



DIY Digital Accessibility Compliance Guide

The DIY Digital Accessibility Compliance Guide was created to help you improve your brand and customer loyalty, Search Engine Optimization, and achieve digital accessibility compliance!

Table of Contents

Here's Why Digital Accessibility Is Important To You & Your Business.....	3
DIY Digital Accessibility Compliance Guide Methodology.....	3
How To Learn Where And What Digital Accessibility Violations Are In Your Code.....	3
How To Perform Automated Digital Accessibility Testing.....	4
How To Perform Manual Digital Accessibility Testing	6
How To Test For Accessibility Violations & Get Compliant.....	8
1. Informative Images & Alternative Text Descriptions.....	9
2. Nested Headings For Structure & Organization	10
3. Adequate Color Contrast	12
4. Form Fields & Appropriate Labels	12
5. Captions, Audio Descriptions, & Transcripts	13
6. Data Table Accessibility	14
7. Descriptive Page Titles	15
8. Website Link Accessibility	15
9. Accessible Notifications and Alerts	16
10. Keyboard Navigation & Interactive Elements.....	17
11. Name Your Frames Appropriately	18
12. Avoid Keyboard Traps.....	19
13. Accessible Skip Links	20
14. Set Your Website's Language	21
15. Avoid Content That Causes Seizures	21
16. Enable Visitors To Report Digital Accessibility Issues	22

Here's Why Digital Accessibility Is Important To You & Your Business

Brand, SEO, compliance, costs.

- Organizations inclusive of people with disabilities are more liked and trusted by customers
- Digital accessibility improves your website's SEO, conversions, and overall UX
- Accessible code is clean, compliant code that mitigates your risks
- Bundling marketing services with compliance lowers your monthly costs

[Talk to an Expert - Get a Quote](#)

DIY Digital Accessibility Compliance Guide Methodology

This Guide contains the specific areas you will need to work on to improve digital accessibility compliance according to the Americans with Disabilities Act (ADA) and the Web Content Accessibility Guidelines (WCAG).

3 things must be accomplished to bring your website into digital accessibility compliance:

1. Learn where and what accessibility violations are in your code
2. Know how to fix the code
3. Make the change to your code.

How To Learn Where And What Digital Accessibility Violations Are In Your Code

You must test your website in 2 ways to learn where and what digital accessibility violations are in your code:

1. Automated digital accessibility testing
2. Manual digital accessibility testing

*Note. A combination of automated testing software and manual testing approaches are required to ensure you achieve digital accessibility compliance. This is detailed below.

Automated Digital Accessibility Testing

Fixing digital accessibility violations you find through automated digital accessibility testing is the best first step to protect yourself from a digital accessibility lawsuit. This helps you quickly identify and fix the "low-hanging" digital accessibility violations that can be found using

software—the primary tool used by the lawyers of plaintiffs in “drive-by” digital accessibility lawsuits. Nearly 5000 digital accessibility lawsuits are filed each year. The average settlement cost is between \$4000–\$7500 plus legal fees.

The increase in lawsuits is driven by greater awareness of digital accessibility issues, increased regulatory focus, and a growing number of plaintiff law firms specializing in this area. The anticipation for 2024 and 2025 is a continued upward trend, with a significant number of new lawsuits filed every week as businesses continue to grapple with compliance.

There are free and paid automated digital accessibility testing tools. All are browser extensions, but some have additional user interfaces and or capabilities.

Download one or more of the below browser extensions for automated digital accessibility testing. Then run accessibility tests across your website starting with the homepage.

[Digital Accessibility Report & Test Suite \(DARTS\)](#)

[WAVE Evaluation Tool](#)

[Accessibility Insights](#)

*Note. You may need help finding the right color contrast with your brand colors. The [Accessible Web Color Contrast Checker](#) is a free tool that can help. Color contrast ratios are covered in detail later.

How To Perform Automated Digital Accessibility Testing

If you have hundreds of webpages, focus on the pages most critical to your website’s intended purpose and function. For example, if you have an online store, then run tests on your product pages, shopping cart, and checkout pages. If you have a content-driven marketing strategy, you will want to be sure to run automated tests on every new release, campaign, and blog or vlog post.

Prioritize fixing the most severe digital accessibility compliance violations first. These have the greatest impact on your visitors with disabilities. Additionally, these violations usually have the biggest impact on your website’s UX, SEO, and conversion rates.

Most severe accessibility violations, also known as “road blocking” violations, are found through manual digital accessibility testing. The reason is because automated testing only tests the static state of your website at any given time. Once a visitor begins interacting with your website, it begins to dynamically change. It is here, after the website has changed, by which many severe and “road blocking” accessibility violations present themselves.

