



LEVEL
a c c e s s

The Must-Have WCAG Checklist

A practical resource for understanding the Web Content Accessibility Guidelines and assessing the accessibility of your website, web content, and web applications

Table of Contents

<u>What is WCAG?</u>	3
<u>WCAG 101</u>	4
<u>POUR principles</u>	4
<u>Various versions</u>	5
<u>What's new from WCAG 2.0 to 2.1?</u>	5
<u>Conformance levels</u>	6
<u>How high should you aim?</u>	7
<u>What's coming in WCAG 2.2?</u>	7
<u>Start testing</u>	8
<u>Automated testing</u>	8
<u>Manual and functional testing</u>	10
<u>WCAG 2.1 checklists</u>	11
<u>Level A</u>	11
<u>Level AA</u>	14
<u>Level AAA</u>	16
<u>We're here to help</u>	19

What is WCAG?

The Web Content Accessibility Guidelines, or WCAG, are a set of technical standards that, when followed, improve the accessibility of web content, websites, and web applications for people with a wide range of disabilities—including auditory, cognitive, neurological, physical, speech, and visual disabilities.

WCAG was created by the Web Accessibility Initiative of the W3C, the World Wide Web Consortium, which is a global community of accessibility experts striving to make the internet as inclusive as possible. WCAG provides a single, common, global standard for web accessibility, enabling designers, developers, and authors to remove barriers users with disabilities may encounter online. Importantly, following WCAG also helps organizations comply with various legal mandates, like the Americans with Disabilities Act (ADA), Section 508 of the Rehabilitation Act, the Accessibility for Ontarians with Disabilities Act (AODA), and others. In some

cases, to comply with legal requirements, web content must conform with WCAG standards.

So how well does your web content, website, or web application conform with WCAG?

This practical guide explains the different WCAG principles, versions, and conformance levels, and how to test for WCAG conformance. It also includes an interactive WCAG checklist to help you evaluate your current state of accessibility.

WCAG 101

Before evaluating your web content for WCAG conformance, it's important to understand the WCAG principles, the versions of WCAG that have been released, and the different conformance levels.

POUR principles

WCAG is organized by four main principles, which state that content must be perceivable, operable, understandable, and robust. They are often referred to by the acronym POUR. These principles can be applied to any kind of digital product or service, no matter the underlying technology:



Perceivable

Information and user interface components must be presentable to users in ways they can perceive. For example, it's important to present information that can be perceived in different ways, where a user can adjust color contrast or font size, or view captions for videos.



Operable

User interface components and navigation must be functional for users in ways they can operate. For example, a user must be able to perform required interactions using a keyboard or voice commands, not just using a mouse.



Understandable

Information and user interface operation must be understandable. For example, information and instructions should be clear and navigation methods should be easy to understand and use.



Robust

Content must be robust enough that it can be interpreted reliably by a wide variety of users and assistive technologies. As technologies evolve, code and content should remain accessible for users of common and current assistive devices and tools.

Various versions

WCAG exists in various versions: 1.0, 2.0, 2.1, and soon, 2.2. Updated versions are released to keep pace with changes in technology. The first version, known as WCAG 1.0, was released in 1999 and is no longer recommended for use. A later version, WCAG 2.0, came out in 2008. In June 2018, the W3C released WCAG 2.1, which builds upon the guidelines in 2.0, introducing additional success criteria related to newer technologies, and addressing a broader range of disability-related needs. The various versions of WCAG are backwards compatible, meaning the more recent version incorporates every success criteria from the earlier version, with additions. Therefore, if content conforms with WCAG 2.1, it also conforms with WCAG 2.0.

What's new from WCAG 2.0 to 2.1?

Mobile

- Improves support for touch interactions, keyboard, and mouse
- Avoids unintended activation of device sensors

Low vision

- Extends contrast requirements to graphics
- Improves text and layout adaptability

Cognitive and learning disabilities

- Enables more detailed description of page controls and elements to support personalization of user interfaces

Conformance levels

There are three levels of WCAG conformance: A, AA, and AAA. Each level builds on the previous level like a pyramid. So, in order to meet Level AA, you must meet all of Level A, and in order to meet Level AAA, you must meet all of Level AA.

- **Level A** = This level represents the base level of conformance. Level A criteria affect the broadest group of users with the most benefits and are essential. But, with the base level of support, some barriers will still exist that impact certain groups of users.
- **Level AA** = This level is the most common target conformance level, often adopted in regulations or negotiated in legal settlements. The criteria at this level establish a level of accessibility that works for more users, including those who use assistive technology.
- **Level AAA** = This is the highest conformance level achievable, meaning it covers the success criteria of all three levels. However, Level AAA is not applicable or realistic in all situations. Some organizations may choose to adopt specific criteria at this level.

