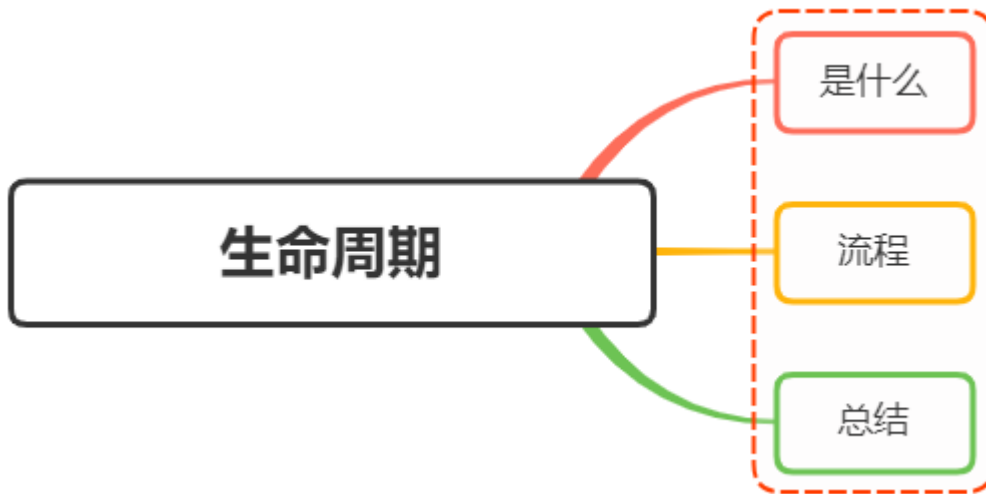


# 面试官：说说 React 生命周期有哪些不同阶段？每个阶段对应的方法是？

---



## 一、是什么

在[以前文章](#)中，我们了解到生命周期定义

生命周期（**Life Cycle**）的概念应用很广泛，特别是在经济、环境、技术、社会等诸多领域经常出现，其基本涵义可以通俗地理解为“从摇篮到坟墓”（**Cradle-to-Grave**）的整个过程

跟Vue一样，**React**整个组件生命周期包括从创建、初始化数据、编译模板、挂载Dom→渲染、更新→渲染、卸载等一系列过程

## 二、流程

这里主要讲述**react16.4**之后的生命周期，可以分成三个阶段：

- 创建阶段
- 更新阶段
- 卸载阶段

### 创建阶段

创建阶段主要分成了以下几个生命周期方法：

- `constructor`
- `getDerivedStateFromProps`
- `render`
- `componentDidMount`

#### **constructor**

实例过程中自动调用的方法，在方法内部通过**super**关键字获取来自父组件的**props**

在该方法中，通常的操作为初始化`state`状态或者在`this`上挂载方法

## getDerivedStateFromProps

该方法是新增的生命周期方法，是一个静态的方法，因此不能访问到组件的实例

执行时机：组件创建和更新阶段，不论是`props`变化还是`state`变化，也会调用

在每次`render`方法前调用，第一个参数为即将更新的`props`，第二个参数为上一个状态的`state`，可以比较`props` 和 `state`来加一些限制条件，防止无用的`state`更新

该方法需要返回一个新的对象作为新的`state`或者返回`null`表示`state`状态不需要更新

## render

类组件必须实现的方法，用于渲染`DOM`结构，可以访问组件`state`与`prop`属性

注意：不要在 `render` 里面 `setState`, 否则会触发死循环导致内存崩溃

## componentDidMount

组件挂载到真实`DOM`节点后执行，其在`render`方法之后执行

此方法多用于执行一些数据获取，事件监听等操作

## 更新阶段

该阶段的函数主要为如下方法：

- `getDerivedStateFromProps`
- `shouldComponentUpdate`
- `render`
- `getSnapshotBeforeUpdate`
- `componentDidUpdate`

