**1. Explain React Components**

**React components** are the building blocks of a React application’s UI. A component is a self-contained module that renders some output based on input called **props** and internal **state** (if applicable).

* They help break the UI into **reusable**, **independent** pieces.
* Each component can manage its own state and logic.
* Components can be **nested**, reused, and composed to build complex UIs.

2. Identify the Differences Between Components and JavaScript Functions

| **Feature** | **React Components** | **JavaScript Functions** |
| --- | --- | --- |
| **Purpose** | Build UI and manage state/props | Perform computations or operations |
| **Return Type** | Return JSX (UI elements) | Return any data (number, string, etc.) |
| **Lifecycle Methods** | Have lifecycle hooks (in class components) | No lifecycle methods |
| **Use of JSX** | Commonly use JSX | Don't use JSX |
| **React Context** | Have access to props, hooks, state | Don't manage UI state |
| **Example Use** | <MyComponent /> | myFunction() |

**3. Identify the Types of Components**

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

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

Modern React mostly favors **Function Components** with **Hooks**, but Class Components are still valid.

**4. Explain Class Component**

A **class component** is a React component defined using ES6 class syntax and extends React.Component.

**Example:**

import React, { Component } from 'react';

class Welcome extends Component {

render() {

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

}

}

**Key Features:**

* Has access to lifecycle methods like componentDidMount(), componentDidUpdate(), etc.
* Uses this.state and this.setState() for state management.
* Uses render() method to return JSX.

**5. Explain Function Component**

A **function component** is a simpler way to write components using regular JavaScript functions.

**Example:**

function Welcome(props) {

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

}

**Key Features:**

* Simpler syntax.
* No this keyword.
* Can use **Hooks** like useState, useEffect.

**6. Define Component Constructor**

The **constructor** is a special method in **class components** used for:

* Initializing **state**
* Binding methods
* Accessing **props**

**Example:**

class MyComponent extends React.Component {

constructor(props) {

super(props);

this.state = { count: 0 };

}

}

**7. Define render() Function**

The **render() function** is a required method in **class components**. It returns the JSX (or UI) to be displayed.

**Example:**

class MyComponent extends React.Component {

render() {

return <h1>Hello World</h1>;

}

}

**Key Points:**

* Should return only **one parent element** (wrap multiple elements in a <div> or <React.Fragment>).
* Called automatically by React to update the UI when state or props change.