**What are some differences between class and functional components?**

**Syntax**: Class Components are defined using JavaScript ES6 classes

Functional Components are defined using functions

**State**: In Class Components state is managed using this.state

In Functional Components state managed using useState

**Lifecycle Methods**: Class components can use lifecycle methods e.g. “componentDidMount”, “componentDidUpade”, “componentWillUnmount”.

Functional Components don’t use lifecycle methods. However, useEffect can mimic the Lifecycle Methods.

**`this` keyword**: Class components use `this` keyword to access props and states

Functional components directly access props and states

**Explain what lifecycle is in a simple way. How do you manage it in class and functional components?**

In React, the lifecycle of a component refers to the series of methods that are automatically invoked at different stages of a component's existence. These stages include:

1. **Mounting**: When the component is being created and inserted into the DOM.
2. **Updating**: When the component is being re-rendered due to changes in state or props.
3. **Unmounting**: When the component is being removed from the DOM.

Managing Lifecycle in Class Components:

We manage Lifecycle in Class Components using the following methods:

`componentDidMount()`: It is invoked only once when component is first time loaded into the DOM

`componentDidUpdate`: It is invoked when there is a state change in the component

`componentWillUnmount()`: It is invoked when the component is removed from the DOM.

Managing Lifecycle in Class Components:

In Functional Components `useEffect` hook can be used to mimic the Life Cycle functions used in Class Components.

useEffect with empty dependency array runs the effect only once during the initial render. It is like ` componentDidMount()`

useEffect with a non-empty dependency array runs the effect every time when there is a state change in the dependency. It is similar to `componentDidUpdate.

useEffect returns an optional function (clean up function) that react will call when the component unmounts. It is like `componentWillUnmount()`

**Explain immutability in one sentence.**

Immutability means that once an object or value is created, it cannot be changed or modified.