

# SASS and PostCSS in React

Last updated by | Subramanya Dixit | May 5, 2025 at 9:08 PM GMT+5:30

## Introduction

As projects grow in size and complexity, maintaining scalable and consistent styles becomes difficult. **SASS** (a CSS preprocessor) and **PostCSS** (a CSS transformer) help improve your workflow by adding functionality that regular CSS lacks, like variables, mixins, nesting, and automated vendor prefixing.

## What is SASS (SCSS)?

**SASS** is a preprocessor scripting language that is interpreted or compiled into CSS. **SCSS** is the most popular syntax of SASS (uses curly braces like CSS).

## Key Features

- Variables ( `$primary-color` )
- Nesting
- Mixins
- Partial and imports

## How to Use SASS in React

> React apps (created with Create React App) support SCSS out of the box.

**\*\*Installation (if not using CRA):\*\***

```
npm install sass
```

## styles.scss

```
$primary-color: #007bff;

.button {
  background-color: $primary-color;
  color: white;

  &:hover {
    background-color: darken($primary-color, 10%);
  }
}
```

## App.jsx

```
import './styles.scss';

function App() {
  return <button className="button">Styled Button</button>;
}
```

## What is PostCSS?

**PostCSS** is a tool that transforms CSS with JavaScript plugins.

## ✓ Popular PostCSS Plugins

- `autoprefixer` : Adds vendor prefixes automatically.
- `postcss-nested` : Enables nesting like SASS.
- `postcss-preset-env` : Allows future CSS features.

## 🔧 How to Use PostCSS

PostCSS is already used behind the scenes in tools like Create React App and TailwindCSS.

### Manual setup (e.g. with Webpack or Vite):

#### Install:

```
npm install postcss autoprefixer postcss-preset-env
```



### postcss.config.js

```
module.exports = {  
  plugins: [  
    require('autoprefixer'),  
    require('postcss-preset-env')({ stage: 1 })  
  ]  
};
```



## 📊 SASS vs PostCSS

Feature	SASS/SCSS	PostCSS
Syntax Enhancements	✓ Yes	✗ Depends on plugins
Nesting	✓ Yes	✓ With plugin
Variables	✓ Yes	✓ With plugin
Vendor Prefixing	✗ No	✓ With autoprefixer
Supported in CRA	✓ Yes	✓ Yes
Ecosystem	Mature	Plugin-driven

## 📄 Practice Exercises

1. Convert a `.css` file to `.scss` and refactor using variables and nesting.
2. Configure PostCSS with `autoprefixer` and test on a flexbox layout.

3. Use both SASS and PostCSS in a small component.

---

## ? Quiz Questions

---

1. Which of these tools helps you write future CSS today?

a) SCSS

☒ b) PostCSS

c) Inline Styles

d) Emotion

---

2. What does `autoprefixer` do in PostCSS?

a) Adds SASS syntax

☒ b) Adds vendor prefixes automatically

c) Minifies CSS

d) Replaces SCSS

---

3. Which one supports mixins and variables natively?

☒ a) SASS

b) PostCSS

c) Inline styles

d) CSS Modules

---

4. What's a limitation of using SASS alone?

a) No nesting

☒ b) No vendor prefixing

c) Requires Babel

d) Doesn't support color variables

---