**Explain React components**

React components are reusable building blocks of a React application’s UI. They let you split the UI into independent, reusable pieces, and think about each piece in isolation. Each component can have its own structure (HTML), logic (JavaScript), and styling (CSS).

**Differences Between Components and JavaScript Functions**

| **Feature** | **JavaScript Function** | **React Component** |
| --- | --- | --- |
| Purpose | Performs logic or computation | Describes UI and how it should appear |
| Returns | Any data (number, string, object, etc.) | JSX (React elements) |
| Naming Convention | camelCase (e.g., calculateSum) | PascalCase (e.g., MyComponent) |
| Lifecycle Methods | Not applicable | Has lifecycle methods (class components) |
| Hooks Support | Not applicable | Supported in function components |

**Types of Components**

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

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

**Explain Class Component**

A **class component** is a React component defined as a **JavaScript class**. It can hold **state** and has access to **lifecycle methods** like componentDidMount, componentDidUpdate, etc.

**Explain Function Component**

A function component is a React component defined as a JavaScript function. It can use React Hooks (like useState, useEffect) to manage state and side effects.

**Define Component Constructor**

In **class components**, the **constructor** is a special method used to:

* Initialize state
* Bind methods

**Define render() Function**

The render() function is **mandatory** in class components. It defines **what gets displayed** on the screen.

**Key points:**

* Must return a **single JSX element**.
* React calls render() whenever state or props change.