Manual Digital Accessibility Testing

Manual Digital Accessibility testing takes 2 forms—testing with assistive technology and manual keyboard testing.

There are various types of assistive technologies for sensory, motor, and cognitive disabilities, but the most efficient and comprehensive tool for manual digital accessibility testing is a screen reader. Screen readers are used by the blind and dyslexic to read content on websites, as well as perform other daily tasks such as shopping, checking email, social media, banking, etc.

NonVisual Desktop Access, commonly known as NVDA, is free, the most widely used screen reader in the world, and AccessAbility Officer's recommended screen reader for manual digital accessibility testing on the desktop.

[Download NVDA here.](#)

VoiceOver is Apple's built-in screen reader on Mac OS and is used by less than 10% of screen reader users. More importantly for digital accessibility compliance, VoiceOver has browser compatibility issues with Chrome and Firefox that create test data reliability and validity issues. VoiceOver is strongest testing websites in the Safari browser. Turn on VoiceOver with the keystroke combination of 'Command + F5'.

Narrator is Microsoft's built-in screen reader for Windows and provides auditory feedback to users who are blind or have low vision. It reads aloud the text on the screen and describes actions, such as opening apps, navigating the web, and interacting with different elements in the interface. Narrator is a considerably basic screen reader and is not recommended for use in manual testing digital accessibility compliance. Turn on Narrator with the keystroke combination of 'Control + Windows Key + Enter'.

JAWS, the second most popular screen reader in the world, is a paid screen reader software for Windows. JAWS is a strong screen reader for manual digital accessibility testing and does offer a "free" 40-minute version that allows users to run JAWS for 40 minutes before having to restart their computer in order to use it again. You can [download a free version of JAWS here.](#)

For manual digital accessibility testing on mobile, VoiceOver on the iPhone is the recommended and most widely used test environment on mobile. However, Android's accessibility features for those with mobility disabilities are making Android devices more and more popular for individuals with significant disabilities like complete paralysis or ALS (Amyotrophic Lateral Sclerosis).

How To Perform Manual Digital Accessibility Testing

An experienced [digital accessibility agency](#) will be able to perform comprehensive digital accessibility tests, remediate accessibility violations for you, and managed accessibility compliance on-going. Comprehensive digital accessibility testing that would be performed by an experienced [digital accessibility agency](#) is more complicated and complex than what is found in the DIY Digital Accessibility Compliance Guide. This approach includes [certified experts](#) and [certified testers with disabilities](#) performing automated accessibility tests, manual screen reader tests, and keyboard navigation tests.

Basic manual digital accessibility testing can be done using just a few different keystrokes and combinations. When performing manual accessibility testing on a website using NVDA or keyboard navigation (no NVDA or other assistive technology), the following keystrokes and combinations will be essential for navigating and interacting with your website.

1. Tab

Function: Moves the focus forward to the next interactive element on the page.

Purpose: Helps test the focus order of interactive elements like links, buttons, and form fields. This ensures that users who navigate by keyboard can move logically through the content in a meaningful sequence. It also allows testers to check whether all interactive elements are reachable by keyboard alone.

Wait! What's "Focus"? Good question.

When an element has focus, it typically receives a visible outline or highlight, known as a "focus indicator." This visual cue shows users where they are on the page and which element they are interacting with. When a button has focus, it might have a colored border, a box shadow, or some other change in appearance to indicate its focus state. Interactive elements like links, buttons, form fields, and menu items can receive focus. Non-interactive elements like plain text or images without links generally cannot.

2. Shift + Tab Combination

Function: Moves the focus backward to the previous interactive element.

Purpose: Allows users to navigate in reverse order through interactive elements. This is important for ensuring that keyboard users can move backward as well as forward, which is essential for usability and accessibility, especially if a mistake is made or the user needs to recheck something.

3. Alt + Down Arrow & Alt + Up Arrow Combinations

Function: Opens and closes dropdown menus or selects the next/previous option in a dropdown.

Purpose: Used to interact with dropdowns in forms or navigation menus. Testing this ensures that dropdowns are accessible and that users can navigate through options using the keyboard.

4. Up and Down Arrow Keys

Function: Moves the focus or selection upwards and downwards within a list, menu, or other navigable elements.

Purpose: Used to navigate vertically within dropdowns, lists, menus, or any other interactive element that supports up and down navigation. It's important for testing that users can navigate these elements efficiently and that the correct item is selected when navigating using up and down arrows.

5. Left & Right Arrow Keys

Function: Moves focus or selection horizontally within navigable elements, such as sliders, carousels, or tabs.

Purpose: Enables users to navigate within horizontal elements like tabbed interfaces or sliders. Testing ensures that users can easily move between items in such controls.

