



Top 10 Tips for Making Your Website Accessible

1. Choose a content management system that supports accessibility.

There are many content management systems available to help you build your website. Common examples include Drupal and WordPress, but there are many other options available.

Once you've chosen a CMS that suits your needs, make sure to choose a theme/template that is accessible. Consult the theme's documentation for notes on accessibility and tips for creating accessible content and layouts for that theme. Be sure to follow the same guidelines when selecting modules, plugins, or widgets.

For elements like editing toolbars and video players, make sure that they support creating accessible content. For example, editing toolbars should include options for headings and accessible tables, and video players should include closed captioning. The CMS administration options (such as creating a blog post or posting a comment) should be accessible as well.

2. Use headings correctly to organize the structure of your content.

Screen reader users can use heading structure to navigate content. By using headings (<h1>, <h2>, etc.) correctly and strategically, the content of your website will be well-organized and easily interpreted by screen readers.

Be sure to adhere to the correct order of headings, and separate presentation from structure by using CSS (Cascading Style Sheets). Do not pick a header just because it looks good visually (which can confuse screen reader users); instead, create a new CSS class to style your text.

Examples of proper use of headings:

- Use <h1> for the primary title of the page. Avoid using an <h1> for anything other than the title of the website and the title of individual pages.
- Use headings to indicate and organize your content structure.
- Do not skip heading levels (e.g., go from an <h1> to an <h3>), as screen reader users will wonder if content is missing.

3. Include proper alt text for images.

Alt text should be provided for images, so that screen reader users can understand the message conveyed by the use of images on the page. This is especially important for informative images (such as infographics). When creating the alt text, the text should contain the message you wish to convey through that image, and if the image includes text, that text should also be included in the alt.

The exception to this rule is when an image is used purely for decoration; in this case, the alt text can be left empty so that the screen reader user is not distracted from the more important content on the page.

If an image is the only content of a link, the screen reader will read the file name if alt text is not provided. Always provide alt text for images that are used as links.

4. Give your links unique and descriptive names.

When including links in your content, use text that properly describes where the link will go. Using "click here" is not considered descriptive, and is ineffective for a screen reader user.

Just like sighted users scan the page for linked text, visually-impaired users can use their screen readers to scan for links. As a result, screen reader users often do not read the link within the context of the rest of the page. Using descriptive text properly explains the context of links to the screen reader user.

The most unique content of the link should be presented first, as screen reader users will often navigate the links list by searching via the first letter.

For example, if you are pointing visitors to a page called "About Us":

- Try not to say: "[Click here](#) to read about our company."
- Instead, say: "To learn more about our company, read [About Us](#)."

5. Use color with care.

The most common form of color deficiency, red-green color deficiency, affects approximately 8% of the population. Using ONLY colors such as these (especially to indicate required fields in a form) will prevent these individuals from understanding your message.

Other groups of people with disabilities, particularly users with learning disabilities, benefit greatly from color when used to distinguish and organize your content.

To satisfy both groups, use color, but also be sure to use other visual indicators, such as an asterisk or question mark. Be sure to also distinguish blocks of content from one another using visual separation (such as whitespace or borders).

There are [several tools](#) you can use to evaluate your website including color contrast, which will assist you in making your page as visually usable as possible to individuals with low vision or varying levels of color blindness.

6. Design your forms for accessibility.

When form fields are not labeled appropriately, the screen reader user does not have the same cues available as the sighted user. It may be impossible to tell what type of content should be entered into a form field.

Each field in your form should have a well-positioned, descriptive label. For example, if the field is for a person's name, it should be labeled appropriately as either "Full Name" or have two separate fields labeled as "First Name" and "Last

Name." Use the <label> tag or an ARIA property (see tip #9) to associate the label text with the form field.

As you are going through a form field, a person should be able to tab through the form and fill out all the fields before getting to the "Submit" button, or they may not even realize that additional fields exist. Essentially, the tab order should follow the visual order.

If you have fields that are related or similar, consider grouping them together using field sets. For example, fields like "Full Name" and "Date of Birth" could be grouped together as "Personal Information." This type of form organization can help a screen reader user keep track of progress, and can provide the context that might be lost while filling out the form.

