

Accessibility Testing in WordPress

Facilitator Guide

Overview

Participants use the Accessibility Checker plugin inside WordPress Playground to scan a page for accessibility issues, fix three common problems directly in the Block Editor, then test the page using only a keyboard. No prior accessibility knowledge required.

Topic	Accessibility Testing in WordPress
Duration	60 minutes
Audience	Beginner — no prior accessibility knowledge needed
Prerequisite	A modern browser; no WordPress account required
Environment	WordPress Playground (playground.wordpress.net) — no account or install needed
Materials	This facilitator guide, slide deck, participants bring their own device

Note: Any tools or plugins mentioned are suggestions, not endorsements. The Accessibility Checker plugin is used here as a free, well-maintained example. Alternatives include Sa11y (wordpress.org/plugins/sa11y/) and Editoria11y (wordpress.org/plugins/editoria11y-accessibility-checker/).

Learning Objectives

By the end of this activity, participants will be able to:

- * Identify at least three categories of common web accessibility issues: missing alt text, incorrect heading order, and non-descriptive link text. (Remember)
 - * Explain what WCAG conformance levels mean and why they matter for a WordPress site. (Understand)
 - * Use the Accessibility Checker plugin to scan a page and interpret its results using current terminology: Problems, Needs Review, and Dismiss. (Apply)
 - * Fix at least two accessibility issues directly in the Block Editor following the plugin's guidance. (Apply)
 - * Distinguish between what automated testing can catch and what only a human can verify. (Analyze)
 - * Assess a page's accessibility by navigating it using only a keyboard, without a mouse. (Evaluate)
-

What Participants Need

Each participant needs:

- * A device with a modern browser (Chrome or Firefox recommended for best Playground compatibility)
- * WordPress Playground open at playground.wordpress.net — no login or account needed
- * No prior WordPress or accessibility experience required

Facilitator note: Test WordPress Playground in the venue's browser at least 24 hours before the session. Playground requires a live internet connection. If your venue's WiFi is unreliable, ask participants to install Local by Flywheel (localwp.com), MAMP (mamp.info), or Studio by WordPress.com (developer.wordpress.com/studio) as a backup.

Browser extension note: Browser extension-based tools such as the WAVE browser extension and the axe DevTools extension do not work reliably inside WordPress Playground because Playground renders inside an iframe. This activity uses the Accessibility Checker plugin, which runs inside WordPress itself and works correctly in Playground.

Facilitator Setup Notes

Before the session:

1. Open playground.wordpress.net in your browser. Let it fully load (20-30 seconds). You should see the WordPress dashboard.
2. Install Accessibility Checker: go to Plugins, then Add Plugin. Search for "Accessibility Checker" by Equalize Digital. Install and activate it.
3. Create a sample post for scanning: go to Posts, then Add Post. Add a title, a paragraph of text, an image without alt text, a few headings (mix up the order intentionally), and a link with the text "click here". Publish the post.
4. Open your slides. Have the Playground tab open alongside the slides so you can switch between them.
5. Allow about 15 minutes for setup.

How to introduce the activity:

- * Accessibility is about making sure everyone can use your site, including people who use screen readers, keyboard-only navigation, or have low vision. It is also increasingly a legal requirement in many countries.
- * WordPress makes it easier to build accessible sites than it used to. But publishing accessible content still takes deliberate attention. Today you will develop a workflow for catching the most common issues before you hit Publish.
- * We will use an automated scanner first, then do a quick manual test that shows you what your keyboard-only visitors experience.

Part 1: What Is Accessibility and Why Does It Matter | 10 minutes

Background

Web accessibility means that websites and tools are designed so people with disabilities can use them. This includes people who are blind or have low vision, people who are deaf or hard of hearing, people with motor impairments who cannot use a mouse, and people with cognitive differences. The Web Content Accessibility Guidelines (WCAG) are the international standard. WCAG 2.2 Level AA is the current conformance standard -- the version WordPress core targets, required or being adopted into law by EU member states, and widely referenced in accessibility legislation across the US, UK, and Australia.

What Participants Do

Facilitated discussion (no hands-on steps). Use the slides to walk through:

- * Who benefits from accessible web content (not just screen reader users)
- * The three most common accessibility issues found on WordPress sites: missing image alt text, incorrect heading order, and non-descriptive link text

- * The difference between automated testing (fast, catches structural issues) and manual testing (required to catch context and usability issues)
- * WCAG conformance levels A, AA, and AAA: what they mean in plain language

Facilitator tip: Keep Part 1 conversational. Ask the group: "Has anyone encountered an accessibility issue on a site they were trying to use?" Real experiences from the group make the topic concrete. Aim to spend no more than 10 minutes here so there is plenty of time for the hands-on work.

