

How to customize Bootstrap with Sass

Bootstrap is a fantastic CSS framework that can help you create stylish and polished websites. Some programmers and development teams find that writing code in Bootstrap is easier to edit than regular CSS.

Bootstrap is a fantastic CSS framework that can help you create stylish and polished websites. Some programmers and development teams find that **writing code in Bootstrap** is easier to edit than regular CSS.

However, if you use **Bootstrap** with its default styles, every website will look the same. The internet would therefore become bland. Fortunately, Bootstrap is also highly customizable.

How to customize Bootstrap

If you're new to Bootstrap, you can customize it with a custom CSS stylesheet. CSS specificity is crucial here. You write your custom CSS with the same or higher specificity and link to it in the header `index.html` after this line links to the original Bootstrap CSS.

This is fine for minor tweaks, but for larger projects, it can be time-consuming and redundant with declarations. In fact, there's a more streamlined way.

How to use Sass with Bootstrap

The solution here is to use Sass – a CSS preprocessing program. Sass compiles CSS before it is used on web pages.

Since version 4, Bootstrap has only used Sass. This source code for Bootstrap frameworks 4 and 5 is written entirely in Sass.

Sass comes with two syntaxes. The older one uses indentation, and the newer SCSS syntax (SCSS for Sassy CSS) uses curly braces just like CSS.

SCSS is a superset of CSS. Therefore, CSS code saved with the `.scss` extension (or SCSS interspersed with CSS) is also valid Sass code.

This guide uses the SCSS version. Regardless of the style used, the Sass compiler can convert it into vanilla CSS for browser use.

The two things we will cover in this guide include:

1. Change the primary and secondary background colors.
2. Change the default media breakpoint that Bootstrap uses.

Once you can do the above, making other customizations will be easier.

Required conditions:

1. Node.js with npm or yarn
2. A code editor, preferably VS Code.
3. Basic understanding of Sass

After installing Node.js, we will use the npm version:

```
npm i bootstrap
```

Of course, you'll also need to install the Sass compiler if you don't already have it. Just download the `dart-sass` package, extract it, and add it to the path (environment variables). You'll have a package `npm sass` and the Live Sass Compiler VS Code extension with over 2 million downloads. Feel free to use any compiler you like.

This example uses `npm package: sass`.

After loading Bootstrap, the Sass compiler, in the `node-modules` directory, has an entry named `bootstrap`, inside which are three folders: `dist`, `js` and `scss`. All CSS is compiled in `dist`. Bootstrap JavaScript code is located in `js`. All Sass files are located in the `scss` directory.

How to change the primary and secondary background colors

The idea here is to overwrite the `.scss` file and recompile it. According to Bootstrap's official guidelines, you should limit changes to the original Bootstrap file. Therefore, a `custom.scss` stylesheet will be created here.

We will import all Bootstrap in the file `custom.css`:

```
//custom.scss @import "./node_modules/bootstrap/scss/bootstrap";
```

Variables have suffixes `!default` (a Sass flag) in Bootstrap. The flag `!default` indicates that the compiler is set to only set the value if it is not already defined.

Therefore, we will set the variables before the entry `@import` so that later, the compiler will choose the provided value instead of the default.

```
//custom.scss $primary: teal; $secondary: green; @import "./node_modules/bootstrap/scss/bootstrap";
```

You will also need an HTML file to preview the results.

```
Welcome! Customize Bootstrap with Sass
```

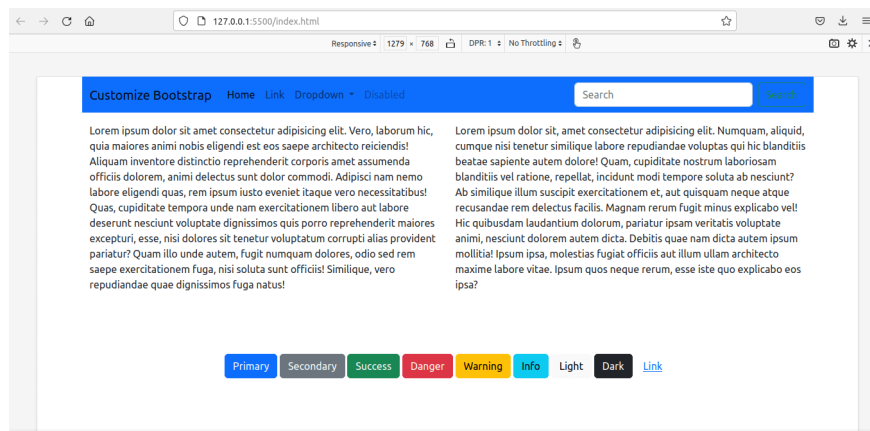
Customize Bootstrap ▣

1. Gate
 2. Link
 3. Dropdown
 1. Action
 2. Another action
 3. Something else here
 4. Disabled
-

Search

Lorem ipsum dolor sit amet consectetur adipisicing elit. Vero, laborum hic, qui

Sass hasn't been compiled yet. To see the default styles, run Live Server. If you haven't installed it, you can download this extension for free from the VS Code extension list. The result will be as follows:



Now it's time to compile the custom Sass file. The compilation formula is simple: Specify the source and destination directories separated by colons.

