| 2. ReactJS-HOL |
| --- |

**React Components**

A React component is a reusable, self-contained piece of UI that can manage its own structure, behavior, and styling.

Think of it like a building block of a React application — similar to a function, but it returns JSX (UI), not just a value.

**Differences Between React Components and JavaScript Functions**

A React component is primarily designed to build and manage user interfaces. It returns JSX, which is a syntax extension that looks like HTML and is used to describe the UI structure. Components are reusable, stateful, and integrated into the React ecosystem. They can manage local state using hooks like useState and perform side effects using useEffect. React components can also participate in the component lifecycle, which includes mounting, updating, and unmounting phases. This enables developers to control how the component behaves throughout its existence in the DOM.

On the other hand, a JavaScript function is a block of reusable code meant to perform a specific task or return a value. It does not have access to JSX or React’s lifecycle by default, and cannot manage state in the way React components can. JavaScript functions typically return primitive values, objects, or arrays, and are used for logic and computations rather than UI.

**Types of React Components**

1. **Functional Components** (Modern, recommended)
2. **Class Components** (Older, still valid)

**Class Component in React**

A class component in React is a more traditional way of creating components. It is built using JavaScript ES6 classes and extends the React.Component base class. Class components must define a render() method, which returns the JSX that describes what should appear on the screen. These components have access to important features like state management and lifecycle methods such as componentDidMount, componentDidUpdate, and componentWillUnmount. The constructor method is often used in class components to initialize state and bind event handlers.

Class components were the only way to manage local state and side effects before React 16.8 introduced hooks. Although function components are now more common due to their simplicity and support for hooks, class components are still widely used in older codebases and for complex component structures.

**Function Component in React**

A function component is a simpler and more modern way to define components in React. It is just a plain JavaScript function that takes props as an argument and returns JSX. Initially, function components were stateless, meaning they couldn't manage internal state or use lifecycle features. However, with the introduction of React Hooks like useState, useEffect, and others, function components became fully capable of handling state and side effects, making them powerful and concise.

Function components are easier to read, write, and test compared to class components. They also lead to cleaner code with fewer lines and improved performance in some cases. Due to these advantages, function components have become the standard in modern React development.

**Component Constructor**

The constructor() is a special function in class components used for:

* Initializing state
* Binding methods
* Accessing props

**render() Function**

The render() function is a required method in every class component. It defines:

* What the component returns (UI)
* It must return JSX

Every time state or props change, React calls render() again.









