

Accessibility Checklist

This checklist aims to quickly help identify the most obvious accessibility issues for WCAG 2.0 AA compatibility. It is not intended to be exhaustive, but should act as way to quickly assess a page in a multi-pass approach. It would be recommended to use a screen reader such as VoiceOver when conducting the assessment.

1 Visual Design

Consider users who have sight impairments not necessarily just those using a screen reader.

Colour contrasts of text and links	<i>Use a colour contrast checker to validate contrast ratios</i>
Colour is not only way to communicate information	<i>Reinforce colours with clear text information</i>
Text-only zoom	<i>+200% zoom & -50% text zoom ensure text is legible, and there is no horizontal scrolling</i>
Pinch zoom (Touch devices)	<i>Pinch zoom must be available on touch devices, at least to x2</i>
Animation / flashing	<i>No content that flashes more than 3 times/second, prevents triggering seizures.</i>
Non-text content	<i>Ensure there are text-based alternatives to key content, doesn't apply to presentational content.</i>

2 Page Structure

Consider users who rely on a screen reader and need the page to be read out to them.

Appropriate and unique page title	<i>Keep the most page relevant text at the beginning</i>
Logical heading structure	<i>Ensure heading levels are not missed, i.e. h2, then a h4</i>
Meaningful headings & links	<i>Enough context? Check the screen reader's web rotor</i>
Semantic mark-up	<i>Is semantic mark-up used throughout the page so it makes sense? Usage of lists, nav sections, field sets, etc.</i>
Tabular data	<i>Tables should be used to mark-up tabular data, and appropriate scope and captions should be utilised.</i>
Progressive enhancement	<i>If JavaScript is disabled, can a user still complete the task?</i>
Image alt tags	<i>Remember if left empty, screen readers skip the image</i>

3 Keyboard Navigation

Consider screen reader users and those using keyboard-only or switch controls because of motor disabilities preventing them from using a mouse/track-pad.

Fully keyboard accessible	<i>All functions are available to keyboard only users</i>
Visible focus	<i>Focus outline is visible and not visually lost on the page</i>
No keyboard traps	<i>Focus is managed correctly where necessary</i>
Clear screen reader focus outline	<i>Screen reader focus outline clearly encloses the area it is currently announcing</i>
Focus management	<i>Focus is managed appropriately for errors, dialogs, form elements, non-native elements, etc.</i>
Focus order	<i>Focus order is clear and logical</i>
No context change on focus	<i>Element doesn't change context on focus, i.e. a drop down doesn't submit upon option selection</i>
Timing	<i>Give users enough time to complete an action, or the ability to extend the time frame by at least 10 times (exceptions are real-time events like auctions etc).</i>

4 Forms & Interactive Elements

Consider error management and providing feedback for interaction.

Links & Buttons	<i>Rule of thumb: links perform navigation, buttons perform actions</i>
Programmatic association of labels and form elements	<i>Labels should be linked programmatically with form elements, so they are announced correctly</i>
Clearly identify errors	<i>Error attributes on form elements (aria-invalid, aria-describedby) need to be maintained and managed in the appropriate states.</i>
Programmatically associated errors	<i>Errors for a form element should be linked programmatically, using aria-describedby</i>
Clear error instructions	<i>Error messages should clearly indicate how to resolve the issue.</i>
Custom widgets	<i>Non-native widgets need to have appropriate roles and behaviours manually applied to them, so screen readers can interpret them correctly, and keyboard only users can interact with them.</i>
Opening new windows	<i>When opening new windows, make it clear that the link opens in a new window</i>
Live regions	<i>Utilise live regions carefully, example applications could be when feedback is needed with complex in-page updates are occurring.</i>

5 Multimedia

Consider audio and visual content and how blind and deaf users will access this content.

Provide audio/video transcripts	<i>Text alternatives to time-based media</i>
Provide audio descriptions	<i>Have descriptions available for pre-recorded video</i>
Provide captions	<i>Have synced subtitles for audio/video content</i>
CAPTCHA user verification	<i>Avoid CAPTCHA if possible, must have an audio alternative if used</i>

Recommended WAI-ARIA usage

Try to utilise native HTML elements and attributes wherever possible, before using ARIA attributes.

For example, utilise native form elements wherever possible, otherwise when non-native elements are used a lot of extra work (primarily scripting) will need to be done to mimic native component behaviours to ensure a consistent and expected behaviour for all users.

Screen reader tips

Spend some time listening to the actual output of the screen reader audio, elements such as dates, times, prices, units of measure are not always announced as expected, ensure that this information is appropriately conveyed to those relying solely on the announcements.

Use the web rotor to review page structure and content out of context. Seasoned screen reader users hop around page content quickly, is the content appropriately landmarked and headings deployed to facilitate this behaviour?

Guides & Resources

Apple Voice Over Guide

<http://www.apple.com/voiceover/info/guide/>

Other accessibility audit checklists

<http://webaim.org/standards/wcag/checklist>

WCAG 2.0 at a glance

<http://www.w3.org/WAI/WCAG20/glance/>

WCAG 2.0 full spec

<http://www.w3.org/TR/WCAG20/>

Tools

Color Oracle – View screens through colour blind users' eyes

<http://colororacle.org>

Colour Contrast Analyser – Check contrast ratios meet WCAG 2.0 success criteria

<http://www.paciellogroup.com/resources/contrastanalyser/>

tota11y – Automated accessibility page review tool

<https://khan.github.io/tota11y/>

aXe: the Accessibility Engine - Automated accessibility page review tool

<http://www.deque.com/products/aXe/>

WAVE Toolbar/extensions – Automated accessibility page review tool

(Remember these tools only highlights certain programmatic errors, and can at best be used as a guide)

<http://wave.webaim.org/extension/>

Chrome

<https://wave.webaim.org/toolbar/>

Firefox