

Seja PRO: Acesse bootcamps ilimitados e +650 cursos por R\$17,90/mês

1d 0h 13m 27s

MATRICULAR AGORA



? 🔔 5 JV ▾



0



React 101: A biblioteca JavaScript da Meta

JV José Otávio
04/06/2024 19:28

✎ EDITAR ✕ EXCLUIR

O Que é o React? 🤔

React é uma biblioteca JavaScript criada pela Meta (antiga Facebook) em 2013. É usada para construir interfaces de usuário (UI) de maneira eficiente e declarativa. Com React, você pode criar componentes reutilizáveis que se atualizam automaticamente quando os dados mudam.

React é focado na construção de interfaces de usuário de uma forma declarativa, o que significa que você descreve como a UI deve parecer para cada estado do aplicativo. Ele cuida de atualizar o DOM de maneira eficiente.

Componentes no React ✨

Componentes são a peça central do React. Um componente é uma função ou classe JavaScript que opcionalmente aceita entradas, chamadas "props", e retorna um elemento React que descreve como uma seção da UI deve aparecer.

```
// Exemplo de um componente funcional
function BemVindo(props) {
  return <h1>Olá, {props.nome}</h1>;
}

// Usando o componente
<BemVindo nome="João" />
```

Pense em componentes como peças de Lego. Cada peça pode ser combinada de diferentes maneiras para construir interfaces complexas, mantendo o código modular e reutilizável.



JSX é uma extensão de sintaxe para JavaScript que permite escrever HTML diretamente dentro do código JavaScript. Embora não seja obrigatório, a maioria dos desenvolvedores React usa JSX por ser mais intuitivo e legível.

```
// Exemplo de JSX
const elemento = <h1>Hello, {name}</h1>;

// JSX dentro de uma função
function Saudacao() {
  const nome = "Maria";
  return <div>Olá, {nome}</div>;
}
```

Com JSX, você pode combinar a lógica JavaScript com a marcação HTML de forma direta. Por exemplo, `<div>Hello, {name}</div>` é uma maneira fácil de incluir variáveis JavaScript na UI.

Estados e Props

"Props" (propriedades) são dados passados para os componentes, semelhantes aos parâmetros de uma função. Eles são imutáveis, ou seja, não podem ser alterados pelo componente que os recebe.

```
// Exemplo de JSX
const elemento = <h1>Hello, {name}</h1>;

// JSX dentro de uma função
function Saudacao() {
  const nome = "Maria";
  return <div>Olá, {nome}</div>;
}
```

"Estado" é um objeto que um componente pode manter e alterar ao longo do tempo. Diferente dos props, o estado é gerenciado internamente pelo componente e pode ser atualizado para refletir mudanças na UI.

```
// Exemplo de estado
import { useState } from 'react';

function Contador() {
  const [contagem, setContagem] = useState(0);

  return (
    <div>
      <p>Você clicou {contagem} vezes</p>
      <button onClick={() => setContagem(contagem + 1)}>
        Clique aqui
      </button>
    </div>
  );
}
```



Componentes React têm um ciclo de vida, que consiste em fases como montagem, atualização e desmontagem. Métodos como `componentDidMount` e `componentDidUpdate` permitem executar código em momentos específicos.

```
// Exemplo de ciclo de vida com classe
import React, { Component } from 'react';

class MeuComponente extends Component {
  componentDidMount() {
    // Código a ser executado após o componente ser montado
  }

  componentDidUpdate(prevProps, prevState) {
    // Código a ser executado após o componente ser atualizado
  }

  render() {
    return <div>Meu Componente</div>;
  }
}
```

Por exemplo, você pode usar `componentDidMount` para buscar dados de uma API quando um componente é montado, garantindo que a UI exiba informações atualizadas.

Hooks: Use State e Use Effect 📌

Hooks são funções que permitem usar estado e outros recursos do React sem escrever classes. `useState` permite adicionar estado a componentes funcionais, enquanto `useEffect` lida com efeitos colaterais, como buscar dados ou subscrições.

```
// Exemplo de useState e useEffect
import { useState, useEffect } from 'react';

function ExemploComHooks() {
  const [dados, setDados] = useState(null);

  useEffect(() => {
    fetch('https://api.example.com/dados')
      .then(response => response.json())
      .then(data => setDados(data));
  }, []);

  return (
    <div>
      {dados ? <p>Dados: {dados}</p> : <p>Carregando...</p>}
    </div>
  );
}
```

Os hooks simplificam a gestão de estado e efeitos colaterais, tornando o código mais limpo e fácil de entender, especialmente para iniciantes.



O Virtual DOM é uma representação leve da UI na memória. Quando o estado ou props mudam, o React atualiza o Virtual DOM antes de sincronizá-lo com o DOM real, garantindo atualizações eficientes e rápidas.

```
// Exemplo de renderização eficiente
function App() {
  const [contador, setContador] = useState(0);

  return (
    <div>
      <p>Contagem: {contador}</p>
      <button onClick={() => setContador(contador + 1)}>Incrementar</button>
    </div>
  );
}
```

Essa abordagem minimiza a manipulação direta do DOM, que pode ser lenta, e melhora a performance das aplicações React.



Dominar o React pode parecer um desafio no início, mas com a prática e compreensão dos conceitos fundamentais, você logo verá como ele facilita o desenvolvimento de interfaces de usuário dinâmicas e eficientes. Componentes, props, estado, e hooks são os blocos de construção principais que você utilizará frequentemente.

Imagine que você está construindo uma casa. Os componentes são os tijolos que você usará para criar cada parte da estrutura. Props e estado são como a argamassa que mantém os tijolos juntos, permitindo que a casa mude e se adapte ao ambiente. JSX é a planta da casa, permitindo que você veja claramente como tudo se encaixa. Finalmente, o Virtual DOM é o projeto digital que você verifica antes de fazer mudanças reais, garantindo que tudo esteja perfeito.

Lembre-se, o mais importante é começar pequeno, experimentar, e aprender com os exemplos. À medida que você ganha confiança, pode explorar recursos mais avançados e integrar React com outras tecnologias.

Continue praticando e construindo seus próprios projetos. A prática constante ajudará a solidificar seu conhecimento e transformar você em um desenvolvedor React competente. Boa sorte na sua jornada!!! 🎓

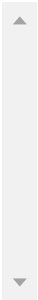


0

12

Comentário

Normal



COMENTAR

Comentários (0)

Seja o primeiro a comentar o artigo do(a) **José Otávio**.

JV

José Otávio

SEGUIR