

# React Hooks

DEVinHouse

Parcerias para desenvolver a sua carreira

**SENAI**

<LAB365>

Hooks são funções que permitem a você “ligar-se” aos recursos de state e ciclo de vida do React a partir de componentes funcionais. Hooks não funcionam dentro de classes — eles permitem que você use React sem classes.

A principal vantagem dos Hooks, é o de poder utilizar componentes funcionais para guardar estado e gerenciar o ciclo de vida do componente. E componentes funcionais nos aproximam mais da natureza do JavaScript, que está relacionada à programação funcional.

Isso elimina a necessidade de utilizar o operador `this`, que é usado para classes e é pouco intuitivo no JavaScript. Outra vantagem é que podemos construir nossos próprios hooks para implementar nas aplicações.

# Exemplo Class Based

```
5 export default class Home extends React.Component {  
6   constructor(props) {  
7     super(props);  
8     this.state = { text: 'Senai' };  
9     this.changeText = this.changeText.bind(this);  
10  }  
11  
12  changeText(event) {  
13    this.setState({  
14      text: event.target.value  
15    });  
16  }  
17  
18  render() {  
19    return (  
20      <div style={{  
21        'display': 'flex', 'flexDirection': 'column', 'max-width': '240px', 'padding': '20px'  
22      }}>  
23        <span style={{'marginBottom': '10px'}}>{this.state.text}</span>  
24        <input value={this.state.text} onChange={this.changeText}></input>  
25      </div>  
26    )  
27  }
```

# Exemplo Componentes Funcionais

```
3  import React, { useState } from 'react';
4
5  export default function Home() {
6    const [text, setText] = useState('Senai')
7
8    return (
9      <div style={{
10        'display': 'flex', 'flexDirection': 'column', 'max-width': '240px', 'padding': '20px'
11      }}>
12        <span style={{ 'marginBottom': '10px' }}>{text}</span>
13        <input value={text} onChange={(e) => setText(e.target.value)}></input>
14      </div>
15    )
16  }
```



# useState



DEVinHouse

Parcerias para desenvolver a sua carreira

**SENAI**

<LAB365>

O ***useState*** nos permite criar estados em um componente criado a partir de uma função, assim como o “*state*” presente em componentes criados a partir de classes. A inicialização do “*useState*” retorna uma variável e um método para atualizá-la.

```
const [text, setText] = useState('Senai')
```

# Declarando useState

```
const [text, setText] = useState('Senai')
```

Variável

Método para atualizar  
a variável

Valor inicial de "text"



# Exemplo

```
1  import '../App.css';
2  import React, { useState } from 'react';
3
4  export default function Home() {
5
6      const [count, setCount] = useState(0)
7
8      return (
9          <div style={{
10              'display': 'flex', 'flexDirection': 'column', 'max-width': '240px', 'padding': '20px'
11          }}>
12              <span style={{'marginBottom': '10px'}}>Você clicou {count} vezes</span>
13              <button onClick={(e) => setCount(prevValue => prevValue + 1)}>Clique aqui</button>
14          </div>
15      )
16  }
```



# useEffect

DEVinHouse

Parcerias para desenvolver a sua carreira

**SENAI**

<LAB365>

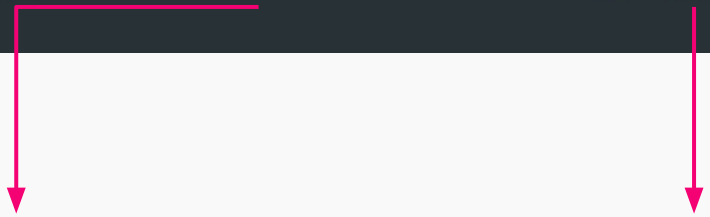
# useEffect

O ***useEffect*** é um Hook que serve para lidar com os efeitos. Podemos usá-los como os lifeCycles `componentDidMount`, `componentDidUpdate` e `componentWillUnmount`.

```
useEffect(() => {  
  console.log('Lorem ipsum.')  
})
```

