**WEEK6**

**1. Explain React Components**

React **components** are the building blocks of any React application. They allow you to split the UI into independent, reusable pieces that can be managed separately.

Each component:

* Can have its own logic and UI
* Accepts input via **props**
* Can maintain its own **state** (especially in class or functional components with hooks)

**2. Identify the Differences Between Components and JavaScript Functions**

| **Feature** | **React Components** | **JavaScript Functions** |
| --- | --- | --- |
| Purpose | Used to build UI elements in React | Used to perform tasks or calculations |
| Return Value | Returns JSX (React elements) | Returns any data type |
| State Management | Can use state (in class components or hooks) | Do not support state natively |
| Lifecycle Methods | Present (in class components) | Not applicable |
| JSX Usage | Commonly returns JSX | Does not return JSX |

**3. Identify the Types of Components**

There are mainly two types of React components:

* **Class Components**: ES6 classes that extend React.Component and include lifecycle methods.
* **Function Components**: Simple JavaScript functions that return JSX. Modern function components can use **hooks** for state and lifecycle features.

**4. Explain Class Component**

A **Class Component** is a more traditional way of writing components in React using ES6 classes.

**Key Features:**

* Extends React.Component
* Must include a render() method that returns JSX
* Can use **state** and **lifecycle methods**

**Example:**

class Welcome extends React.Component {

constructor(props) {

super(props);

this.state = { message: "Hello" };

}

render() {

return <h1>{this.state.message}, {this.props.name}</h1>;

}

}

**5. Explain Function Component**

A **Function Component** is a simpler and more modern way to write components in React using plain functions.

**Key Features:**

* Accepts props as an argument
* Returns JSX
* Can use **hooks** like useState and useEffect for state and side effects

**Example:**

javascript

function Welcome(props) {

return <h1>Hello, {props.name}</h1>;

}

Using hooks:

javascript

import React, { useState } from 'react';

function Counter() {

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

return (

<div>

<p>Count: {count}</p>

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

</div>

);

}

**6. Define Component Constructor**

The **constructor** is a special method in class components that is called when the component is created.

**Purpose:**

* Initialize state
* Bind methods

**Syntax:**

javascript

constructor(props) {

super(props);

this.state = { count: 0 };

}

Note: The super(props) call is required to access this.props in the constructor.

**7. Define render() Function**

The render() function is a mandatory method in every class component.

**Purpose:**

* Describes what the UI should look like
* Returns **JSX**, which React then converts to actual DOM elements

**Example:**

javascript

render() {

return <h1>Hello, world!</h1>;

}