Level	Success Criteria		
	WCAG 2.0	WCAG 2.1	Total WCAG 2.0 and 2.1
A The most basic web accessibility features	25	5	30
AA Deals with the biggest and most common barriers for users with disabilities	13	7	20
AAA The highest (and most complex) level of web accessibility	23	5	28
Total	61	17	78

How high should you aim?

The W3C encourages organizations to conform with the most recent version of WCAG as a best practice. While certain laws, such as the Accessibility for Ontarians with Disabilities Act (AODA) and Section 508 of the U.S. Rehabilitation Act of 1973, require conformance with WCAG 2.0 Level AA (which also requires conforming with the 2.0 Level A success criteria), the U.S. Department of Justice has referenced WCAG 2.1 Level AA (at minimum) as the benchmark in its more recent ADA enforcement actions. Amidst this inconsistency, conformance with the latest version will not only provide improved accessibility for every user, it will ensure your organization is up-to-date in its compliance efforts, especially as we anticipate future policy changes and updated legal rulemaking. Accessibility laws generally mandate conformance with Level A and AA success criteria because all of the technical standards in Level AAA may not be applicable or realistic in all situations.

What's coming in WCAG 2.2?

The W3C has released a [draft of WCAG 2.2](#), which includes nine new success criteria. These criteria focus on accessibility for users with low vision, cognitive and learning disabilities, and limited fine motor skills. The W3C is expected to release its final draft of 2.2 in 2023.

Start testing

To gauge WCAG conformance, first conduct an evaluation of your website or web content. This evaluation can consist of the step-by-step process of thoroughly and diligently testing whether that experience is usable by people with disabilities.

A comprehensive accessibility evaluation typically involves a combination of automated testing and extensive manual evaluation.

Automated testing

Automated testing (or scanning) is a great first step in the testing process. There are several tools to use to conduct automated testing, including our [free page scanner](#), and [free automatic and continuous testing tools](#). We recommend you determine which will work with your firewall settings, as well as for your design and developer teams. Quality assurance (QA) teams will likely leverage even more tools to ensure compliance and usability. Here is a listing of free automated testing tools for consideration as you get started:





Code validation

W3C CSS Validator software was created by the W3C to help web designers and web developers check Cascading Style Sheets (CSS). It can be used via the free web service, or downloaded and used either as a java program or as a java servlet on a web server. This tool allows a comparison of style sheets to the CSS specifications, and helps find errors, typos, and incorrect uses of CSS. It will also advise when the CSS poses some usability risks.



Color contrast

The WebAIM Contrast Checker is an online tool that enables users to enter the hex codes of foreground and background color. It will reveal the contrast ratio between the two, ensuring you meet WCAG recommendations. If the two colors do not have a high enough contrast, this tool enables you to adjust either color until the proper contrast is met, providing the new hex code. Our Accessible Color Picker for Chrome is also free.



Mobile accessibility

Two tools serve the mobile accessibility space. Accessibility Scanner checks the accessibility of Android apps. For iOS, Accessibility Inspector can be used. Both apps are utilized by developer and QA teams.



Document accessibility

The Document Accessibility Toolbar is a dedicated accessibility ribbon menu for Microsoft Word that makes it quicker and easier to create accessible documents. This toolbar features a range of hand-picked and custom-built functions to optimize and validate a document for accessibility.

The PDF Accessibility Checker PAC allows for the checking of PDF accessibility. It works even for those who do not have Adobe Acrobat Professional.



Web accessibility

The WAVE tool by WebAIM scans a URL, delivering a report that uses a simple red, yellow, or green icon to show errors, warnings, and elements that pass. It also has a built-in ARIA checker and color contrast analyzer, and enables a user to turn on/off style sheets.

Manual and functional testing

Manual and functional testing will build upon automated testing results. In this process, accessibility experts, which should include people with disabilities, check website features and flows using assistive technology. This evaluation will confirm or dismiss any issues reported in an automated scan, as well as identify any new issues that should be resolved.

Getting started with manual testing

There are several ways to accomplish manual testing:

Option 1

Build an in-house team of accessibility testers to perform QA on digital assets in development.

Option 2

Hire an outside consultant to provide a one-time report outlining issues identified and barriers encountered.

Option 3 (recommended)

Work with an accessibility partner on an ongoing basis. A partner should provide automated testing as well as give you access to an expert team to manually check digital assets in multiple environments using different assistive technologies. These partners will also work with you to develop prioritization reports outlining the critical, high, medium, and low-level issues; monitor your digital properties on an ongoing basis; and integrate seamlessly into your backend systems for better team collaboration.