6. Enter

Function: Activates the currently focused element.

Purpose: The Enter key is used to interact with elements like buttons, links, and form submissions. It is crucial to test that pressing Enter activates the correct action, such as submitting a form, following a link, or triggering a button's action.

7. Spacebar

Function: Activates buttons and toggles checkboxes or other toggleable elements.

Purpose: The Spacebar is often used to select or toggle elements such as checkboxes, radio buttons, or buttons. Testing this ensures that these elements can be activated without a mouse and that their state changes as expected when toggled.

8. Escape

Function: Closes modals, dropdowns, or other interactive elements that can be dismissed.

Purpose: The Escape key is commonly used to exit or close modal windows, dropdowns, and other overlay elements. Testing this ensures that users can easily dismiss these elements and return to the main content without requiring a mouse, which is critical for maintaining flow and usability.

Now that you understand the basic mechanics for how to perform automated and manual digital accessibility testing, let's put this into action.

How To Test For Accessibility Violations & Get Compliant

To meet and maintain digital accessibility compliance, you must know which success criteria to test, how to test, and then how to fix those issues.

There are 50 success criteria your website will need to comply with according to WCAG 2.1 AA under Title II of the ADA, but 16 areas have been strategically prioritized in the DIY Digital Accessibility Compliance Guide for businesses to focus on.

The 16 focus areas of the DIY Digital Accessibility Compliance Guide were prioritized according to:

1. Level of impact on ADA and WCAG compliance
2. The severity impact violations have on users, and
3. The level of difficulty to find and fix violations.

WCAG Standard	2.0	2.1	2.2
Year Established	2008	2018	2023
Success Criteria	38	50	57
Who's Responsible	All businesses	All businesses	No businesses yet
Settlement Cost	\$4000-\$7500*	\$4000-\$7500*	\$0 - no case law

*Does not include legal fees

All public facing websites owned by businesses that can be accessed in the United States are responsible for meeting and maintaining WCAG 2.1 AA compliance. Case law for WCAG 2.2 compliance is expected in 2025.

All public-facing Government agency websites currently must adhere to WCAG 2.0 AA, but have been mandated to bring their websites into compliance with WCAG 2.1 AA by 2025.

1. Informative Images & Alternative Text Descriptions

Alt tags provide alternative text descriptions for informative images that convey information required for a user to understand its purpose and function. Individuals who are blind, print disabled, and or have a cognitive disability can benefit from alt tags through the additional context and clarification provided. These tags and image descriptions make the content more understandable and reduce cognitive load.

How To Perform Digital Accessibility Testing For Alternative Text Descriptions

You can use accessibility automation software to quickly check which images on your website do not have alt tags, but automated testing cannot determine if the text description of the image is adequate to achieve digital accessibility compliance. This requires a manual check by a human.

Using NVDA and a mouse, simply bring the NVDA cursor close to the image by clicking right before it, either just above or just to the right. Then press the down arrow key to listen for NVDA to navigate to and read the image's alt tag. If there is an image description, it will be read.

If you hear a file name that begins with `https://...` Or "blank", then the image's alt tag is missing an appropriate text description, which is required to achieve ADA and WCAG digital accessibility compliance.

How To Write Alt Tags & Text Descriptions For Your Images

Deciding whether an image needs an alt tag with an informative text description or if it should be marked as decorative depends on the image's purpose and context within the content.

If the image conveys essential information that is not fully provided by the surrounding text, it requires an informative alt text attribute. Good examples include diagrams, charts, or graphs, photos that contribute to the meaning or context of the content, and icons or buttons that trigger actions like search or submit buttons. In this instance, insert a clear, concise description in the image's alt attribute that conveys the image's purpose or meaning.

If the image is purely decorative and does not add any meaningful information to the content, it should be marked as null. For example, background images, images used for layout purposes, and other decorative images should be marked as null.

To mark the image as decorative, use an empty alt attribute (`alt=""`). Notice nothing between the quotes. This is an empty alt tag. This empty alt tag enables screen readers to skip over unimportant images that do not functionally contribute to the user's experience, the website's purpose, or function.

For complex images like infographics or charts where a detailed explanation is needed, the alt text may not be sufficient to convey all the information. Instead, the alt text should briefly describe the image and refer users to a longer description elsewhere on the page or provide the key takeaway from the complex image that the author wants the user to know within the tag.

Your Alt Tag & Text Description Digital Accessibility Checklist

- ✓ If informative, use descriptive alt text.
- ✓ If decorative/redundant, use `alt=""` (null alt text).
- ✓ If functional, use alt text to describe the function.
- ✓ If complex, provide a brief alt text and refer to a more detailed description or provide the key takeaway right in the tag.

