wA11y.io

The wA11y Developer Toolkit

The essential stack for building inclusive products.

Accessibility isn't optional—it's the foundation of great product development.

Use this toolkit during local development, code review, QA, and before release

30%

Automated Detection

What tools can find

70%

Manual Testing

What humans must verify

Part 1: Essential Automated Tools

The Daily Drivers

These browser extensions form the backbone of your automated accessibility testing workflow. Each tool catches different issues, so using them in combination maximizes coverage.

Browser Extensions

axe DevTools	*[Freemium]*	The industry standard for automated testing. Integrates seamlessly with Chrome DevTools and provides detailed remediation guidance.
WAVE	*[Free]*	Visual feedback tool that identifies errors directly in the browser with inline icons and color-coded alerts.
Accessibility Insights	*[Free]*	Guided and automated assessment tool by Microsoft. Excellent for structured WCAG compliance testing.
Google Lighthouse	*[Free]*	Built-in audit tool in Chrome DevTools. Provides accessibility scores alongside performance and SEO metrics.

Color & Code Utilities

Beyond browser extensions, these specialized tools ensure your code and designs meet accessibility standards from the start.

WebAIM Contrast Checker	*[Free]*	The go-to for checking WCAG color ratios. Simple interface for testing text and background combinations.
Stark	*[Freemium]*	Plugin for Figma and Sketch to check contrast and simulate vision conditions during the design phase.
eslint-plugin-jsx-a11y	*[Free]*	Static AST checker for React developers. Catches accessibility issues in JSX before code reaches production.

Screen Readers & Documentation

Testing Software

To truly understand user experience, you must test with the same tools your users rely on. Here are the essential screen readers across platforms.

NVDA

[Free]

The most popular open-source screen reader for Windows. Essential for testing desktop web applications.

nvaccess.org

VoiceOver

[Free]

Built-in to all Apple devices (macOS/iOS). Test both desktop Safari and mobile experiences.

apple.com/accessibility

TalkBack

[Free]

Built-in to Android devices. Critical for mobile app and responsive web testing.

google.com/accessibility

JAWS

[Paid]

Enterprise standard screen reader. Widely used in corporate and government environments.

freedomscientific.com

Essential Bookmarks

Keep these authoritative resources handy for standards, best practices, and technical implementation guidance.

W3C WAI

Official standards body for web accessibility. Home to WCAG guidelines and ARIA specifications.

w3.org/WAI

The A11y Project

Beginner-friendly guides and checklists. Perfect for teams new to accessibility or looking for quick reference.

a11yproject.com

MDN Web Docs

Best technical documentation for ARIA, semantic HTML, and accessibility APIs with clear code examples.

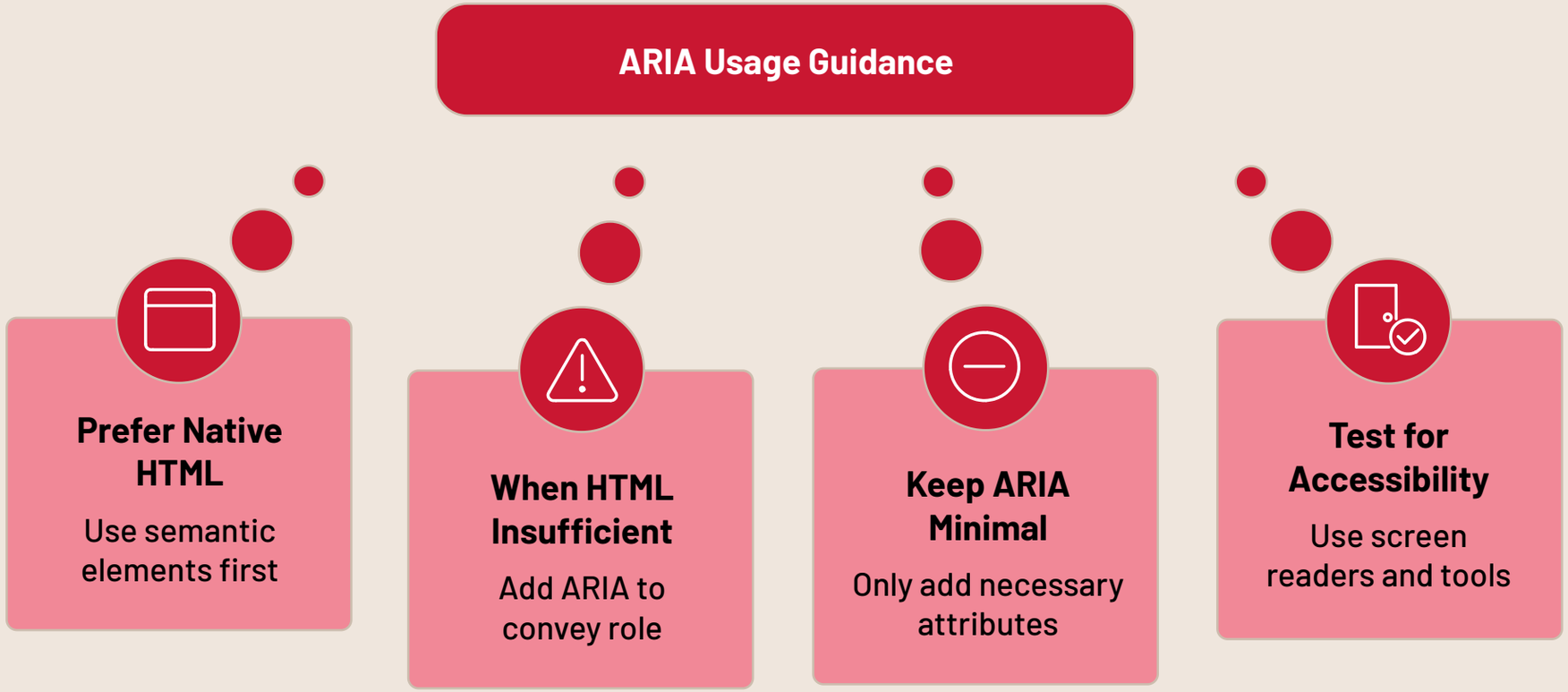
developer.mozilla.org

Part 2: Manual Testing Checklist

The "Human" Check (The Other 70%)

If you can't use it without a mouse, it's not done.

