

Shift Left with Accessibility Checklist

This checklist is intended to help teams launch more accessible websites faster with fewer remediation items left at the end of the project, a.k.a., shifting accessibility left in the website design and development process.

Each section lists items that team members either need to complete or simply be aware of as they complete their normal tasks within that specific phase of the project.

If this feels overwhelming or like you need additional assistance, we're here to help. Feel free to ask questions in our [WordPress Accessibility Facebook group](#), or you may be interested in our [accessibility consulting services](#).

Discovery

- ☐ Establish the standards that need to be met (WCAG 2.0, 2.1, or 2.2; AA or AAA).
- ☐ Finalize list of design and functionality desires/needs taking accessibility into account.
- ☐ Research the accessibility of any new plugins or other third-party tools and determine if time for remediation will need to be built into development.
- ☐ Generate sitemap and navigation structure.
- ☐ Gain a clear understanding of anticipated users and any special needs.
- ☐ Create detailed user journey maps mapping out the process users are expected to take as they move through the website/engage with the organization.
 - ☐ Include people with disabilities in journey maps.
 - ☐ Identify how, if at all, the process may need to change for certain user types or abilities and ensure there are accessible alternatives in the user journey.
- ☐ Review content to be imported from current website and identify accessibility issues that are going to be imported in with content from previous website.
 - ☐ Discuss remediation options with client: they will remediate, scope will be added to project for content team to remediate, or website will launch with some errors?
 - ☐ Identify content that can be deleted/not imported.

Content

- ☐ Client has been provided with a content accessibility checklist and provided instructions on how to provide content that is accessible.
- ☐ Headings are present in long documents and used in proper order.
- ☐ Underline is never used for emphasis.
- ☐ No content is typed out in all capital letters.
- ☐ Link anchor text is meaningful.
- ☐ Content is formatted in HTML lists as applicable.
- ☐ Tabular data is in tables with appropriate headings or in spreadsheets to be added via CSV import.
- ☐ Tables, written descriptions, or other alternate content is provided for complex graphics, such as graphs, charts, or infographics.
- ☐ All videos have *accurate* captions - not auto captions.
- ☐ All videos and audio files have *accurate* transcripts.
- ☐ Videos include optional descriptive narration so they make sense if only hearing the audio (AAA only).
- ☐ Linked PDF files, Word Docs, etc. are accessible and pass CommonLook scans.
- ☐ There are no animated GIFs.
- ☐ Embedded content from social media or other third-party sources has been tested for accessibility.
- ☐ Any iframe codes to be included in the website have a title attribute.
- ☐ Content reading level is appropriate for audience.
- ☐ Simplified summaries have been provided for exceptionally long content or content with reading levels above grade 9 (AAA compliance).
- ☐ Proper alt text has been provided for images either in content request doc or a separate spreadsheet for import.
 - ☐ Purely decorative images should have empty alt text.
 - ☐ Meaningful images have alt text that describes the image's appearance and that makes sense in the context of the page.
 - ☐ Linked images have alt text that describes the image's purpose (where the link goes) rather than the appearance of the image.

Design

- ☐ Selected fonts are readable - ideally sans-serif fonts with spacing that is consistent between characters and wider rather than tight.
 - ☐ Check confusing letter combinations such as “rn”, “Il1”, “O0o”, “bdp”, and “B8” to ensure there is adequate difference and space between the letters.
- ☐ Color palette has been tested to confirm it is color blind safe with [Adobe Accessible Color Palette](#) tool.
- ☐ Colors expected to overlay one another have been tested for appropriate contrast - ideally AAA. (Use [Contrast Grid](#) to create a visual of approved combinations.)
- ☐ Design for multiple screen sizes/devices/orientations, not desktop only.
- ☐ All labels/fields and other related items have close proximity so it's obvious which elements are related to one another.
- ☐ All fields have visible labels and clear required indicators for required fields (not placeholder text only; the word “required” is better than an asterix).
- ☐ Hover, focus, and error states have been designed.
- ☐ Color alone is not used to convey information.
- ☐ No text has been designed in a way that cannot be styled with CSS or that would require spans for every letter. (I.e., wavy text, rainbow text, “WordArt”.)
- ☐ Heading styles across pages are consistent.
- ☐ Use sentence case for readability or limit title case to headings. No all caps.
- ☐ Always underline links outside of the navigation and headings (I.e., post titles).
- ☐ Include pause buttons on content that plays automatically.
- ☐ Keep navigation consistent throughout the site.
- ☐ Include multiple ways of navigating the site/content such as the top menu, a list of related pages in a sidebar, link to a sitemap in footer, etc.
- ☐ A search box or feature is present in an obvious placement and has a visible “search” label (not placeholder text).
- ☐ Avoid designing pop ups as much as possible.
- ☐ No text has been included in the design that is only visible on hover.
- ☐ Limit carousels, accordions, parallax, background videos in the design.
- ☐ Designs pass all color contrast tests (use [Stark](#) in Adobe XD, Sketch, or Figma).
- ☐ Developer/accessibility specialist has reviewed designs for possible accessibility red flags or implementation challenges before showing to client.