2. Nested Headings For Structure & Organization

Nested headings are a way to organize content on a webpage in a logical, hierarchical structure from 1 to 6. Similar to the headings in The DIY Digital Accessibility Compliance Guide, headings within a webpage help both people and search engines understand where content is located and how it is organized.

Correct all heading accessibility violations and you will significantly improve SEO as well as navigation and orientation for all visitors. Here's how to organize your heading structure.

How To Organize Your Heading Structure

Every page on your website should have a single H1 heading. This is the main title of the page, similar to the title of a book or chapter. It tells users what the page is about. Usually your page's title will match the page's H1 heading.

For example: `<h1>Main Title</h1>` or `<h1>Healthy Eating</h1>`

Use H2 for major sections across each page. For example, if your page's title H1 heading is "Healthy Eating," an example list of H2 headings might be "Benefits of Healthy Eating," "Healthy At-Home Recipes," and "Eating & Traveling Healthier."

For example: `<h2>Major Section3</h2>` or `<h2>Eating & Traveling Healthier</h2>`

Use H3 for subsections within an H2 section. If you have a H2 titled “Benefits of Healthy Eating,” an appropriate list of H3 heading sections might be “Physical Benefits of Healthy Eating,” “Mental Benefits of Healthy Eating,” “Financial & Family Benefits of Healthy Eating.”

For example: `<h3>subheading2</h3>` or `<h3>Mental Benefits of Healthy Eating</h3>`

If you are compelled to break down your content further, use H4, H5, and H6 headings. However, the need is rare to go beyond H3 on most websites. However deep your heading structure is, the important thing is to maintain logical order.

No skipsies! Following a logical heading order means do not jump from using an H2 heading down to an H4 heading without having an H3 heading in between. Note the example above. Your headings should flow down each page naturally, as if following the table of contents.

How To Perform Digital Accessibility Testing On Nested Heading Structures

Tools like [Digital Accessibility Report & Test Suite \(DARTS\)](#), [WAVE Evaluation Tool](#), and [Accessibility Insights](#) are effective at evaluating the structure and accessibility of headings in a webpage, but automated accessibility tools have limitations of identifying visual headings that are styled using CSS.

CSS visual styling doesn't provide the semantic HTML tags these tools check for in the code, so automated accessibility tools cannot detect these visual headings. Even though your website looks stunning, we need to perform both automated and manual testing.

First, run an automated accessibility test, locate your website's heading level and role accessibility violations, and then fix them.

Next up is manual testing of your headings. Here's what you need to do:

1. Inspect the page's HTML code to determine if the heading is correctly tagged.

Right-click on the text that looks like a heading and select "Inspect" or "Inspect Element" from the context menu. This will open the browser's developer tools and highlight the HTML element associated with the text.

If the text is visually styled like a heading but is not within a proper heading tag – (For example `<h1>Title Heading Text Here</h1>` or `<h2> Major Section Heading Text Here</h2>`, etc.) it's a strong indication that CSS is being used to create a visual heading–For example, if the text is inside a `<div>`, `<span>`, `<p>`, or any other non-heading element.

2. Test if NVDA reads the heading's programmatic level.

Click your mouse near the heading to bring NVDA's focus there, and then use up and or down arrows to determine the headings level and content. If a screen reader doesn't recognize the text as a heading, it's likely that CSS is being used to style it rather than proper heading tags.

3. Adequate Color Contrast

Appropriate color contrast is required to achieve digital accessibility compliance as poor color contrast makes text difficult to distinguish from the background, leading to eyestrain, fatigue, and decreased readability.

How To Perform Digital Accessibility Testing For Color Contrast

You can determine if you have a color contrast violation using the [Digital Accessibility Report & Test Suite \(DARTS\)](#) and you can find which specific colors for your brand will help you achieve compliance standards with The [Accessible Web Color Contrast Checker](#).

These tools analyze the color values of text and background elements and calculate the color contrast ratio to determine if Your website is compliant with the ADA and WCAG. WCAG states websites should have a minimum color contrast of 4.5:1 between text and the background. For larger text, the contrast should be 3:1.

The 2 Contrast Ratios For Digital Accessibility Compliance

The color contrast ratio of normal text or small text, which is defined as text smaller than 18 points or 14 points if bold, is 4.5:1.

The color contrast ratio for large text and non-text elements is 3:1. Large-scale text is defined as text 18 points and larger or 14 points and larger if bold. Non-text elements are defined as visual elements conveying meaning like icons, graphs, and buttons, as well as UI components that need to be distinguished from other elements.

