State, Props in react.JS

Question 1: What are props in React.js? How are props different from state?

Props (Properties) are used to pass data from a parent component to a child component. They are **read-only** and cannot be changed inside the child component.

Difference Between Props and State:

Feature	Props	State
Data Flow	Passed from parent to child	Managed inside the component
Mutability	Immutable (cannot be changed)	Mutable (can be changed using setState or hooks)
Usage	Used for passing data and functions	Used for managing dynamic data

Example of props:

```
function Message(props) {
  return <h1>Hello, {props.name}!</h1>;
}


// Usage
<Message name="John" />
```

Question 2: What is state in React, and how is it used?

State is used to manage dynamic data inside a component. Unlike props, state can be changed within the component.

Example (Class Component):

```
class Counter extends React.Component {
  constructor() {
    super();
    this.state = { count: 0 };
```

```
  }

  render() {
    return <h1>Count: {this.state.count}</h1>;
  }
}
```

Example (Functional Component with Hooks):

```
function Counter() {
  const [count, setCount] = React.useState(0);
  return <h1>Count: {count}</h1>;
}
```

State helps React track changes and update the UI accordingly.

Question 3: Why is this.setState() used in class components, and how does it work?

this.setState() is used to update the state in class components. It tells React to **re-render** the component with the new state.

Example:

```
class Counter extends React.Component {
  constructor() {
    super();
    this.state = { count: 0 };
  }

  increaseCount = () => {
    this.setState({ count: this.state.count + 1 });
  };

  render() {
```

```
    return (
      <div>
        <h1>Count: {this.state.count}</h1>
        <button onClick={this.increaseCount}>Increase</button>
      </div>
    );
  }
}
```

How it works:

1. this.setState() updates the state.
2. React detects the change and **re-renders** the component.
3. The new state value is shown in the UI.

In short, this.setState() helps **update and manage dynamic data** in class components. 🚀

THEORY EXERCISE

3. Components(Functional &Class Components)

Question 1: What are components in React?

Components in React are reusable building blocks of a UI. They help break the UI into smaller, manageable parts.

Difference Between Functional and Class Components:

Feature	Functional Component	Class Component
Definition	JavaScript function	JavaScript class
State	Uses useState hook	Uses this.state
Lifecycle Methods	Uses hooks like useEffect	Uses lifecycle methods like componentDidMount
Syntax Simplicity	Simple and concise	More code and complexity
Performance	Faster and lightweight	Slightly heavier

Question 2: How do you pass data to a component using props?

You pass data to a component using **props** (short for properties). Props are passed as attributes in JSX.

- **Example:**

```
function Greeting(props) {
  return <h1>Hello, {props.name}!</h1>;
}

// Usage:
<Greeting name="John" />
```

Here, "John" is passed as a prop and used inside the Greeting component.

Question 3: What is the role of render() in class components?

The render() method in class components is responsible for returning JSX, which defines what gets displayed on the screen.

- **Example:**

```
class Welcome extends React.Component {
  render() {
    return <h1>Welcome to React!</h1>;
  }
}
```

In short, render() tells React **what to show** in the UI.