

# How to use Sass in React

Sass is an improved version of CSS that you can use in React projects right now. Here is a tutorial on how to use Sass in React.



Sass (syntactically awesome style sheets) is a CSS extension that contains additional features that make it even more powerful. The best thing about Sass is its compatibility with CSS. That means you can use Sass in web development projects with JavaScript frameworks like React.

However, unlike vanilla CSS, you need a little setup to use Sass. Let's learn how it works by setting up a simple React.js project and integrating Sass inside.

## How to use Sass in a React.js . project

Like other CSS handlers, Sass is not supported by React. To use Sass in React, you need to install third-party dependencies through a package manager like yarn or npm.

You can check if npm or yarn is installed on your local machine by running **npm --version** or **yarn --version** . If you don't see the version number in the terminal, install npm or yarn first.

### Create a React.js project

To follow this tutorial, you can set up a simple React.js app using the **create-react-app** app .

First, use this command to navigate to the directory where you want to create your React project. Then run **npm create-react-app** . This process may take a while. Once it's done, enter the application directory with **cd** . Add

the following to the **App.js** file to get started:

```
import React from "react"; import "./App.css"; function App() { return (
```

## Using Sass in React

Blue Button Red Button Green Button

```
); } export default App;
```

After setting up a basic React project, it's time to integrate Sass.

### Install Sass

You can install Sass via npm or yarn. Install it via **yarn** by running **yarn add sass** or, if you prefer npm, by running **npm install sass**. Your package manager will add the latest version of Sass to the dependencies list in your project's **package.json file**.

### Rename the .css file to .scss or .sass

In the project folder, rename App.css and index.css to App.scss and index.scss respectively.

After renaming these files, you need to update the imports in the App.js and index.js files to match the file extensions as follows:

```
import "./index.scss"; import "./App.scss";
```

From this point on, you should use the .scss extension for any file types you create.

### Import and use variables and combinations

One of the biggest advantages of Sass is that it helps to write clean, reusable code with variables and mixins. While it's not clear how you can do the same thing in React, it's not too much of a difference with using Sass in a project written in plain JavaScript & HTML.

First, create a new **Styles** folder in the src folder. In the Styles folder, create two files: **\_variables.scss** and **\_mixins.scss**. Add the following rule to **\_variables.scss**:

```
$background-color: #f06292; $text-color: #f1d3b3; $btn-width: 120px; $btn-height
```

Add the following code to **\_mixins.scss**:

```
@mixin vertical-list { display: flex; align-items: center; flex-direction: column
```

Then import the variables and mixins in **App.scss** as follows:

```
@import "./Styles/variables"; @import "./Styles/mixins";
```

Use variables and mixins in the App.scss file:

```
@import "../Styles/variables.scss"; @import "../Styles/mixins"; .wrapper { background
```

Here's how to use variables and mixins in React. Besides mixins and variables, you can also use all the other great features in Sass, like functions. There is no limit as long as you set up scss in your React project.

You can now use Sass in React by installing the sass package via npm or yarn following the instructions above. Hope the article is useful to you.

You finished reading the article "**How to use Sass in React**" 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.