4. Form Fields & Appropriate Labels

Forms during the checkout, webinar registration, and newsletter signup process can be difficult or impossible for individuals to complete if they do not have appropriate labels describing their purpose or function.

Missing or unclear labels lead to errors and frustration for individuals with disabilities, especially those reliant on assistive technology like screen readers.

How To Perform Digital Accessibility Testing On Form Fields

To determine whether your forms are compliant with the ADA and WCAG, you need to perform both automated accessibility tests and manual accessibility testing with a screen

reader. Automated digital accessibility testing will help you determine if any of your form fields are missing programmatic labels. While automated accessibility testing will find form fields with missing labels, it will not help you determine if those form field labels are descriptive and appropriate for their purpose and function, which is required to be ADA and WCAG compliant.

After you have determined each of your form field elements do have programmatic labels, including text fields, check boxes, radial buttons, dropdown menus, and buttons, you must confirm manually that each visual form field label matches the programmatic label.

Ensuring your visual and programmatic form field labels match enables blind individuals using a screen reader and those with motor disabilities using speech-to-text software to navigate through, complete, and correctly submit the forms on your website.

To confirm the programmatic label matches a field's visual label, navigate through your website's form with the Tab key and 'Shift + Tab' key combination described in the section "How To Perform Manual Digital Accessibility Testing."

As you Tab through the form using the Tab key and 'Shift + Tab' key combination with NVDA, the visual focus will cycle through the form and will read the programmatic label of each form field. Confirm the visual label (text on the screen) matches what the NVDA screen reader is announcing as it navigates to a form element (the programmatic label).

5. Captions, Audio Descriptions, & Transcripts

If your website has videos or audio content like podcasts, to achieve ADA and WCAG compliance, you need to ensure your multi-media has accurate captions, audio descriptions, and transcripts. These accessibility features make your content more easily consumed by a wide range of visitors, including those with and without disabilities.

While designed for the deaf and hard of hearing, captions allow those with sight to watch your video content in a crowded or private environment where they would not have been able to otherwise. Captions need to be provided for all prerecorded audio content in videos and should accurately reflect the spoken content. Sounds that contribute to understanding the content, such as sound effects, speaker identification, and music that conveys meaning must also be included in captioning. This ensures users who rely on captions have access to the same information as those who hear the audio directly.

Captioning services can create synchronized text transcripts of spoken dialogue and sound effects for your video content. You can embed these captions directly into your videos or provide them as separate text files.

Audio descriptions provide additional narration that describes key visual content in a video. While designed for the blind to understand visual-only information and context in videos, audio descriptions also help individuals with cognitive impairments and second language learners better understand what they are watching. Audio descriptions should capture essential visual details such as physical actions, settings, and facial expressions, and should be synchronized appropriately with the video.

Audio descriptions can be implemented as separate audio tracks or directly embedded in your videos.

For prerecorded audio-only or video-only content, a transcript or media alternative should be provided. This ensures that users who cannot access the audio or video directly can still receive the same information. Transcripts need to be accurate and complete, which includes all spoken content, sound effects, and descriptions of visual content if the transcript is for a video.

6. Data Table Accessibility

Column and row headers in tables primarily benefit people with visual impairments, including those who are blind or have low vision, but they also improve your SEO by helping search engines understand the content within your website.

By ensuring your data tables have row and column headers, you will better achieve digital accessibility compliance, enable screen readers to better navigate data tables, and they will better understand the relationship of data within your tables.

How To Perform Digital Accessibility Testing For Data Tables

To determine if your website has ADA and WCAG compliant data tables, you will need to use both automated and manual digital accessibility testing techniques.

Automated digital accessibility tools like [Digital Accessibility Report & Test Suite \(DARTS\)](#) will help identify if your tables are missing programmatic row and column headers. Use [NVDA](#) to validate that your data table row and column headers are functioning correctly. You can Tab and 'Shift + Tab' through data tables to understand if your row and column headers are correct.

If your data tables are digitally accessible and programmatically correct, NVDA will announce the new row or column the user is navigating into prior to reading the content within the cell the user is navigating into. This enables a screen reader user to know where they are within the table and the information of the cell at the same time.

7. Descriptive Page Titles

Descriptive page titles provide your visitors with meaningful information about the content on your website, which makes navigation easier for individuals with disabilities.

Without a descriptive page title, understanding the purpose, context, and navigation of each webpage on your website is increasingly difficult for people with disabilities.

How To Perform Digital Accessibility Testing For Descriptive Page Titles

Review the title tags across each page of your website. Are your titles clear and relevant to the content of the page? Would Your visitors know exactly what type of content they can expect if they haven't visited the page before?