WCAG 2.1 A, AA, and AAA checklist

As you test your content, or partner with a company to conduct testing, the following interactive WCAG checklist will serve as a helpful tracking guide for applicable success criteria:

WCAG 2.1 Level A checklist

Success Criteria	Description	Pass/Fail	Complete
<u>1.1.1 – Non-text Content</u>	Provide text alternatives for non-text content	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>
<u>1.2.1 – Audio-only and Video-only (Pre-recorded)</u>	Provide an alternative to video-only and audio-only content	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>
<u>1.2.2 – Captions (Pre-recorded)</u>	Provide captions for videos with audio	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>
<u>1.2.3 – Audio Description or Media Alternative (Pre-recorded)</u>	Video with audio has a second alternative	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>
<u>1.3.1 – Info and Relationships</u>	Logical structures	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>
<u>1.3.2 – Meaningful Sequence</u>	Present content in a meaningful order	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>
<u>1.3.3 – Sensory Characteristics</u>	Use more than one sense for instructions	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>
<u>1.4.1 – Use of Color</u>	Don't use presentation that relies solely on color	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>

WCAG 2.1 Level A checklist *(continued)*

Success Criteria	Description	Pass/Fail	Complete
<u>1.4.2 – Audio Control</u>	Don't play audio automatically	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>
<u>2.1.1 – Keyboard</u>	Accessible by keyboard only	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>
<u>2.1.2 – No Keyboard Trap</u>	Don't trap keyboard users	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>
<u>2.1.4 – Character Key Shortcuts</u>	Do not use single-key shortcuts, or provide a way to turn them off or change them	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>
<u>2.2.1 – Timing Adjustable</u>	Time limits have user controls	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>
<u>2.2.2 – Pause, Stop, Hide</u>	Provide user controls for moving content	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>
<u>2.3.1 – Three Flashes or Below</u>	No content flashes more than three times per second, or the flash is below the general flash and red flash thresholds	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>
<u>2.4.1 – Bypass Blocks</u>	Provide a "Skip to Content" link	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>
<u>2.4.2 – Page Titled</u>	Helpful and clear page title	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>
<u>2.4.3 – Focus Order</u>	Logical order	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>
<u>2.4.4 – Link Purpose (In Context)</u>	Every link's purpose is clear from its context	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>

WCAG 2.1 Level A checklist *(continued)*

Success Criteria	Description	Pass/Fail	Complete
<u>2.5.1 – Pointer Gestures</u>	Users can perform touch functions with assistive technology or one finger	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>
<u>2.5.2 – Pointer Cancellation</u>	This requirement applies to web content that interprets pointer actions	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>
<u>2.5.3 – Label in Name</u>	The name contains the text that is presented visually	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>
<u>2.5.4 – Motion Actuation</u>	Functions that are triggered by moving a device or by gesturing towards a device can also be operated by more conventional user interface components	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>
<u>3.1.1 – Language of Page</u>	Page has a language assigned	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>
<u>3.2.1 – On Focus</u>	Elements do not change when they receive focus	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>
<u>3.2.2 – On Input</u>	Elements do not change when they receive input	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>
<u>3.3.1 – Error Identification</u>	Clearly identify input errors	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>
<u>3.3.2 – Labels or Instructions</u>	Label elements and give instructions	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>
<u>4.1.1 – Parsing</u>	No major code errors	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>
<u>4.1.2 – Name, Role, Value</u>	Build all elements for accessibility	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>

WCAG 2.1 Level AA checklist

Success Criteria	Description	Pass/Fail	Complete
<u>1.2.4 – Captions (Live)</u>	Live videos have captions	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>
<u>1.2.5 – Audio Description (Pre-recorded)</u>	Users have access to audio description for video content	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>
<u>1.3.4 – Orientation</u>	Authors do not rely on a screen orientation	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>
<u>1.3.5 – Identify Input Purpose</u>	Ensure common names are provided using the HTML autocomplete list	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>
<u>1.4.3 – Contrast (Minimum)</u>	Contrast ratio between text and background is at least 4.5:1	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>
<u>1.4.4 – Resize Text</u>	Text can be resized to 200% without loss of content or function	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>
<u>1.4.5 – Images of Text</u>	Don't use images of text	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>
<u>1.4.10 – Reflow</u>	Your website must be responsive	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>
<u>1.4.11 – Non-Text Contrast</u>	Minimum of 3:1 color contrast ratio for user interface components and states and graphical objects that convey meaningful information	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>
<u>1.4.12 – Text Spacing</u>	Text spacing can be overridden to improve the reading experience	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>

