

## WDD 231

[Home](#)[W1](#)[W2](#)[W3](#)[W4](#)[W5](#)[W6](#)[W7](#)

# Web Accessibility

## Overview

Web accessibility is the practice of making websites usable by people of all abilities, for example, including those that have difficulty viewing a traditional screen or using a mouse. This learning activity focuses on the basics of web accessibility and the tools that are available to help create accessible websites.

## Prepare

It is important to ensure that the user experience of your website works well for all of your users. This includes making sure that each image contains appropriate alt text and that there is enough contrast among the colors you use so the text can be easily read. You need to make sure the font sizes can be adjusted, interactive controls can be navigated using the keyboard, and that your page follows proper use of HTML tags such as the title, h1-h6 elements, header, nav, main, and so forth.

The **DevTools Lighthouse** report contains a number of checks in the **Accessibility** category to help you verify that your page is accessible. The following items are checked by the **DevTools Lighthouse – Accessibility** category:

Background and foreground colors have a sufficient contrast ratio.

The contrast ratio can be checked in detail using **DevTools CSS Overview** or via a online tool like [Contrast Ratio](#). The contrast ratio is important because it helps ensure that text and other content is readable.

All **images** `<img>` have **alt** attribute content that accurately describes the image's purpose. The **alt** attribute is important because it provides a text alternative for images, which is essential for when screen readers are being used or when image loading has been turned off or is slow.

**Heading** elements (**h1**, **h2**, **h3**, etc.) are in sequentially-descending order. Headings give semantic structure to the document to make it easier to navigate and understand when using assistive technologies. This semantic structure is important for screen readers and other assistive technologies to help users navigate the page.

All interactive controls are **keyboard** focusable. When using a keyboard, the user should be able to navigate through all interactive elements on the page, such as links, buttons, and form fields. This is important for users not using a mouse or other pointing device.

The document has a logical **tab** order. The tab order is the order in which the focus moves when the user presses the **Tab** key. The tab order should follow a logical sequence that matches the visual layout of the page. This is important for users who rely on keyboard navigation to access content.

Simple landmark, such as **header**, **aside**, **footer**, **nav**, **main**, etc., elements are used instead of **divs**. Keep the landmark use basic and consistent.

Offscreen content is hidden with **display: none** or **aria-hidden=true**. Offscreen content is a way to provide additional information about content while not being visible on the screen. It is important for to include this information for screen readers and other assistive technologies to help users understand the content.

Text is resizable. All users should be able to resize text up to 200% without loss of content or functionality.

Document has a meaningful and relevant **title**. The title of the document should be descriptive and relevant to the content of the page to provide accurate information to users and search engines.

Page has a language **lang** attribute defined in the opening **html** tag. The lang attribute assignment is important for browsers and assistive technologies to understand the language of the content on the page.

Page has a **favicon**. The favicon provides a visual representation of the page and helps users quickly identify the page in their browser tabs or bookmarks.

All forms elements have **labels** that are tied to them. Form labels are essential to provide explicit context for form controls and to provide a clickable area for users to select the form control.

Tables have **headers**. Most tables that present data should use **<th>** elements to identify the header cells for assistive technologies users to understand the content of the table.

Video and audio have **captions** in order to support the text alternative for audio and video content for those users needing it. .

A lot of these items are technical aspects and developers need to remember that they are just tools to aid in the overall accessibility evaluation of a page or site. The Lighthouse accessibility tools will not evaluate the usability of a page with certain disabilities, nor if the content is understood, or what the user's experience is as they navigate and interact with the page and site. These aspects

often require manual testing and feedback from users with disabilities to truly understand and improve accessibility.

## Activity Instructions

Use **DevTools Lighthouse** to run an accessibility audit of all your pages. It will provide a list of accessibility issues and offer suggestions or improvement. You should already be familiar with using this tool as part of your normal testing and deployment workflow, so there is no additional activity to complete.

You should consistently be using tools like Lighthouse to analyze the pages you design and develop for areas of improvement including accessibility and performance. You are expected to do this and other testing on your individual website project(s) without being assigned to do so.

### Optional Resources

[Accessibility](#) – MDN

The [Web Content Accessibility Guidelines \(WCAG\)](#) is the standard for web accessibility. The WCAG guidelines are organized into three levels of conformance: A, AA, and AAA.

The WCAG guidelines are also available in a [Quick Reference](#) format.

[How To Do an Accessibility Review](#) – Chrome for Developers

---

**Back**

---

Copyright © Brigham Young University-Idaho | All rights reserved