

Sass

Next.js has built-in support for Sass using both the `.scss` and `.sass` extensions. You can use component-level Sass via CSS Modules and the `.module.scss` or `.module.sass` extension.

First, install [sass](#) ↗:

>_ Terminal



```
npm install --save-dev sass
```

Good to know:

Sass supports [two different syntax](#) ↗, each with their own extension. The `.scss` extension requires you use the [SCSS syntax](#) ↗, while the `.sass` extension requires you use the [Indented Syntax \("Sass"\)](#) ↗.

If you're not sure which to choose, start with the `.scss` extension which is a superset of CSS, and doesn't require you learn the Indented Syntax ("Sass").

Customizing Sass Options

If you want to configure the Sass compiler, use `sassOptions` in `next.config.js`.

 next.config.js




```
1  const path = require('path');
2
3  module.exports = {
4    sassOptions: {
5      includePaths: [path.join(__dirname, 'styles')],
6    },
7  };
```

Sass Variables

Next.js supports Sass variables exported from CSS Module files.

For example, using the exported `primaryColor` Sass variable:

 app/variables.module.scss 

```
1 $primary-color: #64ff00;
2
3 :export {
4   primaryColor: $primary-color;
5 }
```

 pages/_app.js 

```
1 import variables from '../styles/variables.module.scss';
2
3 export default function MyApp({ Component, pageProps }) {
4   return (
5     <Layout color={variables.primaryColor}>
6       <Component {...pageProps} />
7     </Layout>
8   );
9 }
```