If certain form fields are required, the field should be labeled accordingly, and configured to alert the screen reader user. Commonly, required fields are noted as such with an asterisk, which will not be spoken by some screen readers. Asterisks (or similar visual indications) should still be used for sighted users, people with learning disabilities or people who speak English as a second language. Indicating that a field is required to a screen reader can be accomplished by adding ARIA required="true" and ARIA required="false" for optional fields. After submitting the form, user will need to be alerted to submission confirmation and any submission errors. We recommend including any error counts in the page title (after the user has submitted), so the user will immediately be informed that there are errors on the page. If a user submits a form with errors, the user should be brought to a submission page that indicates what the errors are, and provides an easy way to navigate to those errors.

Finally, the use of CAPTCHA is inaccessible and should not be used to validate submissions. [WebAIM has a helpful summary of accessible alternatives \(link is external\)](#) to CAPTCHA to keep forms free of spam submissions.

7. Use tables for tabular data, not for layout.

Using tables for page layout adds additional verbosity to screen reader users. Whenever a screen reader encounters a table, the user is informed that there is a table with "x" number of columns and rows, which distracts from the content.

Also, the content may be read in an order that does not match the visual order of the page. Do not create the layout of a website using a table; instead, use CSS for presentation.

When a data table is necessary (i.e. you have a set of data that is best interpreted in a table format, such as a bank statement), use headers for rows and columns, which helps explain the relationships of cells. Complex tables may have several cells within the table that have a unique relationship to each other, and these should be identified by using the "scope" attribute in HTML. Table captions (HTML5) can be used to give additional information to users about how best to read and understand the table relationships.

8. Ensure that all content can be accessed with the keyboard alone in a logical way.

Users with mobility disabilities, including repetitive stress injuries, may not be able to use a mouse or trackpad. These people are able to access content through the use of a keyboard by pressing the "tab" or "arrow" keys, or through the use of alternative input devices such as single-switch input or mouth stick. As a result, the tab order should match the visual order, so keyboard-only users are able to logically navigate through site content.

Long pages with lots of content should be broken up with anchor links (jump lists), allowing keyboard-only users to skip to relevant portions of the page without having to negotiate through other content. "Skip to main content" should be provided at the top of each page, so keyboard-only users won't have to tab through the page navigation in order to get the main content.

For pages with local menus and multiple levels and sub-items, the menus should be configured so that all menu items can be accessed with the keyboard. Do not use elements that only activate when a user hovers over items with a mouse, as keyboard-only or screen readers users will not be able to activate them.

9. Use ARIA roles and landmarks.

ARIA (Accessible Rich Internet Applications) is an easy and powerful technical specification for ensuring your site structure is accessible. By assigning ARIA roles and landmarks to web elements, you enhance the ability of screen reader users to navigate and interact with your content. ARIA roles and landmarks can be easily added to your HTML, in the same way that you add classes to HTML in order to load attributes from CSS.

Examples of common ARIA usage include:

- Roles of "navigation" and "menu": Used to identify the site's primary navigation and individual page menus.
- Landmarks of "banner" and "main content": Used to identify the header and main content sections of a webpage.
- Alerts of "live" and "atomic": Used to help screen reader users with dynamic page changes, such as stock tickers and search filters.
- Forms: ARIA allows the addition of descriptive text to a form field ("described by" or "labeled by") and the identification of buttons and required controls (more information under tip #6).

10. Make dynamic content accessible.

When content updates dynamically (i.e. without a page refresh), screen readers may not be aware. This includes screen overlays, lightboxes, in-page updates, popups, and modal dialogs. Keyboard-only users may be trapped in page overlays. Magnification software users might be zoomed in on the wrong section of the page.

These functions can easily be made accessible. Options include ARIA roles and alerts, as well as front-end development frameworks that specifically support accessibility.

Ensure that video players do not auto-play (non-consensual sound), and that the players can be used with a keyboard. Additionally, all videos must have options for closed captioning and transcripts for the hearing-impaired.

If your site contains a slideshow, make sure that each photo has alt text and can be navigated via the keyboard. If you are using any unique widgets (such as a calendar picker or drag-and-drops), be sure to test for accessibility.

Online Accessibility Tools & Resources:

W3C: [Accessibility Evaluation Tools List](#)

Tools & Resources when keeping [Web Accessibility in Mind](#)

Website Accessibility Evaluation Tool: [WAVE](#)

The Paciello Group: [Colour Contrast Analyser](#)

The Paciello Group: [Accessibility Viewer](#) (aViewer)

Readability Evaluation Tools: [Juicy Studio](#) (Gunning-Fog Index)

[HTML Code Sniffer](#)

Access for All: [PDF Accessibility Checker](#)