

WCAG 2.1 checklist Level A (Beginner, not indicated) and Level AA (Intermediary, indicated)

Date executed:

Note: * denotes introduced in 2.1 guidelines update

Based on content found at <https://www.wuhcag.com/> by Luke McGrath

Guideline	Summary / Example	Hints (generally all will include some form of visual review or inspection, some could be automated)
1.1.1 – Non-text Content	Provide text alternatives for non-text content E.g. Company logo, ALT="HML Company Logo" Longer descriptions can be used for graphs etc.	WAVE tool can flag Sensible descriptions, 'forest' not '1,534 trees of various types'
1.3.1 – Info and Relationships	Logical structure E.g. organise page using headings; convey relationships through mark up or text	WAVE Check Headings order, H1, H2 etc.
1.3.2 – Meaningful Sequence	Present content in a meaningful order	User experience, judgement call, usability
1.3.3 – Sensory Characteristics	Use more than one sense for instructions E.g. don't rely solely on shape, size or sound	E.g. error messages, focus, message, highlight field, colour
1.3.4* – Orientation AA	If a user wants a monitor or device to be landscape, the content displays in the same readable way as if the orientation is portrait. This does not apply if the orientation affects the performance or operability of the application, like a video.	Responsive design, real device testing required
1.3.5* – Identify Input Purpose AA	This expands on 4.1.2. Customisation of icons, fields or buttons are important to cognitively challenged users. For customising the icons, often on smaller screens, a lot of buttons are replaced with image icons, and what is intuitive to one user is not for another. By conforming to a standard, this will allow a user to be able to specify their own icons and meanings to minimise confusion.	Test compatibility with third party tools / browser add-on's (e.g. Stylish). Apply a different style and use should be unchanged. Note: Security removes autocomplete from scope
1.4.1 – Use of Colour	Don't use presentation that relies solely on colour	Visual Inspection of function focused on colour use
1.4.3 – Contrast (Minimum) AA	Contrast ratio between text and background is at least 4.5:1 Exceptions: Large text; logo or brand name	WAVE contrast check

1.4.4 – Resize Text AA	Text can be resized to 200% without loss of content or function (responsive)	Zoom, check on screen + - keyboard and mouse functions
1.4.5 – Images of Text AA	Use text rather than images of text unless essential E.g. text part of logo or brand is considered essential	Visual Inspection
1.4.10* - Reflow AA	Guidelines are suggested for simple content like text, and screens zoomed to 400 percent which could need two-directional scrolling. Images, videos, and data tables, are considered out-of-scope for this checkpoint. When the screen is zoomed in, text is expected to wrap to new lines so the user only requires one scroll direction to read the content. This checkpoint is important for limited vision users and mobile users.	Zoom and visual inspection for scroll bars, wrapping, text leaking etc.
1.4.11* - Non-Text Contrast AA	Updates for colour contrast include visual items that are required to be seen to understand the context. For instance, video controls needed to stop or start content will need a ratio of 3:1.	WAVE
1.4.12* - Text Spacing AA	Giving a limited vision or cognitively challenged user the ability to change the spacing and making sure the content is still readable informs this guideline. Challenges to developing and testing this kind of flexible content could be around making sure the various pieces of content do not overlap or cause odd layout issues.	Works with third party tools like EasyRead or Sentence Segmenter or text spacing bookmarklet (see appendix) Basically, use one of the above and check for overlaps, cut offs and container leaks
1.4.13* - Content on Hover or Focus AA	Allow a component which is given focus, to be closed by a keyboard or mouse. This allows the user to keep a tooltip open and move the mouse around it, without it closing.	Users with cognitive or physical challenges will see benefits from this guideline as sometimes a short display time is not enough to read a tooltip or pop-up.
2.1.1 – Keyboard	Accessible by keyboard only	Navigate through functionality using only keyboard. All functionality should be available and tasks can be completed.

2.1.2 – No Keyboard Trap	Don't trap keyboard users	A trap is where you can tab in but not back out of an area or you get stuck in a sub area. Confirm Tab and Shift Tab returns through page
2.1.4* - Character Key Shortcuts	Single key shortcuts should have a way to amend or disable them.	These shortcuts can interfere with speech recognition software and cause difficulties for keyboard only users
2.2.1 – Timing Adjustable	Time limits have user controls E.g. simple action to extend (press spacebar) Essential Exceptions: Time out for security	Confirm unless this falls under the exceptions
2.4.1 – Bypass Blocks	Provide a 'Skip to Content' link to bypass blocks of content repeated on multiple web pages E.g. The first interactive item in the Web page is a link to the beginning of the main content	Use tabbing to check if present.
2.4.2 – Page Titled	Use helpful and clear page titles E.g. Pages have titles that describe topic or purpose	Visual Inspection of titles
2.4.3 – Focus Order	Logical order, focusable components receive focus in an order that preserves meaning and operability.	Almost identical to tabbing order but visual indicators
2.4.4 – Link Purpose (In Context)	Every link's purpose is clear from its context E.g. Whenever possible, provide link text that identifies the purpose of the link without needing additional context	Visual Inspection. No 'click here' type links
2.4.5 – Multiple Ways AA	Offer several ways to find pages (menus, site map, links) Except if part of process	Visual Inspection
2.4.6 – Headings and Labels AA	Use clear headings and labels describing topic or purpose	Visual Inspection
2.4.7 – Focus Visible AA	Ensure keyboard focus is visible and clear E.g. link or control is highlighted when focused on, keyboard tab or mouse hover	Visual Inspection

