React - Components, State, Props Assignment

Q:1 What are components in React? Explain the difference between functional components and class components.

A:- Component: Components are independent and reusable bits of code. They are javascript functions and return HTML element. Components come in two types, Class components and Function components.

Function component: These are simple JavaScript functions that take props as input and return JSX elements. Functional components do not have a state or lifecycle methods. They are faster than class components. functional components use React hooks i.e. useState(). React lifecycle methods (for example, componentDidMount) cannot be used in functional components.

Class component: Class components are written as a JavaScript class. Class components are a bit more complex. Class components have a state and can implement lifecycle methods like componentDidMount and componentDidUpdate. They are slower than function component. because they're more complex, they can be harder to read and write.

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

A:- React components use *props* to communicate with each other. Every parent component can pass some information to its child components by giving them props.

In the Parent Component: Pass the data as attributes when rendering the child component.

```
<ChildComponent propName1={value1} propName2={value2} />
```

In the Child Component: Access the passed props using the props object or destructuring the properties.

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

A:- → The role of the function is to display the specified HTML code inside the specified HTML element.

→In the render() method, we can read props and state and return our JSX code to the root component of our a app.

→ In the render() method, we cannot change the state.

Props and State

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

A:- props: Props are arguments passed into React components. React Props are like function arguments in JavaScript *and* attributes in HTML.

State: State holds information about the components. State cannot be accessed by child components. Stateless components cannot have State. The State is internal and controlled by the React Component itself.

Props: Props allow you to pass data from one component to other components as an argument. Props can be accessed by the child component. Stateless component can have Props. Props are external and controlled by whatever renders the component .

Q:2 Explain the concept of state in React and how it is used to manage componentdata.

A:- In React, state refers to an object used to store data or information about the component that may change over time.

Managing componentdata with state:

1. Initialization: State is initialized with default values when the component is created.
2. Updating State: State can be updated through user actions (e.g., button clicks), lifecycle events.
3. Conditional Rendering: Components can render different outputs based on the current state.

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

A:- In react class components, this.setState() is the method used to update the component’s state.

Why this.setState() is Used:

1. State Updates Trigger Re-renders: When this.setState() is called, React schedules an update to the component, re-renders it, and reflects the updated state in the UI.
2. Merges State Updates: this.setState() merges the new state with the existing state, so you only need to specify the parts of the state that are changing
3. Immutable State Updates: State in React should not be updated directly (e.g., this.state.value = newValue). Instead, this.setState() ensures that the state is updated immutably, maintaining the integrity of React's internal processes.
Syntax: this.setState(updater[, callback])