CSS Styling React App

- In Reacts, there are several ways to apply styling to components.
- Here are some common approaches:
  1. **Inline Styles**
  2. **CSS Modules**
  3. **CSS-in-JS Libraries**
  4. **External CSS Files**

1. Inline Styles

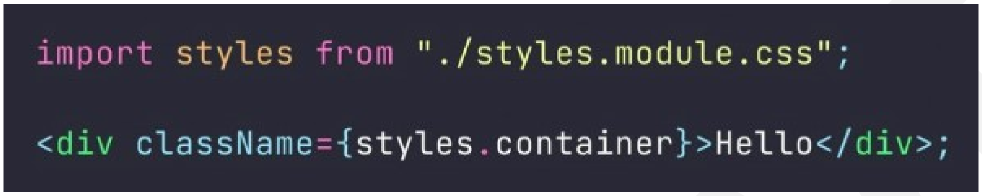
- You can apply inline styles directly to JSX elements using the style attribute.
- The style attribute takes a JavaScript object with camel-cased CSS property-value pairs.

```
<h1 style={{backgroundColor:'yellow'}}>Creative</h1>

const styles={
     color:'blue'
};
<h1 style={styles}>Creative Design</h1>
```

2. CSS Modules

- CSS Modules allow you to write **CSS stylesheets** and **import** them as JavaScript modules.
- CSS class names are scoped locally to the component, preventing style conflicts. You can **import** the CSS file and use the defined class names in your JSX code

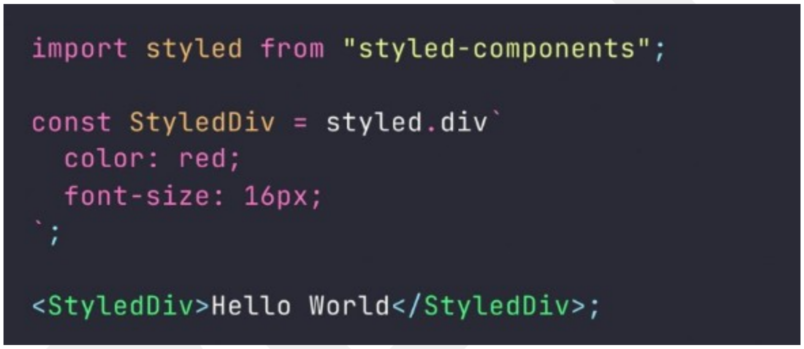


```
import styles from "./styles.module.css";

<div className={styles.container}>Hello</div>;
```

3. CSS-in-JS Libraries

- CSS-in-JS libraries are a popular approach for styling in React.js that allows you to write CSS directly in your JavaScript code. such as styled-components, emotion, etc..



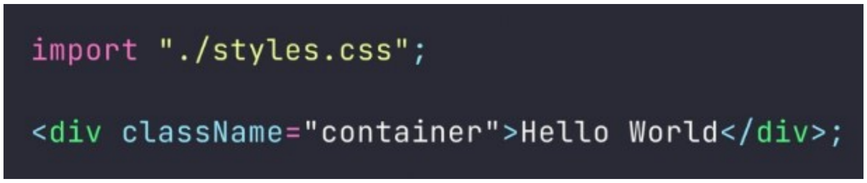
```
import styled from "styled-components";

const StyledDiv = styled.div`
  color: red;
  font-size: 16px;
`;

<StyledDiv>Hello World</StyledDiv>;
```

4. External CSS Files

- You can also **use traditional** external CSS files and import them into your components.
- In this case, you **import the CSS file** and add the corresponding class names to your JSX elements. The styles defined in the external CSS file will be applied **globally**.



```
import "./styles.css";

<div className="container">Hello World</div>;
```