

CSS No React

Tem diversas formas de inserir CSS no React, a mais comum é colocar um arquivo CSS chamado Pagina.module.css junto ao componente jsx:

Pagina.jsx

Pagina.module.css

Styled Components

É uma biblioteca para estilização de componentes no react que mistura JavaScript com CSS.

Instalação: npm install styled-components

Importação: import styled from "styled-components"

Todos os componentes de estilo criados terão a primeira letra maiúscula.

Para ajudar na formatação do styled components é preciso baixar a extensão do vscode: vscode-styled-components.

Utilização Básica: Para utilizar é preciso criar uma variável com o elemento HTML que irá ser estilizado. Pode ser dentro da função de componente e fora do return ou fora da função de componente:

```
const Btn = styled.button `
```

Dentro das crases ficará a estilização em CSS:

```
const Btn = styled.button `
```

```
  color: white;
```

```
  background-color: black;
```

```
`
```

Para funcionar, a variavel vai substituir o elemento que estava no lugar:

```
<button>Enviar</button> → <Btn>Enviar</Btn>
```

Utilização Certa: Da forma anterior fica bagunçado, por isso sempre é utilizado em um arquivo separado:

styles.jsx

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

```
const Btn = styled.button `
  color: white;
  background-color: black;
`
```

```
export default Btn
```

App.jsx

```
import { Btn } from "../styles.jsx"
```

```
<Btn>Enviar</Btn>
```

CSS + JS: É possível passar valores de variáveis do componente para o arquivo de estilo através de atributos:

```
<Btn isOn={variavelDinamica}>Enviar</Btn>
```

styles.jsx

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

```
const Btn = styled.button `
  color: white;
```

```
background-color: ${props => props.isOn ? 'red' : 'gray'};
```

```
export default Btn
```

Dentro do props vai chegar todos os atributos que tem dentro do componente.

Estilização de filhos: Para estilizar filhos de componentes CSS, ou seja, elementos html que estão dentro de um componente CSS, é possível fazer normalmente criando outro componente ou no styles colocar o elemento filho como um seletor normal do CSS:

```
export const imgWrapper = styled.div `
```

```
width: 100px;
```

```
height: 100px;
```

```
img {
```

```
width: 100%;
```

```
height: 100%;
```

```
}
```

É possível também estilizar outros componentes, ao invés de ser um seletor será o nome do componente entre cifrão e chaves:

```
const Pai = styled.div `
```

```
background-color: white;
```

```
${Btn} {
```

```
color: black;
```

```
}
```

Estilos Globais: Para criar arquivos globais é preciso de um arquivo novo com o `createGlobalStyle`, exportar e importar no `index.html` em baixo do App em forma de componente.

globalStyle.jsx

```
import { createGlobalStyle } from "styled-components"
```

```
const globalStyle = createGlobalStyle `
  margin: 0;
  padding: 0;
```

```
`
```

```
export default globalStyle
```

index.html

```
<App />
```

```
<GlobalStyle />
```

Exemplo

styles.jsx

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

```
export const Container = styled.div `
  width: 40px;
  height: 100px;
  background-color: blue;
```

```
  img {
    width: 100%;
  }
```

```
`
```

```
export const Btn = styled.button `
  border: 1px solid black;
```

```
color: white;
background-color: ${props => props.validation ? 'black' : 'white'};
`
```

App.jsx

```
import { Container, Btn } from "../styles"
<Container>
  
  <Btn validation={variavelDinamica}>Enviar</Btn>
</Container>
```

ThemeProvider

```
import { ThemeProvider } from 'styled-components';
```

```
const theme = {
  primary: '#6200ee',
  secondary: '#03dac6',
};
```

```
const Button = styled.button`
  background: ${props => props.theme.primary};
  color: white;
`;
```

```
<ThemeProvider theme={theme}>
  <Button>Themed Button</Button>
</ThemeProvider>
```

Para deixar mais global, é possível colocar o ThemeProvider no main.jsx ou App.jsx pra ele englobar a aplicação inteira, o theme ficaria no main ou App do lado de fora.

CSS + JS 2

```
const fontSize = 30;

const Title = styled.h3`
  font-size: ${fontSize}px;
  a {
    color: orange;
    text-decoration: none;
    &:hover {
      color: #666;
    }
  }
`;

function PostCard() {
```