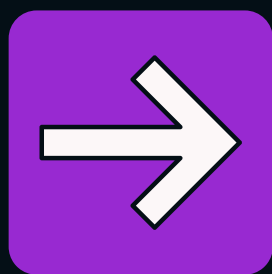


7 DICAS

para
dominar o react



01

Escrevendo HTML no JS?

Sim! O JSX deixa a criação de interfaces mais intuitiva e poderosa. Simplifica a estrutura do seu React!

```
1 export default function Home() {  
2   const BemVindo = () => <h1>Olá, Mundo! 🌍 </h1>;  
3  
4   return <BemVindo />;  
5 }  
6
```

02

Blocos Reutilizáveis = Código Organizado!

React é feito de COMPONENTES!

Crie partes da sua UI que você pode usar em qualquer lugar. Mais organização e menos dor de cabeça!

```
1  function Botao() {  
2    return <button>🚀 Clique Aqui!</button>;  
3  }  
4  export default function Home() {  
5    return <Botao />;  
6  }  
7
```

03

Dados Dinâmicos? Estado! Passando Info? Props!

State guarda dados que mudam. Props enviam dados de pai para filho. Entenda a base para aplicações interativas!

```
1 import { useState } from 'react';
2
3 function Contador() {
4   const [count, setCount] = useState(0);
5   // ...
6 }
```

04

Hooks: Seus Melhores Amigos no React!

useState, useEffect, useContext... Hooks adicionam funcionalidades aos seus componentes. Simplificam e potencializam!

```
1 import { useState } from 'react';
2
3 function Contador() {
4   const [count, setCount] = useState(0);
5   // ...
6 }
```

05

Listas Dinâmicas e Condições? Sem Problemas!

Mostre dados em listas com `.map()` e exiba conteúdo de forma inteligente com renderização condicional.

```
1  const users = ['João', 'Maria', 'Pedro'];
2
3  return (
4    <ul>
5      {users.map((user, index) => (
6        <li key={index}>{user}</li>
7      ))}
8    </ul>
9  );
```

06

Múltiplas Páginas? React Router na Mão!

Crie a navegação da sua aplicação de forma fácil e organizada com o React Router.

```
1 import { BrowserRouter, Route, Routes } from 'react-router-dom';
2
3 function App() {
4   return (
5     <BrowserRouter>
6       <Routes>
7         <Route path="/" element={<Home />} />
8         <Route path="/sobre" element={<Sobre />} />
9       </Routes>
10    </BrowserRouter>
11  );
12 }
```

useMemo, useCallback, Lazy Loading...
Aprenda a deixar suas aplicações mais performáticas e com carregamento sob demanda!

```
1 import { useMemo } from 'react';  
2  
3 function Dobrador({ valor }) {  
4   const resultado = useMemo(() => {  
5     return valor * 2;  
6   }, [valor]);  
7   return <p>O dobro é: {resultado}</p>;  
8 }
```

GOSTOU?
DEIXE NOS
COMENTÁRIOS
SUA DICA!