Some example page titles that are easily recognizable that visitors would immediately know what content is on the page without visiting the page before would include:

About Us. Visitors would immediately know that the content on this page is about your company. For example, here is AccessAbility Officer's About Us page:

<https://accessabilityofficer.com/about>

Blog. Simple and straight forward, visitors immediately would understand they are on the page where your company's blog is located. For example, here is AccessAbility Officer's blog:

<https://accessabilityofficer.com/blog>

Contact. This page title clearly indicates the purpose is to contact your business. For an example, here is AccessAbility Officer's Contact page:

<https://accessabilityofficer.com/contact>

8. Website Link Accessibility

Ensuring your website's links have appropriate text labels is required for digital accessibility compliance, but it also improves your website's SEO by enabling search engines to navigate your website, understand its content, and structure.

To ensure your website achieves and maintains digital accessibility compliance, you will want to provide unique and contextual labels for all your website's links. These labels should describe to users where they are going and provide insight on the content the user will experience at that link's destination.

Some of the most common examples of links not being labeled correctly include "click here", "read more", and empty or blank link text. Often found on product and blog pages, "click here"

and “read more” link descriptions do not tell a user distinctly where they are going or what information will be on the page when they go there. Links that have no visible text or use non-informative characters like dashes or spaces can confuse users and screen readers.

How To Perform Digital Accessibility Testing Of Your Website’s Links

You can check whether your website’s links have labels using automated digital accessibility tools like [Digital Accessibility Report & Test Suite \(DARTS\)](#), but in order to determine if links have unique descriptive labels that achieve ADA and WCAG compliance, you will need to manually review each link.

As you go through your website, test whether your links clearly describe the destination. Here are 3 common examples of inaccessible and accessible link labels.

- ✗ Inaccessible Link #1: “Click Here”
- ✓ Accessible Link #1: “Click here for more information on our summer shorts collection!”

- ✗ Inaccessible Link #2: “Read More”
- ✓ Accessible Link #2: “Read more about Digital Accessibility Compliance violations”

- ✗ Inaccessible Link #3: Download”
- ✓ Accessible Link #3: “Download the DIY Digital Accessibility Compliance Guide”

9. Accessible Notifications and Alerts

Accessible notifications and alerts are essential for ensuring that all users can interact effectively with web content, including form submission confirmations, error messages, and status updates like “Newsletter signup successful,” “Item added to cart,” and the always fun “Error Submitting the Form. Please try again.”

Add ARIA live regions to HTML elements to ensure that dynamic content changes are announced by screen readers, such as `aria-live="polite"` or `aria-live="assertive"`, which control how and when updates are announced.

Polite updates are announced when the user is idle, but assertive updates are announced immediately. This interrupts the current task as it is deemed as an urgent notification.

Ensure notifications are structured in a meaningful and contextual manner. Error messages should be associated directly with the form fields they relate to, and success messages should clearly indicate the action that was successfully completed.

For example:

“Error! Please correct the First Name and Zip Code fields before trying again.”

“3 Items Successfully Added to Cart!”

How to Perform Digital Accessibility Testing for Notifications and Alerts

With NVDA turned on, perform a task that triggers a notification. Verify NVDA clearly announces the notification, that the notification was relevant to the action that was just completed, and or the action that needs to be completed next.

Use DevTools (press F12) to inspect elements and ensure that appropriate ARIA attributes are implemented. Then, trigger the live region content and observe how it’s announced by screen readers.

Ensure that notifications are reachable and readable by users who navigate using the keyboard alone. Test to confirm that focus does not get trapped or skipped over critical notifications. Also ensure that visual notifications have sufficient contrast against the background according to the contrast ratios established in the section “How To Perform Digital Accessibility Testing For Contrast Ratios.”

Some notifications may need to persist until the user dismisses them, while others can be temporary. Be mindful of how long notifications stay visible and provide an accessible method to dismiss them.

10. Keyboard Navigation & Interactive Elements

Keyboard navigation allows users to navigate through web pages, interact with elements, and access content without the need for a mouse or other device.

Without accessible keyboard navigation, individuals may be unable to perform essential tasks such as filling out forms, selecting options from drop-down menus, or accessing links and buttons, effectively excluding them from accessing your site’s content and services.

A website lacking keyboard navigation impacts several types of people:

- Technical resources, engineers, and developers who prefer using the keyboard over a mouse.
- Individuals reliant on screen readers or magnification software who navigate websites using keyboard shortcuts and Tab keys.
- Individuals who are unable to use a mouse due to an amputation, paralysis, or a motor impairment.

