

MDN's new design is in Beta! A sneak peek: <https://blog.mozilla.org/opendesign/mdns-new-design-beta/>


Learn web development

Accessibility

Learning some HTML, CSS, and JavaScript is useful if you want to become a web developer, but your knowledge needs to go further than just using the technologies — you need to use them responsibly so that you maximize the audience for your websites and don't lock anyone out of using them. To achieve this, you need to adhere to general best practices (which are demonstrated throughout the [HTML](#), [CSS](#), and [JavaScript](#) topics), do [cross browser testing](#), and consider accessibility from the start. In this module we'll cover the latter in detail.

Prerequisites

To get the most out of this module, it would be a good idea to either work through at least the first two modules of the [HTML](#), [CSS](#), and [JavaScript](#) topics, or perhaps even better, work through the relevant parts of the accessibility module as you work through the related technology topics.

 **Note:** If you are working on a computer/tablet/other device where you don't have the ability to create your own files, you can try out most of the code examples in an online coding program such as [JSBin](#) or [Thimble](#).

Guides

What is accessibility?

This article starts the module off with a good look at what accessibility actually is — this includes what groups of people we need to consider and why, what tools different people use to interact with the web, and how we can make accessibility part of our web development workflow.

HTML: A good basis for accessibility

A great deal of web content can be made accessible just by making sure the correct HTML elements are used for the correct purpose at all times. This article looks in detail at how HTML can be used to ensure maximum accessibility.

CSS and JavaScript accessibility best practices

CSS and JavaScript, when used properly, also have the potential to allow for accessible web experiences, but if misused they can significantly harm accessibility. This article outlines some CSS and JavaScript best practices that should be considered to ensure that even complex content is as accessible as possible.

WAI-ARIA basics

Following on from the previous article, sometimes making complex UI controls that involve unsemantic HTML and dynamic JavaScript-updated content can be difficult. WAI-ARIA is a technology that can help with such problems by adding in further semantics that browsers and assistive technologies can recognize and use to let users know what is going on. Here we'll show how to use it at a basic level to improve accessibility.

Accessible multimedia

Another category of content that can create accessibility problems is multimedia — video, audio, and image content need to be given proper textual alternatives so they can be understood by assistive technologies and their users. This article shows how.

Mobile accessibility

With web access on mobile devices being so popular, and popular platforms such as iOS and Android having fully-fledged accessibility tools, it is important to consider the accessibility of your web content on these platforms. This article looks at mobile-specific accessibility considerations.

Assessments

Accessibility troubleshooting

In the assessment for this module, we present to you a simple site with a number of accessibility issues that you need to diagnose and fix.

See also

- [Start Building Accessible Web Applications Today](#) — an excellent series of video tutorials by Marcy Sutton.
- [Deque University resources](#) — includes code examples, screen reader references, and other useful resources.
- [WebAIM resources](#) — includes guides, checklists, tools, and more.

Was this article helpful?



Learn the best of web development



Get the latest and greatest from MDN delivered straight to your inbox.

SIGN UP NOW

