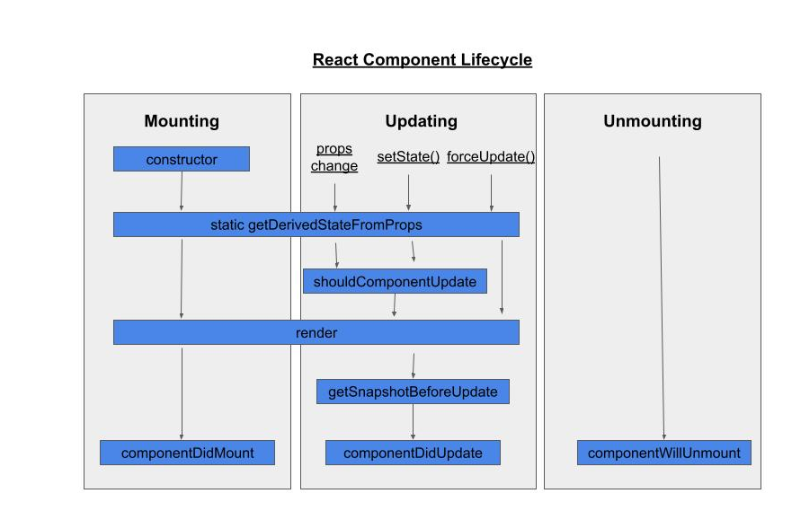
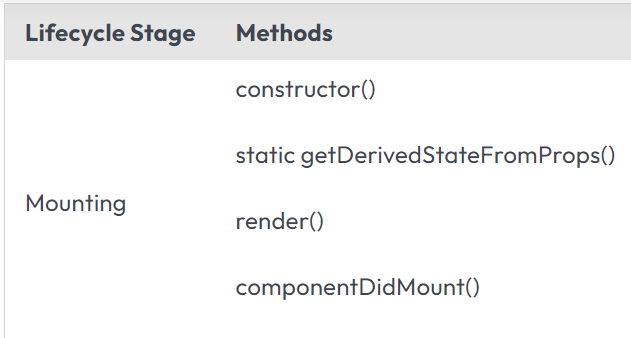


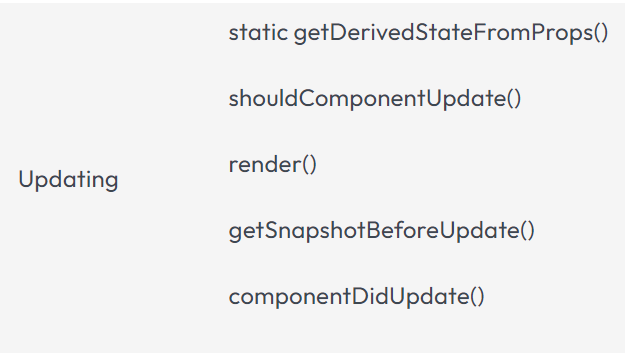
A React component have 3 phases its lifecycle:

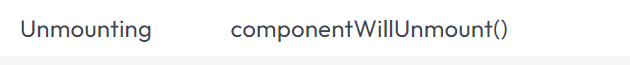
-🡪 mounting, updating, and unmounting.

