Component-Scoped Styles with CSS Modules

Component-scoped CSS allows you to write traditional, portable CSS with minimal side effects: gone are the worries of selector name collisions or affecting other components' styles.

Gatsby works out of the box with CSS Modules, a popular solution for writing component-scoped CSS. Here is an example site that uses CSS Modules.

What is a CSS Module?

Quoting from the CSS Module homepage:

> A **CSS Module** is a CSS file in which all class names and animation names are scoped locally by default.

CSS Modules let you write styles in CSS files but consume them as JavaScript objects for additional processing and safety. CSS Modules are very popular because they automatically make class and animation names unique so you don't have to worry about selector name collisions.

CSS Module example

The CSS in a CSS module is no different than normal CSS, but the extension of the file is different to mark that the file will be processed.

```css
.container {
  margin: 3rem auto;
  max-width: 600px;
}
```

```jsx
import React from "react"
// highlight-next-line
import * as containerStyles from "./container.module.css"

export default function Container({ children }) {
  return (
    // highlight-next-line
    <section className={containerStyles.container}>{children}</section>
  )
}
```

In this example, a CSS module is imported and declared as a JavaScript object called `containerStyles`. Then, a CSS class from that object is referenced in the JSX `className` attribute with `containerStyles.container`, which renders into HTML with dynamic CSS class names like `container-module--container--3MbgH`.

Enabling user stylesheets with a stable class name

Adding a persistent CSS `className` to your JSX markup along with your CSS Modules code can make it easier for users to take advantage of User Stylesheets for accessibility.

Here's an example where the class name `container` is added to the DOM along with the module's dynamically-created class names:

```
import React from "react"
import * as containerStyles from "./container.module.css"

export default function Container({ children }) {
  return (
    <section className={`container ${containerStyles.container}`}>
      {children}
    </section>
  )
}
```

A site user could then write their own CSS styles matching HTML elements with a class name of `.container`, and it wouldn't be affected if the CSS module name or path changed.

When to use CSS Modules

CSS Modules are highly recommended for those new to building with Gatsby (and React in general) as they allow you to write regular, portable CSS files while gaining performance benefits like only bundling referenced code.

How to build a page using CSS Modules

Visit the CSS Modules section of the tutorial for a guided tour of building a page with CSS Modules.