WCAG 2.1 Level AA checklist *(continued)*

Success Criteria	Description	Pass/Fail	Complete
<u>1.4.13 – Content on Hover Focus</u>	Content visible on hover or keyboard focus does not lead to accessibility issues	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>
<u>2.4.5 – Multiple Ways</u>	Offer several ways to find pages	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>
<u>2.4.6 – Headings and Labels</u>	Use clear headings and labels	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>
<u>2.4.7 – Focus Visible</u>	Keyboard focus is visible and clear	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>
<u>3.1.2 – Language of Parts</u>	Tell users when the language on a page changes	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>
<u>3.2.3 – Consistent Navigation</u>	Use menus consistently	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>
<u>3.2.4 – Consistent Identification</u>	Use icons and buttons consistently	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>
<u>3.3.3 – Error Suggestion</u>	Suggest fixes when users make errors	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>
<u>3.3.4 – Error Prevention (Legal, Financial, Data)</u>	Reduce the risk of input errors for sensitive data	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>
<u>4.1.3 – Status Messages</u>	Status messages can be presented to the user by AT without receiving focus	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>

WCAG 2.1 Level AAA checklist

Success Criteria	Description	Pass/Fail	Complete
<u>1.2.6 – Sign Language (Pre-recorded)</u>	Provide sign language translations for videos	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>
<u>1.2.7 – Extended Audio Description (Pre-recorded)</u>	Provide extended audio description for videos	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>
<u>1.2.8 – Media Alternative (Pre-recorded)</u>	Provide a text alternative to videos	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>
<u>1.2.9 – Audio-only (Live)</u>	Provide alternatives for live audio	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>
<u>1.3.6 – Identify Purpose</u>	Anticipates the release of cognitive metadata to be used with assistive technology to simplify interfaces	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>
<u>1.4.6 – Contrast (Enhanced)</u>	Contrast ratio between text and background is at least 7:1	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>
<u>1.4.7 – Low or No Background Audio</u>	Audio is clear for listeners to hear	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>
<u>1.4.8 – Visual Presentation</u>	Offer users a range of presentation options	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>
<u>1.4.9 – Images of Text (No Exception)</u>	Don't use images of text	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>

WCAG 2.1 Level AAA checklist *(continued)*

Success Criteria	Description	Pass/Fail	Complete
<u>2.1.3 – Keyboard (No Exception)</u>	Accessible by keyboard only, without exception	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>
<u>2.2.3 – No Timing</u>	No time limits	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>
<u>2.2.4 – Interruptions</u>	Don't interrupt users	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>
<u>2.2.5 – Re-authenticating</u>	Save user data when re-authenticating	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>
<u>2.2.6 – Timeouts</u>	Users need to be warned of the duration of any inactivity that could cause data loss	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>
<u>2.3.2 – Three Flashes</u>	No content flashes more than three times per second	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>
<u>2.3.3 – Animation from Interactions</u>	Motion animation triggered by interaction can be deactivated	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>
<u>2.4.8 – Location</u>	Let users know where they are	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>
<u>2.4.9 – Link Purpose (Link Only)</u>	Every link's purpose is clear from its text	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>
<u>2.4.10 – Section Headings</u>	Break up content with headings	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>
<u>2.5.5 – Target Size</u>	The size of the target for pointer inputs is at least 44 x 44 CSS pixels	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>

WCAG 2.1 Level AAA checklist *(continued)*

Success Criteria	Description	Pass/Fail	Complete
<u>2.5.6 – Concurrent Input Mechanism</u>	Web content does not restrict use of input modalities available on a platform	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>
<u>3.1.3 – Unusual Words</u>	Explain any strange words	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>
<u>3.1.4 – Abbreviations</u>	Explain any abbreviations	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>
<u>3.1.5 – Reading Level</u>	Users with nine years of school can read your content	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>
<u>3.1.6 – Pronunciation</u>	Explain any words that are hard to pronounce	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>
<u>3.2.5 – Change on Request</u>	Don't change elements until users ask	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>
<u>3.3.5 – Help</u>	Provide detailed help and instructions	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>
<u>3.3.6 – Error Prevention (All)</u>	Reduce the risk of all input errors	<input type="radio"/> <input type="radio"/>	<input type="checkbox"/>

Let's get started

Whether you want help evaluating the state of accessibility for your web content, website, or web application, or you're ready to make them accessible, Level Access is here to help. Our comprehensive accessibility platform combines tooling, testing, and technology with access to expert services and training, enabling you to meet your accessibility objectives and scale with confidence.

[Request a demo](#) of our solution today.

www.levelaccess.com
info@levelaccess.com