Automated tools are powerful, but they can't evaluate user experience. Manual testing reveals the barriers that tools miss—and those barriers affect real people trying to use your product.



Keyboard Testing (No Mouse)

Unplug your mouse and navigate your entire application using only Tab, Shift+Tab, Enter, Space, and Arrow keys. If you get stuck or lost, your users will too.

- [ ] **Visual Focus**

  Can you *always* see where you are? Ensure focus indicators are visible and never suppressed with outline: none without replacement.

- [ ] **Logical Order**

  Does tab order match visual reading order? Content should flow naturally from top to bottom, left to right.

- [ ] **No Traps**

  Can you tab *in* and *out* of every widget? Test modals, carousels, maps, and custom components carefully.

- [ ] **Interaction**

  Can you trigger buttons and menus with Enter or Space? All interactive elements must respond to keyboard input.

- [ ] **Skip Link**

  Is there a "Skip to Content" link on the first tab? This lets keyboard users bypass repetitive navigation.

Screen Reader & Visual Checks

Screen Reader Basics

Turn on a screen reader and navigate your site with your eyes closed. Listen critically to what's announced and how information is structured.

- **Headings:** Do you skip levels (H1 to H4)? Are visual headings coded as actual heading elements?
- **Links:** Do links make sense out of context? Avoid vague phrases like "Click Here" or "Read More."
- **Images:** Do decorative images stay silent? Do meaningful images have descriptive alt text?
- **Forms:** Do fields announce their label and required status clearly?

Zoom & Reflow

Test how your interface adapts to users who need larger text or different viewport sizes.

- **200% Zoom:** Is text readable without horizontal scrolling? Content should reflow naturally.
- **400% (Mobile):** Does the site reflow into a single column? No content should be hidden or cut off.
- **Color:** Are error messages conveyed by more than just color? Include text, icons, or other visual indicators.

The "Gotchas"

What Automated Tools Often Miss

These nuanced issues separate adequate accessibility from excellent user experience. Automated scanners can't evaluate context, clarity, or user flow—which is why manual testing is critical.

Focus Not Obscured (2.4.11 AA)

Ensure sticky headers/footers/banners do not fully cover focused elements. Use `scroll-margin-top`/`scroll-padding-top` equal to sticky height; make banners modal/dismissible.

Test: Tab through with sticky UI active (i.e. sticky headers and overlays); confirm focused control is at least partly visible.

Accessible Authentication (3.3.8 AA)

Allow password managers & copy-paste; provide passkeys/biometrics/magic links.

Test: Complete login without memory/transcription; verify managers fill fields; OTP fields allow paste.

Consistent Help (3.2.6 A)

Place Help/Contact consistently (same DOM order and landmark).

Test: Crawl a sample of pages; check stable placement/order.

Redundant Entry (3.3.7 A)

In multi-step flows, prefill or let users select previously entered info; allow “same as X”.

Test: Confirm no unnecessary retyping; verify security exceptions only.

Target Size Minimum (2.5.8 AA)

Document the 24×24 requirement and exceptions; keep your 44×44 as preferred.

Test: Sample buttons/links on mobile and desktop; confirm size or spacing equivalence.

The "Gotchas"

What Tools Often Miss

These are the nuanced issues that separate adequate accessibility from excellent user experience. Automated scanners can't evaluate context, clarity, or user flow.

Error Messages

Are they helpful and specific? Compare "Password must be at least 8 characters with one uppercase letter" to "Error." Which one helps users succeed?

Focus Management

Does focus return to the trigger button when a modal closes? Or does it jump back to the top of the page, disorienting the user?

Dynamic Content

Are "toast" messages and live updates announced immediately to screen reader users? Use ARIA live regions appropriately.

Mobile Accessibility Checklist

Testing beyond desktop: Ensuring your mobile experience is accessible for all users, including those relying on touch and assistive technologies.

Touch Targets & Gestures

- Touch targets are at least 44×44 CSS pixels
- No critical actions rely on swipe-only, long-press, or multi-finger gestures
- Complex gestures have a single-tap alternative

Screen Reader + Touch

- Test with VoiceOver (iOS) and TalkBack (Android)
- Ensure swipe navigation follows logical reading order
- Interactive elements announce role, state, and label

Motion & Interaction

- Motion and animations can be reduced or disabled
- No interaction depends on device motion (shake, tilt)
- Drag-and-drop has a non-gesture fallback

If an action can’t be completed with simple taps and screen reader gestures, it isn’t accessible.

> "Accessibility is not a step in the process; it is the process."

Building accessible products requires both automated tools and human judgment. Use the tools in this guide to catch technical issues early, then validate with real-world manual testing.

Remember: you're not building for compliance—you're building for people.

---

wA11y.io

hello@wa11y.io 🌐 wa11y.io