

> Getting started

> Customize

Overview

Sass

Options

Color

Components

CSS variables

Optimize

> Layout

> Content

> Forms

> Components

> Helpers

Customize

View on GitHub

Learn how to theme, customize, and extend Bootstrap with Sass, a boatload of global options, an expansive color system, and more.



API Design, API Management, API Security & Access Management, & API Productization. Plus some more.
ads via Carbon

Sass

Utilize our source Sass files to take advantage of variables, maps, mixins, and functions.

Color

Learn about and customize the color systems that support the entire toolkit.

CSS variables

Use Bootstrap's CSS custom properties for fast and forward-looking design and development.

Options

Customize Bootstrap with built-in variables to easily toggle global CSS preferences.

Components

Learn how we build nearly all our components responsively and with base and modifier classes.

Optimize

Keep your projects lean, responsive, and maintainable so you can deliver the best experience.

Overview

There are multiple ways to customize Bootstrap. Your best path can depend on your project, the complexity of your build tools, the version of Bootstrap you're using, browser support, and more.

Our two preferred methods are:

1. Using Bootstrap [via package manager](#) so you can use and extend our source files.
2. Using Bootstrap's compiled distribution files or [jsDelivr](#) so you can add onto or override Bootstrap's styles.

While we cannot go into details here on how to use every package manager, we can give some guidance on [using Bootstrap with your own Sass compiler](#).

For those who want to use the distribution files, review the [getting started page](#) for how to include those files and an example HTML page. From there, consult the docs for the layout, components, and behaviors you'd like to use.

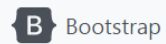
As you familiarize yourself with Bootstrap, continue exploring this section for more details on how to utilize our global options, making use of and changing our color system, how we build our components, how to use our growing list of CSS custom properties, and how to optimize your code when building with Bootstrap.

CSPs and embedded SVGs

Several Bootstrap components include embedded SVGs in our CSS to style components consistently and easily across browsers and devices. **For organizations with more strict CSP configurations**, we've documented all instances of our embedded SVGs (all of which are applied via [background-image](#)) so you can more thoroughly review your options.

- [Accordion](#)
- [Close button](#) (used in alerts and modals)
- [Form checkboxes and radio buttons](#)
- [Form switches](#)
- [Form validation icons](#)
- [Select menus](#)
- [Carousel controls](#)
- [Navbar toggle buttons](#)

Based on [community conversation](#), some options for addressing this in your own codebase include replacing the URLs with locally hosted assets, removing the images and using inline images (not possible in all components), and modifying your CSP. Our recommendation is to carefully review your own security policies and decide on the best path forward, if necessary.



Designed and built with all the love in the world by the [Bootstrap team](#) with the help of our contributors.

Code licensed MIT, docs CC BY 3.0.

Currently v5.0.2.

Links

[Home](#)

[Docs](#)

[Examples](#)

[Themes](#)

[Blog](#)

Guides

[Getting started](#)

[Starter template](#)

[Webpack](#)

[Parcel](#)

Projects

[Bootstrap 5](#)

[Bootstrap 4](#)

[Icons](#)

[RFS](#)

[npm starter](#)

Community

[Issues](#)

[Discussions](#)

[Corporate sponsors](#)

[Open Collective](#)

[Slack](#)

[Stack Overflow](#)