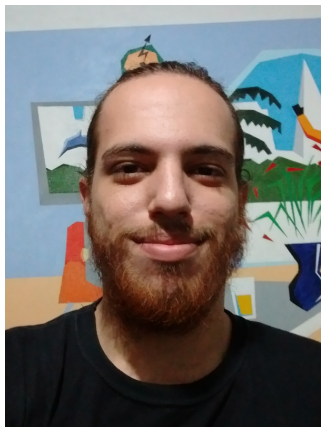
Estilizando componentes com STYLED-COMPONENTS

> whoami_



Juliano Rafael

Lead Front End Developer na Getty/IO
FreeCodeCamp contributor
Usuário UNIX desde 2014
E sim, eu escrevo JS sem `;`

Tópicos

1. Porque CSS-in-JS?
2. Porque styled-components?
3. Como começar a usar styled-components?
4. O que posso fazer com styled-components?

PORQUE CSS-IN-JS?

```
import React from 'react'
import './example_1.css'

const TodoItem = ({ name, due, complete, small }) => (
  <div className={
    `todo-item ${small ? 'todo-item--small' : ''} ${due ? 'is-due' : ''} ${complete ? ' is-complete' : ''}`
  }>
    <span className="todo-item__mark" />
    <span className="todo-item__name">{name}</span>
  </div>
)

export default TodoItem
```

```
.todo-item {
  color: #000
}
.todo-item--small {
  font-size: 80%
}
.todo-item__mark::before {
  content: '*'
}
.todo-item.is-complete {
  color: #aaa
}
.todo-item.is-complete .todo-item__mark::before {
  content: '✔'
}
.todo-item.is-complete .todo-item__name {
  text-decoration: line-through
}
.todo-item.is-due .todo-item__name {
  color: red
}
```

Que resulta em:

`<TodoItem name="item" />`	*item
`<TodoItem name="item" complete />`	✔~~item~~
`<TodoItem name="item" due />`	*item
`<TodoItem name="item" small />`	*item

Pain points

```
import './example_1.css'
```

```
<div className={
    `todo-item ${small ? 'todo-item--small' : ''} ${due ? 'is-due' : ''} ${
complete ? ' is-complete' : ''}`
  }>
```

O componente fica separado em duas partes.

O componente com React e a folha de estilos. Um não funciona adequadamente sem o outro.

Os estilos ficam entrelaçados com o componente tornando a reusabilidade do CSS difícil.

Se não vamos reutilizar o CSS, porque criar classes?

React Native

If you're writing React, you have access to a more powerful styling construct than CSS class names. You have components.

- Michael Chan (learnreact.com)

PORQUE
STYLED-COMPONENTS?

Remove a necessidade de classes.

Nada conflitos de nome, regras de nomenclatura, etc.

Não requer praticamente nenhum setup.

Não requer nenhuma sintaxe especial.

Transforma seu CSS em componentes.

Novamente, transforma seu CSS em componentes.

COMO COMEÇAR A USAR STYLED-COMPONENTS?

< 💅 >

```
npm install --save styled-components
```

```
yarn add styled-components
```

```
import styled from 'styled-components'

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

  cursor: pointer;
  font-size: 1em;
  margin: 1em;
  padding: 0.25em 1em;
  border: 2px solid palevioletred;
  border-radius: 3px;

  &:hover {
    background: ${props => props.primary ? 'white' : 'palevioletred'};
    color: ${props => props.primary ? 'palevioletred' : 'white'};
  }
`;

export default Button
```

Botão

Botão primário

`styled.button`/* css */`

é uma função que retorna um componente de React de vai renderizar um botão na DOM.

O uso da crase no javascript ES6 equivale a invocar a função passando uma string como parâmetro.

E no React Native?

```
import React from 'react'
import styled from 'styled-components'

const Button = styled.TouchableOpacity`
  background: ${props => (props.primary ? 'palevioletred' : 'white')};

  margin: 1em;
  padding: 0.25em 1em;
  border: 2px solid palevioletred;
  border-radius: 3px;
`

const Text = styled.Text`
  font-size: 1em;
  color: ${props => props.primary ? 'white' : 'palevioletred'};
`

export default ({ children, ...props }) => (
  <Button {...props}>
    <Text primary={props.primary}>{children}</Text>
  </Button>
)
```

Voltando ao to do...

```
import React from 'react'
import styled, { css } from 'styled-components'

const Wrapper = styled.div`
  color: #000;
  ${props => props.complete && css`color: #aaa;`}
  ${props => props.due && css`color: #f00;`}
  ${props => props.small && css`font-size: 80%;`}
`

const Name = styled.span`
  ${props => props.complete && css`text-decoration: line-through;`}
  ${props => props.due && css`color: #f00;`}
`

const TodoItem = ({ name, ...props }) => (
  <Wrapper {...props}>
    <span>{props.complete ? '✔' : '*'}</span>
    <Name {...props}>{name}</Name>
  </Wrapper>
)

export default TodoItem
```

DIFERENÇAS

- Menos código no total (CSS + JS).
- Sem classes.
- Não há necessidade de importar estilos.
- Todos os recursos do CSS estão disponíveis para os estilos.
- Todos os recursos do JS estão disponíveis para os estilos.
- Funciona aproximadamente da mesma maneira no React Native.

O QUE MAIS PODEMOS FAZER COM STYLED-COMPONENTS?

TEMAS

```jsx
import React from 'react'
import { ThemeProvider } from 'styled-components'
import Button from './example_5'

export default () => (
  <div>
    <Button>Default Button</Button>
    <ThemeProvider theme={{ main: 'royalblue' }}>
      <Button>Themed</Button>
    </ThemeProvider>
  </div>
)
```

```
import styled from 'styled-components'

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

  color: ${props => props.theme.main};
  border: 2px solid ${props => props.theme.main};
`

Button.defaultProps = {
  theme: {
    main: 'darkorange'
  }
}

export default Button
```

CONCLUSÃO

STYLED-COMPONENTS

FACILITA MUITO A CRIAÇÃO DE COMPONENTES DE UI REUTILIZÁVEIS EM NOSSOS APPS.

FAZ PARTE DO STANDARD NA GETTY/IO
QUANDO SE TRATA DE APLICAÇÕES COM
REACT E REACT NATIVE.

OBRIGADO!