React Components - Detailed Explanation

## Explain React components

React components are independent, reusable pieces of code used to build the UI in a React application. Each component returns JSX to describe what the UI should look like. Components can be nested, managed, and handled independently.

## Identify the differences between components and JavaScript functions

React components return JSX and are used specifically to define UI logic and structure. They can have state, props, and lifecycle methods (in class components). JavaScript functions, on the other hand, are general-purpose code blocks that do not inherently support UI rendering or state management.

## Identify the types of components

There are two types of React components:  
1. Class Components: ES6 classes that extend React.Component.  
2. Function Components: JavaScript functions that return JSX.

## Explain class component

Class components are more traditional React components defined using ES6 classes. They must include a render() method and can maintain internal state and lifecycle methods like componentDidMount or componentDidUpdate.

## Explain function component

Function components are declared as plain JavaScript functions. They are simpler and more concise than class components. With the addition of React Hooks, function components can now manage state and side effects.

## Define component constructor

The constructor in a class component is used to initialize state and bind methods. It’s called before the component is mounted and should call super(props) before any other statement.

## Define render() function

The render() function is mandatory in class components. It returns the JSX that should be rendered in the UI. It is called every time the component's state or props change.