How To Perform Digital Accessibility Testing Using Keyboard Navigation

Using the keystrokes and combinations in the DIY Digital Accessibility Compliance Guide's section on "How To Manually Test For Digital Accessibility", you can test whether your website is keyboard accessible. Here are the instructions:

1. Navigate with the Tab key through links, buttons, form fields, and interactive elements on the page. When you navigate to a given interactive element, ensure each element receives the visible focus indicator.
2. Use the Enter and Spacebar keys to activate buttons, links, sliders, check boxes, and radial buttons. Validate the status of each interactive element appropriately changes when pressing the Enter and spacebar keys.
3. Use the Tab key and 'Shift + Tab' key combination to test for Tab order. Validate the visual focus of the Tab order has a logical sequence that matches the visual layout of your website.

11. Name Your Frames Appropriately

Frames, including `<iframe>` elements, are commonly used in web development to embed content from other sources, such as videos, maps, or documents, within a webpage.

To achieve digital accessibility compliance, each frame should have a clear, descriptive name that accurately reflects its content or purpose. For example, a frame that contains a map should be named "Map of Our Location," rather than a generic name like "frame1" or "iframe2."

Use the title attribute of the `<iframe>` element to provide a name to your frames. These names are what screen readers and other assistive technologies will announce when users navigate to the frame.

How to Perform Digital Accessibility Testing On Frames

1. Use DevTools (Press F12) to inspect the HTML of the webpage. Look for all `<iframe>` elements and check if they have a title attribute. Verify that each title attribute is filled with descriptive and relevant text.
2. Navigate the webpage using a screen reader and listen to how frames are announced. The screen reader should clearly articulate the frame's purpose based on its name. If the screen reader reads something generic like "frame" without additional context, it indicates that the frame needs to be properly named.
3. Test the navigation experience using only the keyboard. Ensure that when focus shifts to a frame, the name provided makes sense and accurately describes the content or function of that frame.

Properly named frames also have SEO benefits because search engines can better understand and index the content within named frames. This is what potentially improves your webpage's visibility and search ranking.

If a frame contains nested frames—a frame within a frame—each nested frame should also be appropriately named. This ensures that users can understand the hierarchical structure of the content and navigate through it effectively.

Be cautious when embedding third-party content using `<iframe>` elements. Ensure that the embedded content is accessible and that you provide a clear and descriptive name for the frame. If the third-party content itself is not accessible, consider providing an alternative accessible version of the content.

In cases where an iframe might fail to load or is blocked by certain browser settings, provide fallback content or a descriptive message within the iframe element itself. This ensures users are informed even if the embedded content cannot be displayed.

12. Avoid Keyboard Traps

Keyboard traps occur when a keyboard user navigates into a particular part of a webpage and can't get out of it, preventing them from moving through to the rest of the page.

This is extremely frustrating for those with motor impairments and blindness who often exclusively use the keyboard to navigate websites on their desktop.

How to Perform Digital Accessibility Testing For Keyboard Traps

Upon page load, start navigating using the Tab key. Tab through all interactive elements—links, buttons, and form fields—to ensure that the focus moves logically and does not get stuck in any particular area.

If you reach an area like a modal window, dropdown menu, or iframe where you can't move focus out by pressing Tab or 'Shift + Tab', you've encountered a keyboard trap.

Digital Accessibility Testing For Modal Windows and Pop-Ups

Pay special attention to modal windows, pop-ups, and overlays. Ensure that when a modal opens, the focus is correctly set on the first interactive element within the modal and that users can navigate out of the modal using the keyboard.

When the modal is closed, focus should return to the element that triggered the modal. Test this by opening and closing the modal using the keyboard and verifying that focus returns to the correct place.

After keyboard navigation testing, use a screen reader to navigate the site and ensure that the focus does not get trapped in any specific area. This helps confirm that users who rely on assistive technologies can navigate freely without encountering keyboard traps.

How To Fix & Avoid Keyboard Traps

1. Ensure your focus can move out of interactive elements.

When designing interactive elements like modal windows or dropdown menus, ensure that focus can move out of these areas using both the Tab and 'Shift + Tab' keys. This might involve adjusting the JavaScript handling focus events or rethinking how elements are coded.

2. Use ARIA attributes to manage focus within modal windows and pop-ups.

`aria-modal="true"` can help manage focus within a modal, ensuring that users can navigate correctly without getting trapped.

3. Be cautious with auto-focus features.

If an element is automatically focused on page load, ensure that users can navigate away from it without getting stuck. Quickly test this by pressing the Escape key.

4. Be cautious when setting `tabindex` on elements as improper use can create traps.

Avoid setting `tabindex` on non-interactive elements and ensure that interactive elements are correctly included in the natural Tab order.

13. Accessible Skip Links

Skip links are typically placed at the very beginning of a webpage, allowing users to bypass repetitive navigation menus and jump directly to the main content.

