THE 80/20 WEB ACCESSIBILITY CHEAT SHEET

Eliminate 80% of Assistive Technology Barriers with 6 Core HTML Fixes | **cantclickthis.dev**

1. The Core Principle: Pareto's Rule & WebAIM Data

The 80/20 Rule: Effort and outcomes are rarely 1:1. Vilfredo Pareto observed that 20% of inputs produce 80% of results. According to the **2026 WebAIM Million Report**, 96% of all detected accessibility failures across top homepages stem from just six basic categories. By focusing on these vital HTML fundamentals, developers can remove 80% of digital barriers for screen reader and keyboard users.

> **Key Concept: What is Semantic HTML?**
>
> Semantic HTML means using an HTML element for **what it means**, not how CSS renders it visually. CSS controls visual styling; HTML tags communicate structure, interactivity, and role to search engines, browsers, and screen readers. A styled with an onClick handler is just a silent box to a keyboard or screen reader user.

2. The 6 Essential Screen Reader & Keyboard Fixes

Fix & Focus Area	The Golden Rule	Bad Pattern (Avoid)	Accessible Code (Use)
#1 Image Alt Text (Content)	Describe the image's meaning as if replacing it with text. Omit redundant 'image of' phrases.	<img src="cat.jpg" alt="image"> *(Reads: Graphic, image)*	<img src="cat.jpg" alt="Orange tabby sleeping on windowsill">
#2 Headings & Landmarks (Navigation)	Use <h1>–<h6> sequentially. Wrap layout in <header>, <nav>, <main>, <footer>.	Schedule *(Screen reader cannot skim)*	<main> Event Schedule </main>
#3 Form Input Labels (Forms)	Every input needs an explicit <label>. Placeholders disappear when typing and don't replace labels.	<input type="email" placeholder="Email"> *(Reads: Edit text)*	<label for="e">Email</label> <input id="e" type="email">
#4 Buttons vs. Links (Interactivity)	**Buttons do something** (Spacebar/Enter). **Links take you somewhere** (Enter). Never use clickable .	Submit *(Skipped by Tab key)*	<button type="submit"> Submit</button>
#5 Descriptive Link Text (Context)	Link text must make sense out of context when isolated in a screen reader's link list.	Click Here *(Reads: Click Here, link)*	Register for DevFest 2026
#6 Document Language (Voice Engine)	Declare lang="en" on the <html> tag so screen readers select the correct pronunciation engine.	<html> *(Screen reader uses default system voice, mispronouncing text)*	<html lang="en"> *(Screen reader uses proper English voice synthesis)*

3. Speaker & Developer Q&A Cheatsheet

Question / Challenge	The 80/20 Technical Answer
Q: Is it okay to make an anchor link look like a button with CSS?	**Yes!** Visual styling doesn't dictate semantics. If clicking the element changes the URL or navigates to a new page, use an <a> tag (even if CSS styles it like a rounded button). If it triggers an in-page action (submitting a form, opening a modal, toggling a drawer), use a <button>.
Q: Why is a clickable bad if I add an onClick listener?	A is non-interactive by default. It cannot receive keyboard focus via the Tab key, does not respond to the Spacebar key, and isn't announced as a button by screen readers. You would have to manually add tabindex="0", role="button", and keydown listeners to replicate what <button> gives you for free.
Q: What about AI code generators (Copilot, Cursor, ChatGPT)?	AI tools frequently generate visually polished UIs using non-semantic HTML (e.g., placeholder-only inputs, buttons). Always inspect generated markup or explicitly prompt AI: *'Ensure semantic HTML5, explicit form labels, and proper ARIA button semantics.'*

4. The 5-Minute Accessibility Audit

> **Action Plan: One Tool, One Habit**
>
> **One Tool:** Install a free browser contrast & accessibility extension (e.g., axe DevTools, WAVE, or Accessibility Insights) to catch structural issues automatically.
>
> **One Habit:** Unplug your mouse and **Tab through your application** once before shipping to production.
> - Can you see where keyboard focus is at all times? (No hidden focus rings)
> - Can you trigger all buttons and forms with Spacebar and Enter?
> - Can you navigate through all interactive controls without getting trapped?

5. Audience Engagement: Slido Live Poll Reference

Poll Question	Multiple Choice Options	Key Talk Takeaway
Q1: AI Building Habits How often do you use AI tools (Copilot, Cursor, ChatGPT) to generate front-end code?	• Daily / Core workflow • Weekly / Occasional assistant • Barely / Only for snippets • Never	Establishes high baseline of AI adoption in modern developer workflows (~80%).
Q2: Accessibility Reliance When using AI, how often do you explicitly prompt for WCAG compliance or audit the HTML?	• Always (Explicit prompts) • Sometimes (Visual check) • Rarely (Assume clean AI output) • Never thought about it	Highlights the 'AI Accessibility Gap'—explaining why WebAIM reports show surging automated errors.