<p>2.5.1* - Pointer Gestures</p>	<p>The use of single-point activation, not including gestures required to operate assistive technology, should help users with cognitive issues, or people not able to perform complicated activations. The simpler an interface, the easier for the majority to use. When a graphic requires a slider zoom action, it can be complemented with keyboard interaction with the + or – buttons.</p>	<p>Mouse, keyboard and Visual Inspection</p>
<p>2.5.2* - Pointer Cancellation</p>	<p>Users with visual, motor, and/or cognitive challenges have issues with accidental activation of features. Users should be able to reverse accidental activation of a feature and recover from it if necessary. This applies to most transactions but not when the timing is important, like an action game.</p>	<p>Mouse, keyboard and Visual Inspection</p>
<p>2.5.3* - Label in Name</p>	<p>Users that rely on speech input need labels for user interface components. For someone using a voice command tool, the named fields should match the labels the voice command tool will reference. It also helps users with cognitive challenges that rely on speech input reducing the cognitive load for remembering item names.</p>	<p>NVDA Screen Reader Visual Inspection of code using developer tools</p>
<p>2.5.4* - Motion Actuation</p>	<p>Navigating web content on a mobile phone can be hard for average users. Using the device to change content with a tilt motion gives the user more ways to control content, along with having next and previous page buttons available.</p>	<p>Real device testing of function</p>
<p>3.1.1 – Language of Page</p>	<p>Page has a language assigned and can be programmatically determined</p>	<p>WAVE</p>
<p>3.1.2 – Language of Parts AA</p>	<p>Tell users when the language on a page changes E.g. change is visually distinctive (bold, italic etc.)</p>	<p>WAVE</p>
<p>3.2.1 – On Focus</p>	<p>Elements do not change context when they receive focus</p>	<p>Visual Inspection of function</p>

3.2.2 – On Input	Elements do not change context when they receive input	Visual Inspection of function
3.2.3 – Consistent Navigation AA	Use menus and navigation consistently where repeated on multiple pages E.g. templates	Visual Inspection
3.2.4 – Consistent Identification AA	Use icons and buttons consistently, not necessarily identical	Visual Inspection
3.3.1 – Error Identification	Clearly identify input errors E.g. errors automatically detected are highlighted and described to the user in text	Visual Inspection of error handling
3.3.2 – Labels or Instructions	Label elements and give instructions when content requires user input	Visual Inspection of labels and instructions
3.3.3 – Error Suggestion AA	Suggest fixes when users make errors if solutions are known, unless it would jeopardise security E.g. jump to error; highlight missing mandatory field; help text	Visual Inspection of error handling
3.3.4- Error Prevention (Legal, Financial, Data) AA	Reduce the risk of input errors for sensitive data E.g. submissions are reversible; validate field entries; review before submission (you sure?)	Visual Inspection of error handling
4.1.1 – Parsing	No major code errors E.g. start and end tags, no duplicate attributes, unique IDs (if applicable)	WAVE Visual Inspection Code Review
4.1.2 – Name, Role, Value	Build all elements for accessibility E.g. supports assistive technology	WAVE Visual Inspection Code Review
4.1.3* - Status Messages AA	It is to notify low vision or blind users or allow a user that may not see the updates of items to be notified. If you've selected an item to buy, but after selecting a specific colour, and the size is no longer in stock, then a notification of the change is heard by the assistive technology and reported to the user. When searching for items and results are returned, without moving the focus, the user should hear the results returned from their screen reader application.	Visual Inspection of function

Appendix:

Many examples at: <http://www.w3.org/WAI/WCAG20/quickref/Overview.php>
Guide (POUR): 1 = Perceivable; 2 = Operable; 3 Understandable; 4 = Robust

The 12 new checkpoints in WCAG2.1 are;

- [1.3.4 - Orientation](#) - mobile and low or poor vision users.
- [1.3.5 - Identify Input Purpose](#) - screen reader and cognitively challenged users.
- [1.4.10 - Reflow](#) - mobile, low or poor vision and cognitively challenged users.
- [1.4.11 - Non-text Contrast](#) - low or poor vision users.
- [1.4.12 - Text Spacing](#) - low or poor vision users.
- [1.4.13 - Content on Hover or Focus](#) - low or poor vision and cognitively challenged users.
- [2.1.4 - Character Key Shortcuts](#) - keyboard and cognitively challenged users.
- [2.5.1 - Pointer Gestures](#) - mobile and cognitively challenged users.
- [2.5.2 - Pointer Cancellation](#) - mobile and cognitively challenged users.
- [2.5.3 - Label in Name](#) - cognitively challenged users.
- [2.5.4 - Motion Actuation](#) - keyboard mobile and cognitively challenged users.
- [4.1.3 - Status Messages](#) - screen reader and cognitively challenged users.

Useful info on keyboard expectations: <https://davatron5000.github.io/a11y-nutrition-cards/>

WAVE Tool can be a browser extension or available at <https://wave.webaim.org/>
(site needs to be publically published for the URL to work).

WAVE Chrome extension: <https://chrome.google.com/webstore/detail/wave-evaluation-tool/jbbplnpkjmmebjpifedlgcdilocofh>

WAVE Firefox extension: <https://addons.mozilla.org/en-US/firefox/addon/wave-accessibility-tool/>

NVDA Screen Reader: <https://webaim.org/resources/shortcuts/nvda>
<https://www.nvaccess.org/download/>

Bookmarklet: <https://www.html5accessibility.com/tests/tsbookmarklet.html>