

Title: What is the difference between state and props?

Answer:

State

This is data maintained inside a component. It is local or owned by that specific component. The component itself will update the state using the `setState()` function.

Example:

```
class Employee extends React.Component {
  constructor() {
    this.state = {
      id: 1,
      name: "Alex"
    }
  }

  render() {
    return (
      <div>
        <p>{this.state.id}</p>
        <p>{this.state.name}</p>
      </div>
    )
  }
}

export default Employee
```

Props

Data passed in from a parent component. `props` are read-only in the child component that receives them. However, callback functions can also be passed, which can be executed inside the child to initiate an update.

Example:

```
class ParentComponent extends Component {
  render() {
    return (
      <ChildComponent name="First Child" />
    )
  }
}

const ChildComponent = (props) => {
  return <p>{props.name}</p>
}
```

Difference between State and Props

Props

Props are read-only.

Props are immutable.

Props allow you to pass data from one component to other components as an argument.

Props can be accessed by the child component.

Props are used to communicate between components.

Stateless component can have Props.

Props make components reusable.

Props are external and controlled by whatever renders the component.

Tags: react-hooks

State

State changes can be asynchronous.

State is mutable.

State holds information about the components.

State cannot be accessed by child components.

States can be used for rendering dynamic changes with the component.

Stateless components cannot have State.

State cannot make components reusable.

The State is internal and controlled by the React Component itself.