## getDerivedStateFromProps

该方法介绍同上

## shouldComponentUpdate

用于告知组件本身基于当前的`props`和`state`是否需要重新渲染组件，默认情况返回`true`

执行时机：到新的`props`或者`state`时都会调用，通过返回`true`或者`false`告知组件更新与否

一般情况，不建议在该周期方法中进行深层比较，会影响效率

同时也不能调用`setState`，否则会导致无限循环调用更新

## render

介绍如上

## getSnapshotBeforeUpdate

该周期函数在`render`后执行，执行之时`DOM`元素还没有被更新

该方法返回的一个`Snapshot`值，作为`componentDidUpdate`第三个参数传入

```
getSnapshotBeforeUpdate(prevProps, prevState) {  
  console.log('#enter getSnapshotBeforeUpdate');  
  return 'foo';  
}  
  
componentDidUpdate(prevProps, prevState, snapshot) {  
  console.log('#enter componentDidUpdate snapshot = ', snapshot);  
}
```

此方法的目的在于获取组件更新前的一些信息，比如组件的滚动位置之类的，在组件更新后可以根据这些信息恢复一些UI视觉上的状态

## componentDidUpdate

执行时机：组件更新结束后触发

在该方法中，可以根据前后的`props`和`state`的变化做相应的操作，如获取数据，修改`DOM`样式等

卸载阶段

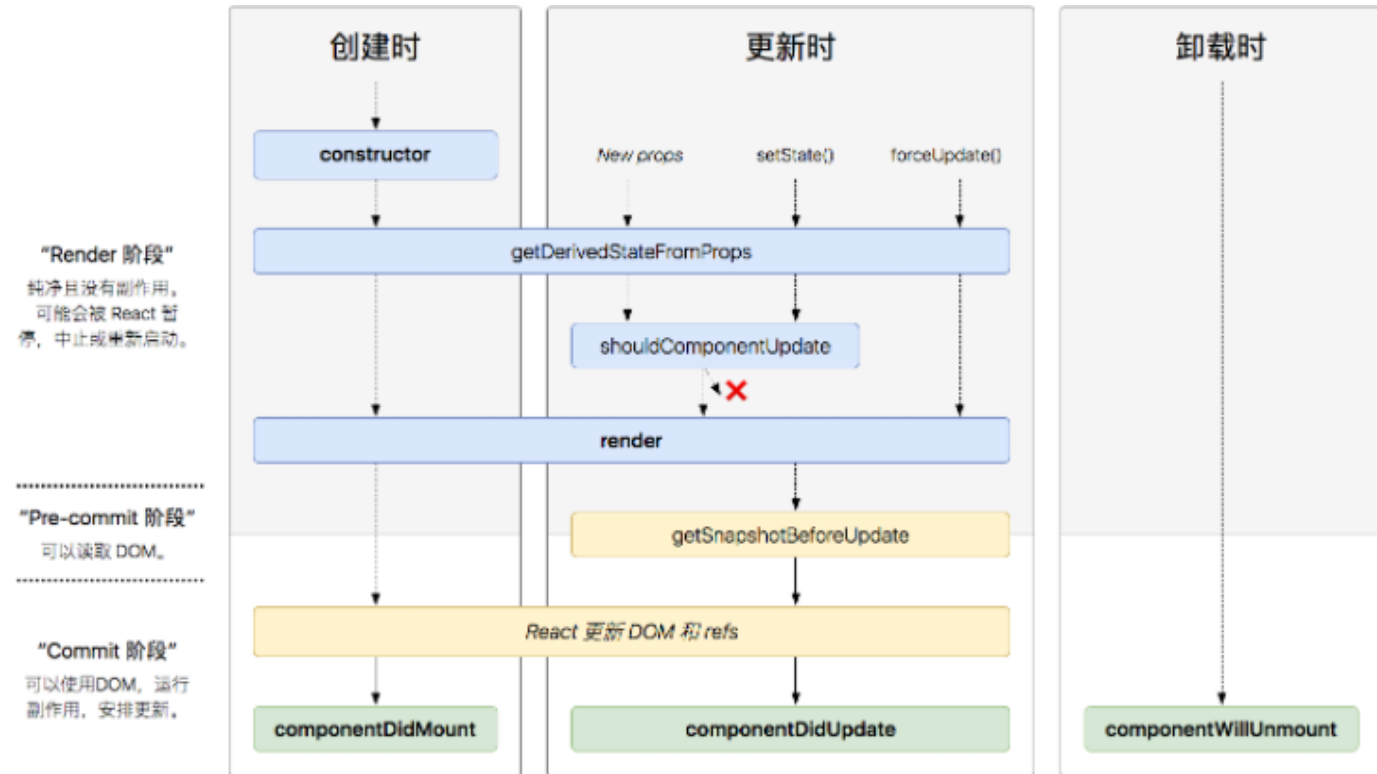
## componentWillUnmount

此方法用于组件卸载前，清理一些注册是监听事件，或者取消订阅的网络请求等

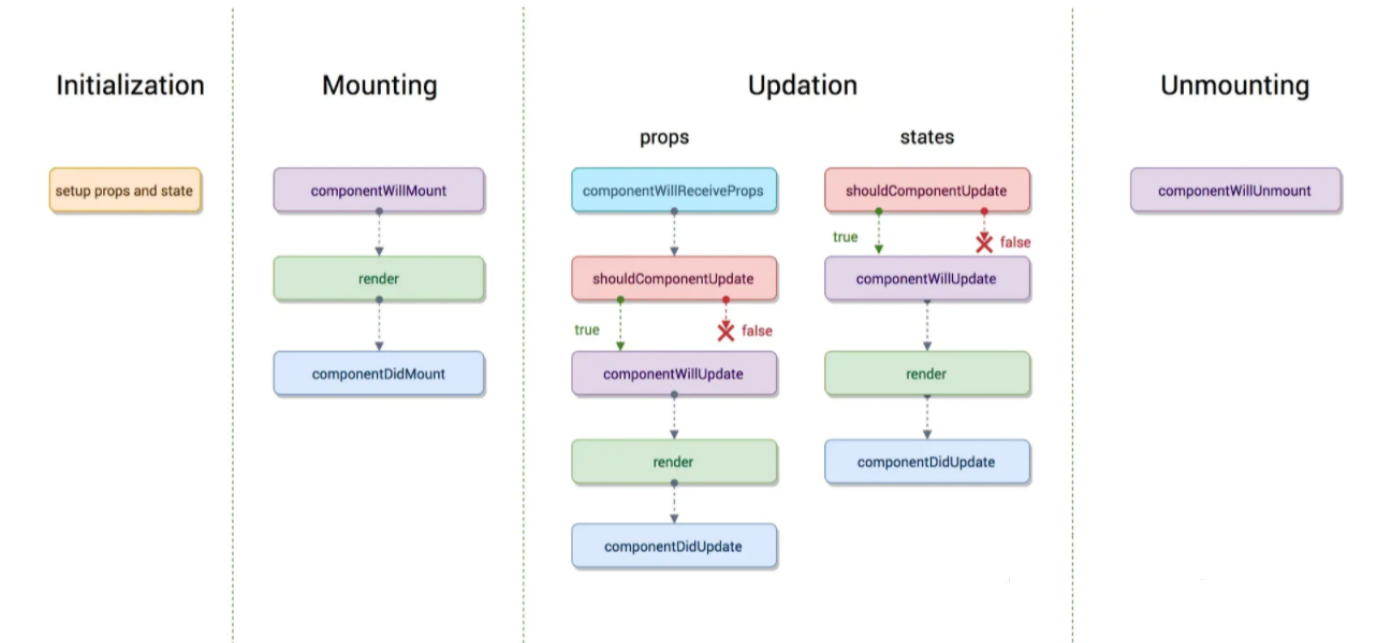
一旦一个组件实例被卸载，其不会被再次挂载，而只可能是被重新创建

## 三、总结

新版生命周期整体流程如下图所示：



旧的生命周期流程图如下：



通过两个图的对比，可以发现新版的生命周期减少了以下三种方法：

- componentWillMount
- componentWillReceiveProps
- componentWillUpdate

其实这三个方法仍然存在，只是在前者加上了UNSAFE\_前缀，如UNSAFE\_componentWillMount，并不像字面意思那样表示不安全，而是表示这些生命周期的代码可能在未来的 react 版本可能废除

同时也新增了两个生命周期函数：

- getDerivedStateFromProps

- `getSnapshotBeforeUpdate`

## 参考文献

- <https://github.com/pomelovico/keep/issues/23>
- <https://segmentfault.com/a/1190000020268993>