Functional Components Vs. Class Components

**Introduction:** React is a widely used JavaScript library for building user interfaces, and it highlights the two types of components available in React: **Class components and Functional components.** Although both are used to achieve the same purpose, there are significant differences between the two that need to be understood by us.

**Class Components:** Class components are defined using ES6 classes, and they have access to the state and lifecycle methods of React. They use the "this" keyword to refer to the component instance and can have their state changed using the "setState()" method. Class components can also use lifecycle methods such as "componentDidMount()",

"componentDidUpdate()", and "componentWillUnmount()" to perform actions at specific points in the component lifecycle.

**Functional Components:** Functional components are defined using JavaScript functions. They do not have access to state or lifecycle methods and are primarily used for presenting data. They take in props as an argument and return React elements. Since functional components do not have a state, they are simpler and more lightweight than class components.

**Differences:**

|  |  |
| --- | --- |
| **Functional Component** | **Class Component** |
| **Syntax :** JavaScript Functions. | **Syntax :** JavaScript Classes (ES6). |
| Functional components do not have access to the state and lifecycle methods. | Class components have access to the state and lifecycle methods |
| **Performance:** Functional components are faster and more lightweight | Class components since they do not have a state or lifecycle methods to manage. |
| Functional components can be easily reused across different parts of an application. | Class components require more code to be reused. |

**Conclusion:** Both Class Components and Functional Components can be used to build user interfaces, but they differ significantly in terms of syntax, state, lifecycle methods, performance, and code reusability. We need to choose the right type of component for our application based on its requirements and complexity.