We have the file `custom.scss` in the folder with the same name:

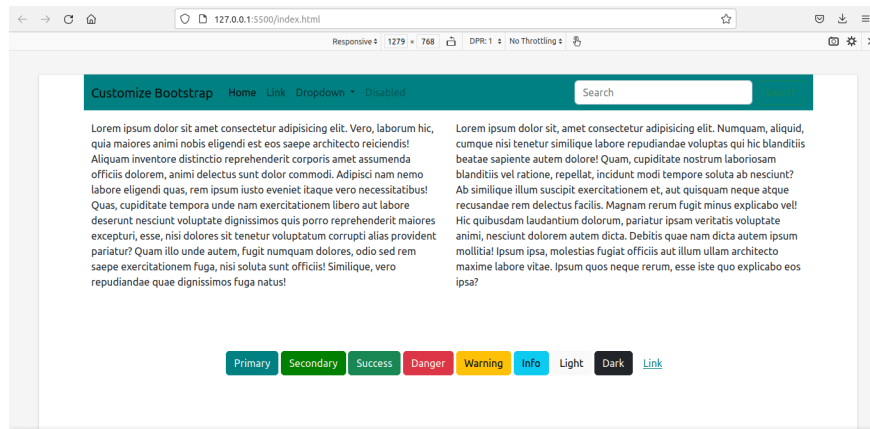
```
sass custom_scss/custom.scss:assets/css/custom_bootstrap.css
```

After recompiling, we have the customized Bootstrap in the `file assets/css/custom_bootstrap.css`.

Instead of the default Bootstrap file, we'll use this custom Bootstrap stylesheet:

Restart the Live Server.

We will have a customized website with new styles.



How to change grid breakpoints in Bootstrap 5

We will now customize the media breakpoints and redefine the container-max-widths.

The easiest way is to overwrite the variables:

```
$primary: teal; $secondary: green; $grid-breakpoints: ( xs: 0, sm: 576px, md: 768px);
```

Using functions `map-merge()` would be better in this case. First, we need to import the functions into the file beforehand `custom.scss` to make them `map.merge()` readily available.

We also need to input the variables `$grid-breakpoints` defined there.

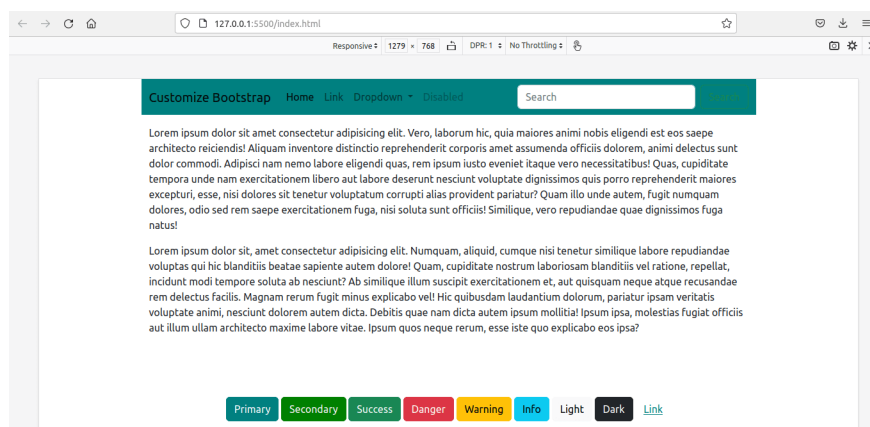
```
//custom.scss $primary: teal; $secondary: green; @import './node_modules/bootstrap/scss/_functions';
```

Here is the code:

```
//custom.scss $primary: teal; $secondary: green; //We have to import the functions
```

Recompile and use the latest file instead of the old one:

This is the final result:



Above is **how to customize Bootstrap with Sass** . I hope this article is helpful to you.

You finished reading the article "**How to customize Bootstrap with Sass**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.
