

CSS-in-JS refers to an approach where styles are written in JavaScript instead of in external CSS files to easily scope styles in components, eliminate dead code, encourage faster performance and dynamic styling, and more.

CSS-in-JS bridges the gap between CSS and JavaScript:

1. **Components:** you'll style your site with components, which integrates well with React's "everything is a component" philosophy.
2. **Scoped:** this is a side effect of the first. Just like [CSS Modules](#), CSS-in-JS is scoped to components by default.
3. **Dynamic:** style your site dynamically based on component state by integrating JavaScript variables.
4. **Bonuses:** many CSS-in-JS libraries generate unique class names which can help with caching, automatic vendor prefixes, timely loading of critical CSS, and implementing many other features, depending on the library you choose.

CSS-in-JS, while not required in Gatsby, is very popular among JavaScript developers for the reasons listed above. For more context, read Max Stoiber's (creator of CSS-in-JS library [styled-components](#)) article [Why I write CSS in JavaScript](#). However, you should also consider whether CSS-in-JS is necessary, as not relying on it can encourage more inclusive frontend skill-sets. It is also more difficult to port styles from JSX to and from CSS.

Note that this functionality is not a part of React or Gatsby, and requires using any of the many [third-party CSS-in-JS libraries](#).

Adding a stable CSS class to your JSX markup along with your CSS-in-JS can make it easier for users to include [User Stylesheets](#) for accessibility. See [Styled Components](#) example.

Keep in mind that styles aren't applied until the JavaScript loads hence a plugin to extract the styles is necessary to prevent flash of unstyled content (FOUC). To cater for this, every CSS-in-JS library has a Gatsby plugin which you need to extract styles and insert them into the HTML during builds and this prevents FOUC.

This section contains guides for styling your site with some of the most popular CSS-in-JS libraries, including how to set up global styles using each library.