**React State**

**What is State in React?**  
**State** is a built-in React object used to store **dynamic data** in a component. It allows a component to **remember information** and **react to user input**, events, or other changes over time.

Unlike **props**, which are passed from parent to child and are read-only, **state is managed and updated within the component itself**.

**Why State is Important:**

* It enables **interactive** and **dynamic behavior** in components.
* When the **state changes**, React automatically **re-renders the component** to reflect the new data.
* It is essential for handling:
  + Form inputs
  + Toggle buttons
  + Counters
  + API responses
  + Conditional rendering

**How to Use State**

**In Function Components (using Hooks):**

import { useState } from 'react';

function Counter() {

const [count, setCount] = useState(0);

return (

<div>

<p>You clicked {count} times</p>

<button onClick={() => setCount(count + 1)}>Click Me</button>

</div>

);

}

* useState(0) initializes the state with a value of 0.
* setCount is used to update the state.
* Updating the state causes the component to re-render.

**In Class Components:**

class Counter extends React.Component {

constructor(props) {

super(props);

this.state = { count: 0 };

}

render() {

return (

<div>

<p>You clicked {this.state.count} times</p>

<button onClick={() => this.setState({ count: this.state.count + 1 })}>

Click Me

</button>

</div>

); }

}

**Key Characteristics of State:**

* **Private**: State is local to the component unless shared via props.
* **Dynamic**: Changes over time based on user actions or logic.
* **Triggers re-render**: Updating state re-renders the component with new data