# Declarando useEffect

```
useEffect(() => { /* Realiza Algo */ }, [])
```



Execução do useEffect

Condição para  
execução

# Aplicação

```
4 export default function Home() {
5   const [categoria, setCategoria] = useState('todos');
6   useEffect(() => {
7     if (categoria === 'todos') {
8       console.log('Buscando todos os dados.')
9       return;
10    }
11    console.log(`Buscar dados da categoria ${categoria}.`)
12  }, [categoria])
13
14  return (
15    <section style={{ 'maxWidth': '250px', 'padding': '20px' }}>
16      <div style={{
17        'display': 'flex', 'flexDirection': 'row', 'justifyContent': 'space-between'
18      }}>
19        <button style={{ 'fontSize': '14px' }} onClick={() => setCategoria('todos')}>Todos</button>
20        <button style={{ 'fontSize': '14px' }} onClick={() => setCategoria('celular')}>Celulares</button>
21        <button style={{ 'fontSize': '14px' }} onClick={() => setCategoria('roupa')}>Roupas</button>
22      </div>
23      <span style={{ 'display': 'block', 'marginTop': '20px' }}>Categoria selecionada: {categoria}</span>
24    </section>
25  )
26 }
```

O que acontecerá com o código a seguir?

```
16  useEffect(( ) => {  
17    |   setCount(prevValue => prevValue+1)  
18  | })  
19  
20  return (  
21    |   <section style={{ 'maxWidth': '250px', 'padding': '20px' }}>  
22    |     <span>{count} vezes</span>  
23    |   </section>  
24  | )
```

- Documentação oficial sobre Hooks - <https://pt-br.reactjs.org/docs/hooks-intro.html>
- Diferenças entre componentes funcionais e componentes de classe  
<https://www.geeksforgeeks.org/differences-between-functional-components-and-class-components-in-react/>
- Site com diversos exemplos de aplicações dos Hooks - <https://usehooks.com/>
- Repo com os códigos das aulas - <https://github.com/yanestevesufjf/clamed-react>

## AVALIAÇÃO DOCENTE

O que você está achando das minhas aulas neste conteúdo?

[Clique aqui](#) ou escaneie o QRCode ao lado para avaliar minha aula.

Sinta-se à vontade para fornecer uma avaliação sempre que achar necessário.







# DEVinHouse

Parcerias para desenvolver a sua carreira

**OBRIGADO!**



<LAB365>

# AGENDA

- Informações do template
- Exemplo com texto (Light mode)
- Exemplo com texto (Dark mode)
- Exemplo com imagens (Light mode)
- Exemplo com imagens (Dark mode)

# INFORMAÇÕES DO TEMPLATE

- Título da Apresentação:
  - Fonte: Ubuntu Bold
  - Formato: Maiúsculo
  - Tamanho: 34
  - Cor: Branco
- Título do Slide:
  - Fonte: Ubuntu Bold
  - Formato: Maiúsculo
  - Tamanho: 22
  - Cor: Branco
- Parágrafos:
  - Fonte: Open Sans Normal
  - Tamanho: 14 a 18
  - Cores: Branco (Dark Mode) ou Preto (Light Mode)
- Marcadores de tópicos:
  - Formatos: Símbolos ou Alfanuméricos
  - Cor: Laranja
- Padrão de Cores:
  - Cinza - #868584
  - Preto - #1C1C19
  - Branco - #FAFAFA
  - Laranja - #F08305
  - Rosa - #c71d81
  - Azul - #0e1d8e

## EXEMPLO COM TEXTO (LIGHT MODE)

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

## EXEMPLO COM TEXTO (DARK MODE)

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

# INTERVALO DE AULA

## DEV!

Finalizamos o nosso primeiro período de hoje. Que tal descansar um pouco?!

Nos vemos em 20 minutos.

**Início:** 20:20

**Retorno:** 20:40



# EXEMPLO COM IMAGENS (LIGHT MODE)

## Título

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.



## Título

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

# EXEMPLO COM IMAGENS (DARK MODE)

## Título

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.



## Título

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.