**1. Explain React Components**

**React Components** are the building blocks of a React application.  
They define **how a part of the UI should appear and behave**.

* Components allow you to **split the UI into independent, reusable pieces**.
* Each component is a **JavaScript function or class** that returns JSX (UI markup).
* Components can have **state**, **props**, and **lifecycle methods** (in class components).

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

| **Feature** | **React Component** | **JavaScript Function** |
| --- | --- | --- |
| Returns | JSX (UI code) | Any value |
| Purpose | Defines UI and behavior | Performs a task or calculation |
| Props Handling | Receives props as input | Receives parameters |
| Lifecycle | Has lifecycle methods (in class components) | No built-in lifecycle |
| Side Effects | Use useEffect() (in functional components) | Handled manually |
| State | Can manage state | No built-in state management |

**3. Identify the Types of Components**

There are **two main types** of React components:

1. **Class Components**
2. **Function Components**

**4. Explain Class Component**

A **Class Component** is a traditional way of creating components using ES6 classes.

**Syntax:**

import React, { Component } from 'react';

class MyComponent extends Component {

constructor(props) {

super(props);

this.state = { count: 0 };

}

render() {

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

}

}

**Key Features:**

* Uses render() method to return JSX.
* Can manage internal state (this.state).
* Has access to lifecycle methods (e.g., componentDidMount()).

**5. Explain Function Component**

A **Function Component** is a simpler way to write components using plain functions.

**Syntax:**

function MyComponent(props) {

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

}

Or with **Hooks**:

import { useState } from 'react';

function Counter() {

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

return <button onClick={() => setCount(count + 1)}>Count: {count}</button>;

}

**Key Features:**

* No this keyword needed.
* Simpler and cleaner syntax.
* Uses **React Hooks** (useState, useEffect) for state and lifecycle logic.

**6. Define Component Constructor**

**Constructor:**  
A special method used in **class components** to initialize state and bind methods.

constructor(props) {

super(props);

this.state = { count: 0 };

}

* Called once when the component is created.
* super(props) must be called to access this.props.

**7. Define render() Function**

**render():**  
A method in **class components** that tells React what to display on the screen.

render() {

return <div>Hello World</div>;

}

* Returns **JSX** to be rendered in the DOM.
* Called every time the component needs to update its UI.