WordPress 7.0 update (May 2026): WordPress 7.0 (released May 2026) targets WCAG 2.2 Level AA for all core development. Notable accessibility improvements in 7.0 include: the login password reset flow now pre-populates with the username to comply with WCAG 2.2; the admin UI received a design refresh (new 'Modern' color scheme) with improved color contrast; voice control usability was improved so that visible control labels match their accessible names (post list tables and media library); and view transitions now respect the OS-level 'reduce motion' preference and are disabled when it is set.

Two developer-facing a11y improvements worth highlighting: (1) a new `wp_get_image_alttext()` function automatically imports alt text from an image's IPTC metadata — useful context when discussing how WordPress helps authors fill in alt text. (2) Title attributes have been removed by default from three author link functions (`get_author_posts_url` and `related`), since title attributes are largely inaccessible to keyboard and touch users and are considered an anti-pattern. Check make.wordpress.org/accessibility/current-release-progress/ for the full list.

Part 2: Automated Testing with Accessibility Checker | 20 minutes

Background

Accessibility Checker by Equalize Digital runs automated scans inside the WordPress Block Editor and on the frontend of your site. It checks for 40+ issues and can automatically fix 10 of them. The current version (1.38+) uses the following terminology: Problems (issues that fail a WCAG rule), Needs Review (items a human should check), and Dismiss (mark as reviewed). These replaced the older terms Errors, Warnings, and Ignore.

What Participants Do

1. Go to playground.wordpress.net in your browser. Wait for the WordPress dashboard to load (about 20-30 seconds).
2. In the left menu, click **Plugins**, then **Add Plugin**.
3. In the search box, type **Accessibility Checker**. Find the plugin by **Equalize Digital**. Click **Install Now**, then **Activate**. After activation the plugin will redirect to its dashboard. If an "Unlock Detailed Accessibility Reports" popup appears, press Escape to dismiss it.

4. Go to **Posts** in the left menu, then click **Add Post**.
5. Add a title, then add some content. Include: at least one image (skip the alt text for now), a few headings in the wrong order (for example, H2 then H4, skipping H3), a paragraph with a link that just says "click here", and a couple of sentences of regular text.
6. Click **Publish**, then **Publish** again to confirm.
7. After saving, look in the right-hand **Post** sidebar panel. You will see a collapsible **Accessibility Checker** section. It scans automatically when the post is saved. The summary will show the number of problems and items that need review. Click **Open Accessibility Panel** to view the full results.
8. The Accessibility Checker panel opens. Under **Accessibility Analysis**, click the **Problems** tab. Each issue has a description and a link to learn more. Read through two or three of them.
9. Click the **Needs Review** tab. Notice that these items require a human judgment call, not just a code fix.

Facilitator tip: The Accessibility Checker panel is in the right-hand Post sidebar, not below the editor. After saving the post, participants should see it listed as a collapsible section between the post metadata and the Categories panel. If they do not see it, make sure the Post tab (not Block tab) is selected at the top of the sidebar, and that the panel is expanded.

Terminology reminder: Version 1.38 (March 2026) changed the labels. "Problems" was previously called "Errors." "Needs Review" was previously called "Warnings." "Dismiss" was previously called "Ignore." Use the current terms in your facilitation.

Part 3: Fix Common Accessibility Issues | 20 minutes

Background

The three issues participants intentionally created in Part 2 represent the most commonly found accessibility problems on WordPress sites: missing image alt text, incorrect heading order, and non-descriptive link text. Each can be fixed directly in the Block Editor.

What Participants Do

Fix 1: Add alt text to an image

1. In the Block Editor, click on the image you added.
2. In the block settings panel on the right, find the **Alt text (alternative text)** field.

3. Type a clear, descriptive alt text that explains what the image shows. Avoid starting with "Image of" or "Picture of" — screen readers already announce it is an image.
4. Click **Update** to save the post.

Facilitator tip: Good alt text describes the content and function of the image. A photo of a WordPress event: "A group of WordPress contributors collaborating at a laptop" is good. "IMG_4532.jpg" or "WordPress" is not. Decorative images (dividers, backgrounds) can use empty alt text (alt="") to tell screen readers to skip them.

Fix 2: Correct heading order

1. Click on a heading in the post. Check its level in the toolbar (the dropdown that shows H1, H2, H3, etc.).
2. Headings must follow a logical order without skipping levels. H1 is the page title (set automatically). Body content should start at H2 and step down to H3, H4, and so on without gaps.
3. Fix any heading that skips a level. Change an H4 to H3, or an H3 to H2, so the heading structure reads like an outline.
4. Update the post and re-run the Accessibility Checker scan to confirm the issue is resolved.

Facilitator tip: A useful analogy: headings work like a book outline. You would not go from Chapter 1 to Chapter 1, Section 3 without a Section 2. Screen readers use headings to help users navigate and understand the structure of a page.

Fix 3: Improve link text

1. Find the link with the text "click here" in the post.
2. Select the text, then retype it so it describes the destination. For example: "Read the WCAG 2.2 guidelines" instead of "click here".
3. Update the post and re-run the scan. Confirm the non-descriptive link text issue is resolved.

