

React

Una biblioteca en Javascript para crear interfaces

Betology

10 de junio de 2024

Amor, comprensión y ternura

Tabla de contenidos

1. ¿Qué es React?
2. Componentes de React
3. Props y Estado
4. Ciclo de Vida de los Componentes

¿Qué es React?

- Biblioteca de JavaScript para construir interfaces de usuario.
- Desarrollada por Facebook.
- Utiliza componentes para construir interfaces modulares y reutilizables.

Componentes de React

Definición de un Componente

- Los componentes son bloques de construcción en React.
- Pueden ser definidos como clases o funciones.



```
1 class MyComponent extends React.Component {  
2   render() {  
3     return <h1>Hello, world!</h1>;  
4   }  
5 }
```

Componentes Funcionales

- Más simples y fáciles de escribir.
- No tienen estado propio (antes de los Hooks).



```
1 function MyComponent() {  
2   return <h1>Hello, world!</h1>;  
3 }
```

Props y Estado

Props

- Los ‘props’ son entradas que los componentes reciben de sus padres.
- Permiten pasar datos a los componentes hijos.



```
1 function Greeting(props) {  
2   return <h1>Hello, {props.name}!</h1>;  
3 }  
4  
5 <Greeting name="Alice" />;
```

Estado

- El estado es un objeto privado del componente.
- Permite a los componentes gestionar y responder a cambios de datos.

```
● ● ●

1 class MyComponent extends React.Component {
2     constructor(props) {
3         super(props);
4         this.state = { count: 0 };
5     }
6
7     render() {
8         return (
9             <div>
10                 <p>You clicked {this.state.count} times</p>
11                 <button onClick={() => this.setState({ count: this.state.count + 1 })}>
12                     Click me
13                 </button>
14                 </div>
15         );
}
```

Ciclo de Vida de los Componentes

Métodos del Ciclo de Vida

- 'componentDidMount'
- 'componentDidUpdate'
- 'componentWillUnmount'

Métodos del Ciclo de Vida

- ‘componentDidMount’
- ‘componentDidUpdate’
- ‘componentWillUnmount’
- Permiten ejecutar código en diferentes etapas del ciclo de vida del componente.

Ejemplo de Método del Ciclo de Vida



```
1  class MyComponent extends React.Component {  
2      componentDidMount() {  
3          console.log("Component has mounted");  
4      }  
5  
6      componentDidUpdate() {  
7          console.log("Component has updated");  
8      }  
9  
10     componentWillUnmount() {  
11         console.log("Component will unmount");  
12     }  
13  
14     render() {  
15         return <div>My Component</div>;  
16     }  
17 }
```

Resumen

- React utiliza componentes para construir interfaces modulares.
- Los componentes pueden ser definidos como clases o funciones.
- Props y estado son fundamentales para la gestión de datos en React.
- Los métodos del ciclo de vida permiten manejar eventos en las diferentes etapas del componente.