ReactJS Concept - React State

# Explain React State

React State is an object that represents the parts of an app that can change.

Each component can maintain its own state, which lives in an object called `this.state` in class components.

When the state of a component changes, the component re-renders to reflect the new state.

State is used to make components interactive and dynamic. For example, tracking user input, toggling views, updating counters, etc.

### Key Characteristics of React State:

- \*\*Mutable\*\*: Unlike props, state can be changed using `this.setState()` method.

- \*\*Local to component\*\*: The state is private and fully controlled by the component.

- \*\*Triggers re-render\*\*: Whenever the state is updated, the component re-renders with the new state.

### Defining State:

In class components, state is usually initialized in the constructor.

Example:

```

class Counter extends React.Component {

constructor(props) {

super(props);

this.state = { count: 0 };

}

}

```

### Updating State:

Use `this.setState()` method to change the value of state.

Example:

```

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

```

### Use Case:

React State is ideal for data that changes over time, such as form inputs, button clicks, toggling views, etc.

React also supports hooks like `useState` in functional components for managing state.