Facilitator tip: Screen reader users often navigate by jumping from link to link. If every link says "click here" or "read more," they have no idea where each link goes. Descriptive link text also benefits SEO.

Stretch goal: Ask participants to click the Accessibility Checker's "Auto Fix" button (if present for their remaining issues) and observe which issues it can resolve automatically versus which require a human decision. This reinforces the automated vs. manual distinction from Part 1.

Part 4: Manual Keyboard Navigation Test | 5 minutes

Background

Automated tools catch structural issues but cannot evaluate usability. A keyboard navigation test gives participants a direct experience of what keyboard-only users and screen reader users encounter when moving through a page. Many motor impairment users and screen reader users navigate entirely without a mouse.

What Participants Do

1. Click the **View Post** link at the top of the editor to open the published post in a new tab.
2. Put your mouse aside. Use only the keyboard from here: press **Tab** to move forward through interactive elements (links, buttons, form fields), **Shift+Tab** to move backwards, **Enter** to activate a link, and **Space** to activate a button or checkbox.
3. Navigate through the entire post. Notice: Can you see where the keyboard focus is at all times? Is there a visible focus indicator (usually a highlight or outline) on the focused element?
4. Try to find and activate your link using only the keyboard. Can you tell where it goes before pressing Enter?

Facilitator tip: Ask the group: "Was the keyboard focus always visible? Were there any places you felt lost?" Twenty-Five (the default WordPress block theme) has good keyboard focus indicators. If participants are on a theme with poor focus styling, this test can be eye-opening. Note that focus indicator visibility is a WCAG 2.2 requirement (Success Criterion 2.4.11).

Facilitator note: This is a 5-minute introduction to manual testing, not a comprehensive audit. Participants who want to go further can try VoiceOver (Mac: Command+F5) or NVDA (Windows, free download) to experience the page with a screen reader.

Stretch goal -- Zoom testing (5 extra minutes): Ask participants to zoom the browser to 200% (Ctrl/Cmd + three times) and then to 400% (Ctrl/Cmd + six times from 100%). At each zoom level, have them check two things: (1) Keyboard navigation -- zoom until the mobile navigation menu

appears, then use only the keyboard to confirm the mobile menu toggle can be activated with the Space bar; (2) Visual check -- scroll through the page and confirm that no content is overlapped, cut off, or missing. Zoom reflow and responsive design are WCAG 2.1 requirements (Success Criteria 1.4.4 Resize Text and 1.4.10 Reflow). Return the browser to 100% before moving on.

Debrief and Wrap-Up

5 minutes. Bring the group back together for a short discussion.

Discussion questions

- * **What surprised you most about the accessibility issues the scanner found?**
- * Which of the three fixes (alt text, heading order, link text) do you think is most common on WordPress sites you have seen?
- * The scanner can catch structural problems, but it cannot tell you if your alt text is actually descriptive, or if your content is easy to understand. What would a more thorough accessibility review look like for your site?

Win statement

You just ran a real accessibility audit, fixed three of the most common issues on any WordPress site, and experienced keyboard navigation firsthand. These are the same checks professional accessibility reviewers run. Building this into your publish workflow will make a real difference for your visitors.

Next step

Point participants to learn.wordpress.org/lesson/tools-for-testing-theme-accessibility/ for a deeper dive into accessibility testing tools, and to make.wordpress.org/accessibility/ if anyone is interested in contributing to WordPress accessibility. For ongoing accessibility news, resources, and community support, point participants to the [WP Accessibility Knowledge Base](https://wpaccessibility.org/) at wpaccessibility.org.

Sources

All URLs verified as of May 2026. Check before running the activity.

1. [Accessibility Checker — WordPress plugin](#): Free plugin by Equalize Digital; 40+ automated checks; runs inside the Block Editor; tested up to WordPress 6.9.
2. [Accessibility Checker changelog](#): Verify current version and any UI terminology changes before running the activity.

3. [Web Content Accessibility Guidelines \(WCAG\) 2.2 — Overview](#): The international standard for web accessibility; explains conformance levels A, AA, and AAA. WordPress core targets WCAG 2.2 Level AA.
4. [Tools for testing theme accessibility — Learn WordPress](#): Official Learn WordPress lesson covering automated and manual accessibility testing tools.
5. [Make WordPress Accessible](#): The official WordPress accessibility team hub; handbook, contributor info, and current initiatives.
6. [Sa11y — WordPress plugin](#): Free alternative accessibility checker by Toronto Metropolitan University; geared toward content authors.
7. [Editoria11y Accessibility Checker — WordPress plugin](#): Free in-editor accessibility checker; alternative approach to Accessibility Checker.
8. [Introduction to Web Accessibility — W3C WAI](#): Plain-language introduction to why accessibility matters and who benefits.
9. [WP Accessibility Knowledge Base](#): Community-maintained knowledge base covering WordPress accessibility tools, techniques, plugins, and resources.