

# Styling Components in React

---

In this lecture, we will learn different ways to style components in React. Styling is important to make your application look polished and user-friendly.

## Using Regular CSS Files

---

You can write your styles in a separate CSS file and import it into your component.

App.css :

```
.container {  
  text-align: center;  
  padding: 20px;  
  border: 1px solid gray;  
}
```

App.jsx :

```
import "./App.css"  
  
function App() {  
  return <div className="container">Styled with CSS file</div>  
}  
  
export default App
```

Here, we use `className` instead of `class` because `class` is a reserved word in JavaScript.

## Inline Styles

---

You can also apply styles directly to elements using inline style objects.

```
function App() {
  const style = {
    color: "blue",
    fontSize: "20px",
    padding: "10px"
  }

  return <h2 style={style}>Styled with Inline Styles</h2>
}

export default App
```

Notice that CSS properties are written in **camelCase** (e.g., `fontSize` instead of `font-size` ).

## CSS Modules

---

For component-specific styles, you can use CSS Modules. This prevents class name conflicts.

`Button.module.css` :

```
.button {
  background-color: green;
  color: white;
  padding: 10px;
  border: none;
  border-radius: 5px;
}
```

`Button.jsx` :

```
import styles from "./Button.module.css"

function Button() {
  return <button className={styles.button}>Click Me</button>
}
```

```
export default Button
```

Here, `styles.button` applies the scoped CSS class to the button.

## Recap

---

- You can style React components with regular CSS files, inline styles, or CSS Modules.
- `className` is used instead of `class`.
- CSS Modules are useful to avoid class name conflicts.