

CSS Modules are a way to locally scope your CSS by default. This means that styles defined in a CSS Module are limited in scope to the component or file where they are imported. This helps prevent global naming conflicts and makes it easier to maintain and organize your styles in larger applications. Here's a detailed explanation of CSS Modules:

Creating a CSS Module

To create a CSS Module, you typically use a `.module.css` file extension (e.g., `styles.module.css`).

Import your CSS Module in your JavaScript or TypeScript file like this:

```
import styles from './styles.module.css';
```

Local Scope

Styles defined in a CSS Module are scoped to the component or file where the module is imported. This means that class names are unique and specific to that component, reducing the chance of naming conflicts.

Class Naming

In a CSS Module, class names are automatically transformed to be unique and locally scoped. For example, if you define a class name in your CSS Module like this:

```
.myButton {  
  background-color: blue;  
}
```

When imported into a component, you access it like this:

```
<button className={styles.myButton}>Click me</button>
```

The generated class name would be something like `styles_myButton__[unique-id]`.

No Need for BEM or Naming Conventions

CSS Modules eliminate the need for BEM (Block Element Modifier) or other naming conventions because each class name is already unique to the component.

Maintainability

CSS Modules improve maintainability by encapsulating styles with components. Changes to a component's styles won't affect other components.

Avoiding Element Selectors

In CSS Modules, element selectors (e.g., `div`, `p`, `span`) are not typically used because CSS Modules are designed to provide locally scoped styles for components or elements, and element selectors target all instances of a specific HTML element globally. This could lead to unintended and unpredictable styling conflicts.

Instead, CSS Modules encourage you to use class names to style your components or elements.

```
.nav{}  
  
.nav ul  
{  
  display: flex;  
  justify-content: space-between;  
  background-color: aquamarine;  
  list-style: none;  
}
```