1. React Components

In React, **components** are the building blocks of a user interface. A component is a **self-contained piece of UI** that can have its own logic, state, and rendering. Think of components as **custom HTML elements** that can be reused, nested, and composed to build complex interfaces.

For example, in a blog page:

- <Header />, <Post />, <Sidebar />, and <Footer /> can all be individual components.

Each React component:

- **Receives data via props**
- **Manages its own state (if needed)**
- **Returns JSX to render UI**

2. Difference Between React Components and JavaScript Functions

While **React function components** are written like JavaScript functions, they are **not exactly the same**.

Aspect	JavaScript Function	React Component
Purpose	Performs a task or returns a value	Returns UI (JSX)
Output	Data (number, string, etc.)	JSX (React elements)
Usage	Called directly	Used in JSX like <Component />
Lifecycle	No lifecycle	Has lifecycle (via hooks or class methods)
React Integration	Cannot manage React state or props	Designed to work with React state, props, context
Return Value	Any JS type	JSX or null

3. Types of Components in React

There are two main types of components:

A. Class Components

- Defined using ES6 class syntax
- Can use lifecycle methods (**componentDidMount, render**, etc.)
- Can hold local **state**

B. Function Components

- Declared as JavaScript functions
- Use **React Hooks** for state and lifecycle (e.g., **useState, useEffect**)
- More concise and preferred in modern React

4. Explain Class Component

Class components are traditional React components that extend from **React.Component** and contain:

- A constructor for initializing state
- A render() method for returning JSX
- Optional lifecycle methods (like **componentDidMount**, **componentWillUnmount**)

5. Explain Function Component

A function component is a simpler way to write components using a regular JavaScript function. They:

- Accept props as arguments
- Return JSX
- Can use hooks like **useState**, **useEffect** to handle state and side effects

6. Define Component Constructor

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

- State
- Method bindings

It is called **before** the component is mounted.

7. Define render() Function

The render() function is **mandatory** in class components. It returns the **JSX** that will be displayed on the screen.