- ❑ If issues exist, either resolve in design or come up for a plan on how the feature will be implemented accessibly.

Development

- ❑ Ensure [Webhint](#) is active in the project to test for accessibility issues as you code.
- ❑ If using a personal Github repo, ensure [axe Linter](#) is configured to check for accessibility issues in pull requests on JS.
- ❑ Start from an accessibility-ready theme or custom base theme that is accessible.
- ❑ Include skip links as the first focusable element.
- ❑ Style clear focus states for all focusable elements.
- ❑ Ensure all controls and links can be accessed by the keyboard only.
- ❑ Anything that behaves as buttons or links must use `<button>`, `<input>`, or `<a>` tags - **not** `<div>` or `<span>`.
- ❑ All form inputs must have explicitly associated `<label>` elements.
- ❑ Use correct ARIA landmark roles and semantic HTML to describe structural landmarks (banner, main, complementary, navigation, etc.).
- ❑ There are CSS styles in the theme that allow a screen reader class to be applied as needed to any element in the content editor or page builder. Typically `.sr-only`
 - ❑ Screen reader elements are not hidden on the edit screen - I.e., these styles should be excluded from any admin stylesheets.
 - ❑ If building with the block editor, install and activate the [Screen Reader Text Format plugin](#).
- ❑ There are no ambiguous links anywhere on the site.
 - ❑ Append post titles or other unique details to ambiguous links (such as “read more”) with screen reader text or in an aria-label, this applies to both coded links and links or buttons added in the editor/a page builder.
- ❑ Minimize or remove all links that open in a new tab or window.
 - ❑ If there will be links that open in a new tab or window, install and activate the [Accessibility New Window Warnings plugin](#).
- ❑ Install and activate [Accessibility Checker plugin](#).
- ❑ Developer has keyboard and screen reader tested all key page(s) during development and prior to handing the site off to the project manager/team for review.
- ❑ Any identified accessibility issues have been either fixed or flagged for project manager if difficult to fix/possibly out of scope.

Testing

- ☐ Run Accessibility Checker Pro automated scan to check for accessibility problems throughout the entire site.
 - ☐ Go through Fast Track (on the Open Issues page) and identify/fix any code snippets that are flagging issues on every page or most pages.
 - ☐ Create global ignores (in Fast Track) for any false positives or manual review items coming up on all or many pages
- ☐ Accessibility specialist or second team member (not developer so there's a second set of eyes) manually keyboard and screen reader tested a representative page of every type (home, all archives + singles, and any pages with special features/in scope of the project).
- ☐ Manual Testing Items
 - ☐ Page title is correct.
 - ☐ Skip links are the first focusable element and jump to the correct location on the page.
 - ☐ Tab order follows order of page.
 - ☐ Content contained in correct HTML landmarks
 - ☐ Dialog boxes/pop-ups are keyboard accessible and dismissable.
 - ☐ When popup opens, focus is shifted to the open popup and screen readers announce that a popup has opened.
 - ☐ Popup contains a heading and clear text.
 - ☐ Any forms contained in popups follow accessibility best practices for forms.
 - ☐ Focus when hitting the tab key is locked so user cannot leave the popup.
 - ☐ Close button is appropriately labeled and works with keyboard only.
 - ☐ When popup is closed user's focus returns to where they were on the page when it opened.
 - ☐ Focus returns to a logical location when dialog boxes or pop-ups are closed.
 - ☐ Carousels can be fully interacted with via keyboard alone.
 - ☐ Alt tags are accurate.
 - ☐ Language changes are announced.
 - ☐ Any CAPTCHAs are accessible (I.e., no "choose pictures of signs").
 - ☐ Users control whether or not to play audio or video.

- ☐ Page content does not change unexpectedly (I.e., no AJAX search without warning).
- ☐ All issues from scan and manual testing have been resolved.

User Testing

(If applicable.)

- ☐ Drafted testing brief outlining important tasks (relevant to user journey) that need to be completed during a user testing session.
- ☐ Native screen reader users have tested key pages and features on the website to confirm accessibility.
 - ☐ Minimum of two users of different experience levels using different devices and screen readers.
- ☐ Other relevant user types representative of audience (elderly, cognitive disabilities, motor impairment, etc.) have tested the website.
- ☐ Any issues identified during user testing sessions have been resolved.

Training/Post Launch

- ☐ Client has been trained on Accessibility Checker, how to read reports, and knows what to do to fix identified issues or how to get accessibility support moving forward.
- ☐ Client knows how to enter content in an accessible manner. Most common mistakes clients make should be reiterated from the content phase, for example:
 - ☐ How to add/edit alt tags on images.
 - ☐ Headings must be added in the proper order.
 - ☐ Not to use ambiguous anchor text.
 - ☐ Color contrast basics if using a page builder that does not warn about contrast issues.
- ☐ Client knows how to tab through the front end of their website to keyboard test it.
- ☐ Client knows how to use a screen reader or how to request screen reader testing assistance for any new content they add.
- ☐ Client knows how to evaluate new plugins for accessibility before/when adding new plugins or how to ask for help evaluating plugins for accessibility.

- ☐ Client has been offered a care plan or support plan that includes accessibility monitoring or ongoing assistance.