Long and complex navigation tables are exceptionally frustrating for keyboard users, blind screen reader users, and people with disabilities who find it cumbersome to Tab through numerous navigation links before reaching the primary content of your website.

Imagine being forced to manually Tab through every link and button from the top of the YouTube homepage before you ever got to your video feed. Wait... Don't.

A skip link at the top of the page, which is usually hidden until it receives focus, allows them to bypass this step and go directly to your content before they pull their hair out.

Skip links should be one of the first elements on the page, typically within the header. They should be coded so that they are the first focusable element when a user presses the Tab key. It's essential to ensure that when skip links become focused, they become fully visible.

This visual indication should be clear and easily distinguishable, helping users know that they can use the skip link to navigate directly to the content.

Make the link text for your skip links descriptive according to their purpose. Instead of a generic “Skip to content,” it could be more specific, like “Skip to main content” or “Skip to product listings,” depending on the page's structure.

How to Perform Digital Accessibility Testing for Skip Links

Upon page load, press the Tab key to begin moving focus. The skip link should appear as the first focusable element. When you press Enter or Spacebar, it should immediately move the focus to the main content, skipping over the navigation menu.

Check that the styling is distinct and contrasts well with the background when the skip link becomes visible and receives focus. This will help make it easy for users to understand its purpose to skip to the main content.

Skip links are most commonly used on desktop websites, but it's also important to consider their importance for accessibility on mobile devices, where navigation may also be cumbersome for users with disabilities.

Consider providing multiple skip links on big pages with complex structures, such as a skip to content, skip to search, or skip to footer. This gives users more control over their navigation experience.

14. Set Your Website's Language

Setting the language of your website ensures screen readers and other assistive technologies use the correct pronunciation rules and language-specific features when reading the content out loud to the user. Blind, cognitively impaired, dyslexic, and otherwise print disabled individuals benefit from ensuring your website's language is set correctly.

How To Test Your Website's Language Settings

You can check whether your website has set a language for a page by inspecting the HTML code of your site. Look for the lang attribute within the <html> element, which specifies the language of the document.

15. Avoid Content That Causes Seizures

Web content that flashes rapidly or presents intense visual stimuli can trigger seizures in individuals with photosensitive epilepsy. This condition is particularly sensitive to content that flashes more than three times per second with high contrast between flashes.

The WCAG guidelines specify that content should not flash more than three times per second. However, even if the content flashes less than three times per second, it's important to avoid high-contrast flashes that could be problematic.

Instead of flashing content, consider static images, gentle animations, or color changes that do not involve flashing.

How to Perform Digital Accessibility Testing for Content That Causes Seizures

Manually review your content for any elements that flash, flicker, or rapidly change. Pay special attention to videos, animated GIFs, and CSS animations.

Use tools like PEAT to analyze your content for seizure-inducing patterns. PEAT evaluates video content, animations, and other visual elements to ensure they comply with accessibility standards.

16. Enable Visitors To Report Digital Accessibility Issues

User feedback is some of the best data businesses can source. Digital accessibility compliance issues reported by your customers provide you a valuable opportunity to improve the accessibility of your website, enhance the user experience for all users, and demonstrate responsiveness to the needs of users with disabilities.

You can enable users to provide this feedback in various ways. Here's what's most common for businesses:

- Contact forms

Provide a dedicated contact form on your site where users can submit reports of accessibility issues.

- Email

Allow users to send emails to a designated accessibility contact or support address to report issues and provide feedback.

- Online chat or support

Offer live chat support or online support services where users can communicate directly with a representative to report accessibility issues in real-time.

- Social media

Encourage users to report accessibility issues through your social media channels by providing specific hashtags or handles for accessibility-related inquiries and feedback.

DIY Digital Accessibility Compliance Guide Conclusion

Following the best-practices in the DIY Digital Accessibility Compliance Guide will help you better achieve ADA and WCAG compliance, mitigate legal risks, improve your brand loyalty with your customers, website SEO, and much more!

However, if you're feeling a little like digital accessibility compliance may be too much for a DIY effort...

Ever considered bundling your other marketing services with digital accessibility compliance?

[Talk to an Expert - Get a Quote](#)

Over 80% of organizations will save money by bundling their marketing services and digital accessibility compliance with AccessAbility Officer.

DIY Digital Accessibility Compliance Guide Disclaimer

This DIY digital accessibility Compliance guide is an informational resource. Following the directions herein does not guarantee any website, mobile app, or electronic file will be ADA and or WCAG compliant. These digital accessibility testing best-practices are organized to support businesses to independently achieve and maintain a reasonable level of digital accessibility compliance.