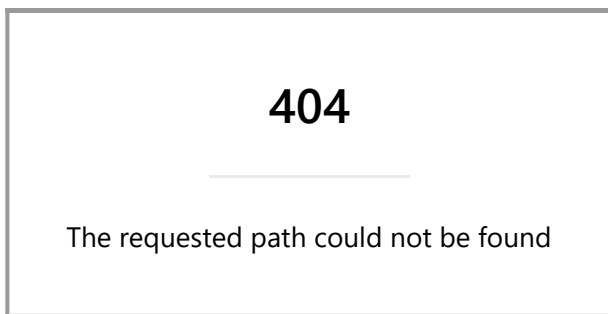CSS is a core part of any web app. By extension, a UI framework that doesn't have a built-in way to add styles to your components is unfinished.

That's why Svelte allows you to add CSS in a component's `<style>` tag. Co-locating your CSS with your markup means we can [solve the biggest problems developers face when writing CSS](#) without introducing new ones, all while providing a rather nice development experience.

But Svelte's style handling does have some limitations. It's too difficult to share styles between components, or apply app-level optimisations. These are areas we plan to address in future versions, but in the meantime if you need those things you can use any framework-agnostic CSS-in-JS library.

## For example

Here, we're using [Emotion](#) to generate scoped class names that can be used across multiple components:

# 404

---

The requested path could not be found

It's important to note that most CSS-in-JS libraries have a runtime library, and many don't support statically extracting styles out into a separate `.css` file at build time (which is essential for the best performance). You should therefore only use CSS-in-JS if it's necessary for your application!

Note that you can mix-and-match — you can still use Svelte's built-in CSS handling alongside a CSS-in-JS library.