

Accessibility Compliance Best Practices

HTML Structure and Style

The best screen readers use natural voices and alter tone and inflection based on HTML tags, so choose layout elements carefully. Properly structured semantic HTML is far easier to interpret than poorly structured HTML. For example, text contained within a `` or `<div>` element has no semantic structure, while proper use of headings and paragraphs is far easier to understand.

Text markup should always use proper semantic structure. This means elements are used for their intended purpose. In the context of Americans with Disabilities Act (ADA) compliance, pay particular attention to and utilize text elements such as headings and paragraphs.

When a browser encounters an `<h1>` heading, it interprets that tag to mean that the contents of the `<h1>` element constitute the most important heading of the section that contains the element. The semantic meaning of an `<h1>` element is that it is used to identify the most important header of a specific page or section.

Menus and Navigation

Navigation is significantly slower when a customer uses a screen reader. Customers who don't use a screen reader can find links quickly and can navigate to sought items, often without having to do any reading at all. Minimizing graphics also helps shorten reading times and speed navigation for customers who rely on assistive tools such as screen readers.

High-Contrast Colors

Use high-contrast ratios for text to ensure that it's visible to customers with visual perception difficulties. The visual presentation of text and images of text should have a contrast ratio of at least 4.5:1.

You can use the [Contrast-Ratio](#) online calculator when choosing colors to use. Insert two colors and a ratio is calculated.

Font Size

Except for captions and images of text, text can be resized without assistive technology up to 200% without loss of content or functionality.

Images

Whenever possible, provide text alternatives for any non-text content so that it can be changed into other forms people need, such as large print, braille, speech, symbols, or simpler language.

This is best accomplished with the use of the `alt` attribute on most images. The `alt` descriptions describe the purpose of the object(s). This *does not* need to be done for decorative or purely stylistic graphics with no function or interactivity.

If an image is also used as a link, ensure the `alt` attribute describes the graphic and the link destination.

Complex graphics should be accompanied by detailed text descriptions. For most clients, this applies to hero images, especially if they convey information specific to a deal or sale. Whenever possible, use an HTML action with properly structured semantic HTML content to create hero replacement content and use a background image for styling.

Links and Buttons

All links should be able to be navigated to using the keyboard. In most browsers, customers should be able to use the Tab key to select the link and the Enter key to click it. A link, `<a>`, should be used if defining the `href` attribute. This applies to external links (to another page) or internal links (to another section on the same page).

The purpose of the link should be readily understood based on the link text. Often, customers with screen readers skim through a document going from link to link without reading the words in between. Avoid link phrases such as *Click here* that are ambiguous out of context. Linked images should include `alt` text to provide link descriptions.

Use a `<button>` tag to call a JavaScript function or method such as apply filter, sort page, or submit form. Buttons should be interpretable based on their text alone or within content when applicable.

Avoid using `<div>`, `<p>`, ``, etc. for click-through actions.

Forms and Inputs

Inputs should use placeholder text and corresponding `<label>` element and `alt` attribute. When form controls are text input fields, use the `<label>` element. When text is not available, use the `title` attribute.

Ensure that form fields are in a logical tab order. You can use the `tabindex` attribute that allows the developer to customize the tabbing navigation order of a document and enables a tabbing order that differs from the default source code order. Elements that aren't normally tab-navigable (such as paragraphs) are then able to receive tab focus. Include any special instructions within field labels.

Video

Accessible videos include captions, a written transcript, an audio description, and are delivered using an accessible media player. When using videos on your site, consider creating a link to the video rather than embedding the media item directly onto the page. Where possible, include a link to the media player download.

Audio

Add text descriptions to any audio files and create a text transcript for video and audio files. Add an additional link to the text transcript.

Avoid Flashing or Strobing Content

Avoid using strobe effects or repeatedly or rapidly flashing images, colors, or animations. These can trigger seizures in people who have epilepsy.

Key Takeaways

- Customers should be able to reach all actionable content using only a keyboard ([tabindex](#)).
- Use semantic-valid HTML5 (properly tiered and structured with readable content).
- All non-text content should use [alt](#) attributes with valid descriptions with very few exceptions.
- Test everything! Numerous tools exist to test for compliance, but Google's [Screen Reader](#) and [axe DevTools – Web Accessibility Testing](#) for Chrome are useful tools to check for accessibility.