

Accessibility

Contents

Overview and limitations	1
Structural markup	1
Interactive components	1
Color contrast	2
Visually hidden content	2
Reduced motion	2
Additional resources	3

Bootstrap provides an easy-to-use framework of ready-made styles, layout tools, and interactive components, allowing developers to create websites and applications that are visually appealing, functionally rich, and accessible out of the box.

Overview and limitations

The overall accessibility of any project built with Bootstrap depends in large part on the author’s markup, additional styling, and scripting they’ve included. However, provided that these have been implemented correctly, it should be perfectly possible to create websites and applications with Bootstrap that fulfill WCAG 2.1 (A/AA/AAA), Section 508, and similar accessibility standards and requirements.

Structural markup

Bootstrap’s styling and layout can be applied to a wide range of markup structures. This documentation aims to provide developers with best practice examples to demonstrate the use of Bootstrap itself and illustrate appropriate semantic markup, including ways in which potential accessibility concerns can be addressed.

Interactive components

Bootstrap’s interactive components—such as modal dialogs, dropdown menus, and custom tooltips—are designed to work for touch, mouse, and keyboard users.

Through the use of relevant WAI-ARIA roles and attributes, these components should also be understandable and operable using assistive technologies (such as screen readers).

Because Bootstrap’s components are purposely designed to be fairly generic, authors may need to include further ARIA roles and attributes, as well as JavaScript behavior, to more accurately convey the precise nature and functionality of their component. This is usually noted in the documentation.

Color contrast

Some combinations of colors that currently make up Bootstrap’s default palette—used throughout the framework for things such as button variations, alert variations, form validation indicators—may lead to *insufficient* color contrast (below the recommended WCAG 2.1 text color contrast ratio of 4.5:1 and the WCAG 2.1 non-text color contrast ratio of 3:1), particularly when used against a light background. Authors are encouraged to test their specific uses of color and, where necessary, manually modify/extend these default colors to ensure adequate color contrast ratios.

Visually hidden content

Content which should be visually hidden, but remain accessible to assistive technologies such as screen readers, can be styled using the `.visually-hidden` class. This can be useful in situations where additional visual information or cues (such as meaning denoted through the use of color) need to also be conveyed to non-visual users.

```
<p class="text-danger">
  <span class="visually-hidden">Danger: </span>
  This action is not reversible
</p>
```

For visually hidden interactive controls, such as traditional “skip” links, use the `.visually-hidden-focusable` class. This will ensure that the control becomes visible once focused (for sighted keyboard users). **Watch out, compared to the equivalent `.sr-only` and `.sr-only-focusable` classes in past versions, Bootstrap 5’s `.visually-hidden-focusable` is a standalone class, and must not be used in combination with the `.visually-hidden` class.**

```
<a class="visually-hidden-focusable" href="#content">Skip to main content</a>
```

Reduced motion

Bootstrap includes support for the `prefers-reduced-motion` media feature. In browsers/environments that allow the user to specify their preference for reduced motion, most CSS transition effects in Bootstrap (for instance, when a modal dialog is opened or closed, or the sliding animation in carousels) will be disabled, and meaningful animations (such as spinners) will be slowed down.

On browsers that support `prefers-reduced-motion`, and where the user has *not* explicitly signaled that they'd prefer reduced motion (i.e. where `prefers-reduced-motion: no-preference`), Bootstrap enables smooth scrolling using the `scroll-behavior` property.

Additional resources

- [Web Content Accessibility Guidelines \(WCAG\) 2.1](#)
- [The A11Y Project](#)
- [MDN accessibility documentation](#)
- [Tenon.io Accessibility Checker](#)
- [Color Contrast Analyser \(CCA\)](#)
- [“HTML Codesniffer” bookmarklet for identifying accessibility issues](#)
- [Microsoft Accessibility Insights](#)
- [Deque Axe testing tools](#)
- [Introduction to Web Accessibility](#)