

React

1. Explain React Components

React components are the fundamental building blocks of a React application. They allow developers to split the UI into independent, reusable pieces that can be managed separately. Each component operates like a JavaScript function but returns HTML (or more precisely, JSX) to describe what should appear on the screen.

Components help in:

- Reusing code across the application.
- Managing the application's UI in a modular way.
- Maintaining a clear separation of concerns.

There are two main types of components: **Class Components** and **Function Components**.

2. Identify the Differences Between Components and JavaScript Functions

Although React components are written in JavaScript and resemble functions, there are important differences:

Aspect	JavaScript Function	React Component
Purpose	Performs computation or returns a value	Returns JSX (UI structure)
Return Type	Returns primitive values or objects	Returns JSX (React elements)
Reusability in UI	Not designed for UI reuse	Designed to build reusable UI elements
Lifecycle Methods	Not applicable	Class components have lifecycle methods
Hooks Usage	Not applicable	Function components can use React Hooks
JSX Support	Not used	Directly returns JSX

3. Identify the Types of Components

React supports the following main types of components:

1. Function Components

- Simple JavaScript functions.
- Return JSX.
- Can use hooks for state and lifecycle features.

2. Class Components

- ES6 classes that extend `React.Component`.
 - Use lifecycle methods and `this.state`.
-

4. Explain Class Component

A class component is a more traditional way to define a component using ES6 classes. It must extend `React.Component` and include a `render()` method that returns JSX.

Example:

```
import React, { Component } from 'react';

class Welcome extends Component {
  render() {
    return <h1>Hello, {this.props.name}</h1>;
  }
}
```

Features:

- Supports state management using `this.state`.
 - Lifecycle methods like `componentDidMount`, `componentDidUpdate`, and `componentWillUnmount` are available.
 - Accesses props using `this.props`.
-

5. Explain Function Component

Function components are JavaScript functions that return JSX. They are simpler and preferred in modern React development due to their lightweight nature and the ability to use hooks.

Example:

```
JavaScript ∨  
  
function Welcome(props) {  
  return <h1>Hello, {props.name}</h1>;  
}
```

Advantages:

- Less code and boilerplate.
 - Easier to read and test.
 - Can use hooks like useState, useEffect, etc.
-

6. Define Component Constructor

The **constructor** in a class component is a special method used to initialize state and bind methods. It is called before the component is mounted.

Syntax:

```
JavaScript ∨  
  
constructor(props) {  
  super(props);  
  this.state = {  
    message: "Welcome"  
  };  
}
```

Key Points:

- super(props) must be called before accessing this.
 - It is used to initialize this.state.
 - It's commonly used to bind event handler methods.
-

7. Define render() Function

The render() method is a mandatory part of every class component. It returns the JSX that defines the UI layout of the component.

Example:

```
JavaScript ∨  
  
render() {  
  return (  
    <div>  
      <h1>{this.state.message}</h1>  
    </div>  
  );  
}
```

Characteristics:

- Called every time the component needs to re-render.
- Should be a pure function (i.e., not causing side effects).
- Can return only a single parent element (wrap multiple elements inside a <div> or React fragment).