

Styling React components

Styling React components can be done in a variety of ways, but there are a few popular approaches:

1. **Inline styles:** You can use inline styles directly in your JSX components by passing a JavaScript object as the style prop. For example:

```
<div style={{ backgroundColor: 'red', color: 'white' }}>Hello world</div>
```

2. **CSS modules:** You can use CSS modules to define styles in a separate CSS file and import them into your React components. This approach allows you to write standard CSS rules and use them in your components just like regular class names. For example:

```
import styles from './styles.module.css';
function MyComponent() {
  return (
    <div className={styles.container}>
      <h1 className={styles.title}>Hello world</h1>
    </div>
  );
}
```

3. **CSS-in-JS:** CSS-in-JS libraries like styled-components, Emotion, and JSS allow you to write CSS code directly in your JavaScript files. This approach provides greater control over the styling of your components and enables you to reuse styles across your application. For example:

```
import styled from 'styled-components';
const Container = styled.div`
  background-color: red;
  color: white;
`;
function MyComponent() {
  return (
    <Container>
      <h1>Hello world</h1>
    </Container>
  );
}
```

```
</Container>  
);  
}
```

4. **CSS Frameworks:** You can also use CSS frameworks like Bootstrap, Material-UI, or Tailwind CSS to style your React components. These frameworks provide pre-built components and styles that you can use to quickly and easily create your UI. For example:

```
import Button from 'react-bootstrap/Button';  
  
function MyComponent() {  
  return <Button variant="primary">Click me</Button>;  
}
```