

MODULE-9 REACT

Components (Functional & Class Components)

THEORY EXERCISE

Question 1:

What are components in React? Explain the difference between functional components and class components.

Answer:

Components are the building blocks of a React application. They let you split the UI into independent, reusable pieces that can be thought of as custom, isolated HTML elements.

- Functional Components:

These are JavaScript functions that accept props as input and return React elements (JSX). They are simple, stateless (until hooks were introduced), and easier to write.

- Class Components:

These are ES6 classes that extend `React.Component`. They have lifecycle methods, state management, and a required `render()` method that returns JSX. Class components are more feature-rich but more verbose.

Question 2:

How do you pass data to a component using props?

Answer:

Props (short for properties) are used to pass data from a parent component to a child component in React. You pass props as attributes in JSX, and the child component accesses them via a props object.

Example:

```
<Greeting name="Divya" />
```

Inside Greeting component, `props.name` will be "Divya". Props are read-only and cannot be changed by the child component.

MODULE-9 REACT

Question 3:

What is the role of `render()` in class components?

Answer:

The `render()` method is a required lifecycle method in class components. It defines what the UI should look like by returning JSX (React elements). React calls `render()` whenever the component's state or props change, so the UI stays in sync with data.

Props and State

THEORY EXERCISE

Question 1:

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

Answer:

Props are inputs passed to components from their parents. They are read-only, meaning a component cannot modify its own props. Props are used to pass data and event handlers down the component tree.

- State:

State is data managed inside a component. It is mutable and can be updated using `setState()` in class components or `useState` hook in functional components. Changes to state cause the component to re-render.

Difference:

Props are immutable and passed from parent to child, while state is mutable and managed within the component.

Question 2:

Explain the concept of state in React and how it is used to manage component data.

MODULE-9 REACT

Answer:

State is a JavaScript object that holds dynamic data for a component. It allows components to create and manage their own data that can change over time in response to user actions or network responses.

In class components, state is initialized in the constructor or as a class field, and updated via `this.setState()`. In functional components, the `useState` hook is used. Whenever state changes, React re-renders the component to update the UI.

Question 3:

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

Answer:

`this.setState()` is a method used to update the component's state object in class components. It schedules an update to the component's state and tells React that the component needs to re-render with the new state.

`setState()` merges the new state with the existing state asynchronously, ensuring efficient UI updates. You should never modify state directly because it does not trigger a re-render.