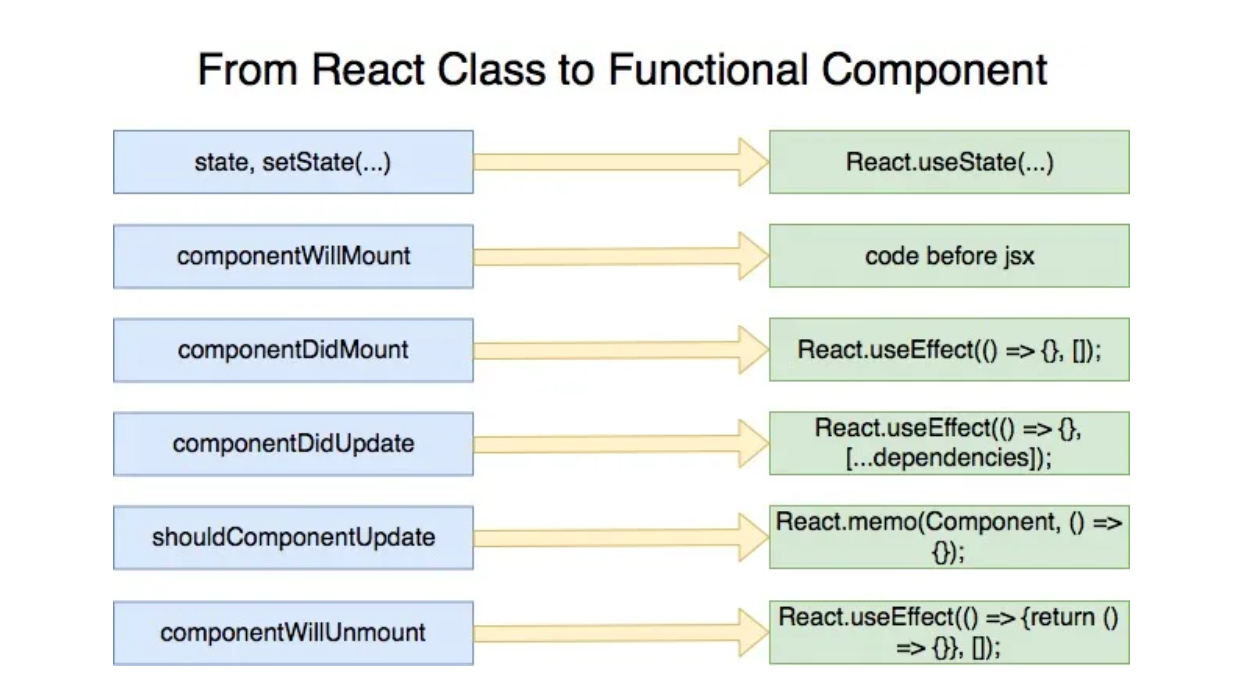
1. The *mounting phase* is when a new component is created and inserted into the DOM or, in other words, when the life of a component begins. This can only happen once, and is often called “initial render.”
2. The *updating phase* is when the component updates or re-renders. This reaction is triggered when the props are updated or when the state is updated. This phase can occur multiple times, which is kind of the point of React.
3. The last phase within a component's lifecycle is the *unmounting phase*, when the component is removed from the DOM.





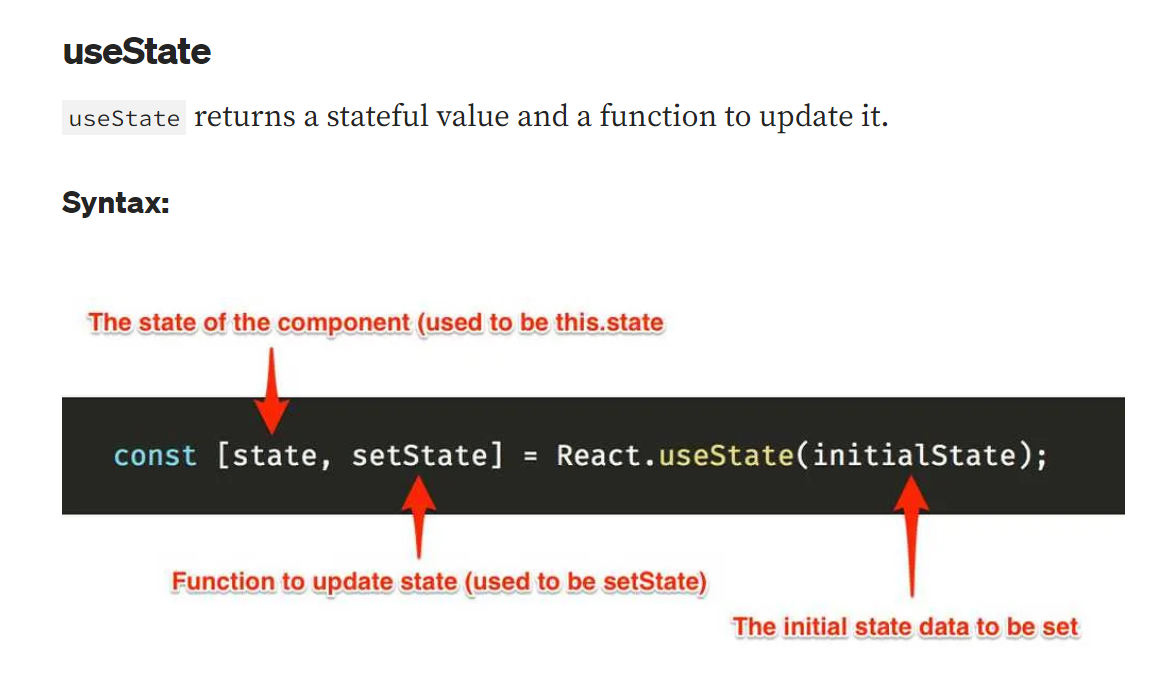






Before we start, let’s get some understanding about following hooks:

* useState
* useEffect
* UseContext



**useEffect**

With functional component, we are not **allowed** to have side effects like mutations, timers, mutations or any other side effects inside the main body of the component.

So, here we have useEffect which runs **after** the render happens. By default, effects run after every render but there are ways to control it which we will look into it later in the article.

