the user is guided back to the incorrect field.
- [] **Sufficient Time:** There are no unexpected time limits. If a timeout exists, the user can request more time.

Part 3: Structure & Code (Understandable & Robust)

3.1. Semantic HTML & Page Structure

- [] **Headings:** Heading levels (`<h1>`, `<h2>`, etc.) are used to create a logical document outline. Levels are not skipped (e.g., no `<h3>` directly after an `<h1>`).
- [] **Page Regions:** Main content is wrapped in `<main>`, navigation in `<nav>`, the header in `<header>`, and the footer in `<footer>`.
- [] **Lists:** Content that is a list is marked up correctly using ``, ``, and ``.
- [] **Buttons vs. Links:** Links (`<a>`) are used for navigation. Buttons (`<button>`) are used for actions (e.g., submitting a form, opening a modal).

3.2. ARIA for Dynamic Content

- [] **Use Native HTML First:** ARIA roles are only used when a standard HTML element is not sufficient.
- [] **Dynamic Updates:** Content that changes without a page reload (e.g., a "Product added to cart" message) is announced by screen readers, often using `aria-live` regions.
- [] **Descriptive Controls:** Custom controls have appropriate ARIA roles (e.g., `role="button"`) and accessible names (e.g., `aria-label` for an icon-only button).

Part 4: Final Testing

- [] **Automated Tools:** The page has been run through an automated checker (like Lighthouse, WAVE, or Axe) and all critical issues have been addressed.
- [] **Manual Screen Reader Test:** The page has been navigated using a screen reader (NVDA, VoiceOver, JAWS) to check for a logical and understandable experience.
- [] **User Feedback:** If possible, the page has been tested by users with disabilities.

.html