

Introdução ao **CSS-in-JS**

O que é, como funciona, quais vantagens e mais.

//

CSS é muito difícil! É sério!

Quem diz o contrário, é porque quer ser legal.



Problemas do CSS

- Falta de escopo local (uma lib de terceiro pode colidir com meu código)
- Especificidade e novamente, colisão de estilos!
- Código não utilizado (dead code)
- Falta de modularidade
- Dificuldade na manutenção e código quebrando sabe-se lá Deus onde

// Os pré-processadores como **Less, Sass, Stylus** vieram ajudar em algumas coisas, mas continuaram com os problemas clássicos, além de abrir margem para **péssimas práticas**.

```
nav {  
  ul {  
    li {  
      a {  
  
      }  
    }  
  }  
}
```



```
nav ul li a { ... }
```

CSS-in-JS

// O **CSS-in-JS** se refere a um conjunto de ideias para resolver os complexos problemas do **CSS**. E existem diferentes bibliotecas que fazem essa prática.

Bibliotecas de **CSS-in-JS**

- Aphrodite
- Emotion
- Glamor
- Styled Components
- Styled JSX
- Muitas outras (<https://github.com/michelebertoli/css-in-js>)

Styled Components

Vantagens do **Styled Components**

- Critical CSS automático
- Escopo definido (sem colisão de classes)
- Remoção de CSS não utilizado
- Estilos dinâmicos (props, themes)
- Manutenção simplificada e sem dor
- Vendor prefixing automático

Como funciona?

```
import styled from 'styled-components'

export const Main = styled.div`
  align-items: center;
  display: flex;
`
```



```
.styled__Main-sc-11b8j8d-1-bSsuBw {
  -webkit-box-align: center;
  -webkit-box-pack: justify;
  align-items: center;
  display: flex;
  justify-content: space-between;
}
```

Estilo dinâmico

```
const Button = styled.button`
  background-color: ${props => props.primary ? 'palevioletred' : 'white'};
  color: ${props => props.primary ? 'white' : 'palevioletred'};
`

render(
  <div>
    <Button>Normal</Button>
    <Button primary>Primary</Button>
  </div>
)
```

Estender estilos

```
const Button = styled.button`
  color: palevioletred;
  font-size: 1em;
  margin: 1em;
  padding: 0.25em 1em;
  border: 2px solid palevioletred;
  border-radius: 3px;
`

const TomatoButton = styled(Button)`
  color: tomato;
  border-color: tomato;
`

render(
  <div>
    <Button>Normal Button</Button>
    <TomatoButton>Tomato Button</TomatoButton>
  </div>
)
```

E muito mais...