



# wA11y Stop & Fix Guide

A quick reference for identifying and resolving common accessibility red flags in web development. Use this guide to catch issues before they reach production.

V1.0

DEVELOPER REFERENCE



# Structural Red Flags: HTML Semantics

## Red Flags

- Using `<div>` or `<span>` for clickable elements instead of semantic HTML
- Missing or multiple `<h1>` elements on a single page
- Skipping heading levels (e.g., `<h2>` followed directly by `<h5>`)
- Non-semantic markup for interactive components

## ✓ Operational Fixes

- Use `<button>` for actions, `<a>` for navigation with proper href attributes
- Ensure exactly one `<h1>` per page to define the main content area; do not use `<h1>` for subsections or styling
- Maintain sequential heading hierarchy: `<h1>` → `<h2>` → `<h3>` without skipping levels
- Leverage native semantic elements before considering custom components

# Interactive Red Flags: Keyboard & Focus

## Red Flags

- Removing focus outlines with `outline: none` without providing a visible alternative
- Adding `tabindex="0"` to non-interactive elements like paragraphs or divs
- Focus traps without proper escape mechanisms

## ✓ Operational Fixes

- Implement high-visibility focus rings (WCAG standard: 3px solid) using the `:focus-visible` pseudo-class to ensure outlines only appear for keyboard users.
- Only interactive elements (buttons, links, inputs) should receive keyboard focus
- Manage focus programmatically when opening overlays or modals: move focus to the container and trap it until the element is dismissed.
- Test full keyboard navigation: Tab, Shift+Tab, Enter, Space, and Arrow keys

# Form & Input Red Flags

## Red Flags

- Using placeholder attribute as a substitute for `<label>` elements
- Using `aria-label` for inputs that have a redundant or missing visible label.
- Communicating error states through color alone (red borders without text)
- Missing or incorrect `for/id` associations between labels and inputs

## ✓ Operational Fixes

- Always provide permanent, visible `<label>` elements programmatically linked via `for/id`
- If a label is visually hidden for design reasons, use a `.sr-only` class on a `<label>` element rather than relying solely on ARIA attributes.
- Combine visual indicators with descriptive text using `aria-describedby` for error messages
- Include icons and text descriptions for all form validation states

# Landmarks & Data Red Flags

## Red Flags

- Using `<table>` elements for visual layout or grid-based designs.
- Pages missing `<header>`, `<main>`, `<nav>`, or `<footer>` landmarks.
- Missing or incorrect `aria-label` or `aria-labelledby` on complex data widgets or custom components.

## ✓ Operational Fixes

- Use CSS Flexbox or Grid for layout; reserve `<table>` strictly for tabular data with defined `<thead>` and `scope="col/row"`.
- Wrap core page regions in semantic landmark tags to allow screen reader users to jump to specific sections.
- Ensure all complex widgets have clear, programmatically associated labels using `aria-label` or `aria-labelledby`.

# Visual & Media Red Flags

## Red Flags

- Meaningful images missing alt text or decorative images with generic alt="image"
- Write descriptive link text that conveys destination and intent without context (e.g., "Download Token Spreadsheet" instead of "Click Here").
- Videos without captions or transcripts

## ✓ Operational Fixes

- Use descriptive alt text for content images and alt="" (null) for purely decorative elements
- Write descriptive link text that explains the destination: "Download Token Spreadsheet" instead of "Click Here"
- Provide synchronized captions and full transcripts for all video and audio content

# The Golden Rule for wA11y Developers

**Use native HTML elements first.** Only implement ARIA when native semantics are insufficient. Remember: "No ARIA is better than bad ARIA."

## Test Early

Integrate accessibility testing into your development workflow, not as an afterthought

## Use Real Devices

Test with actual screen readers (NVDA, JAWS, VoiceOver) and keyboard-only navigation

## Reference Library

Visit the wA11y Component Library for full specifications, code examples, and implementation patterns