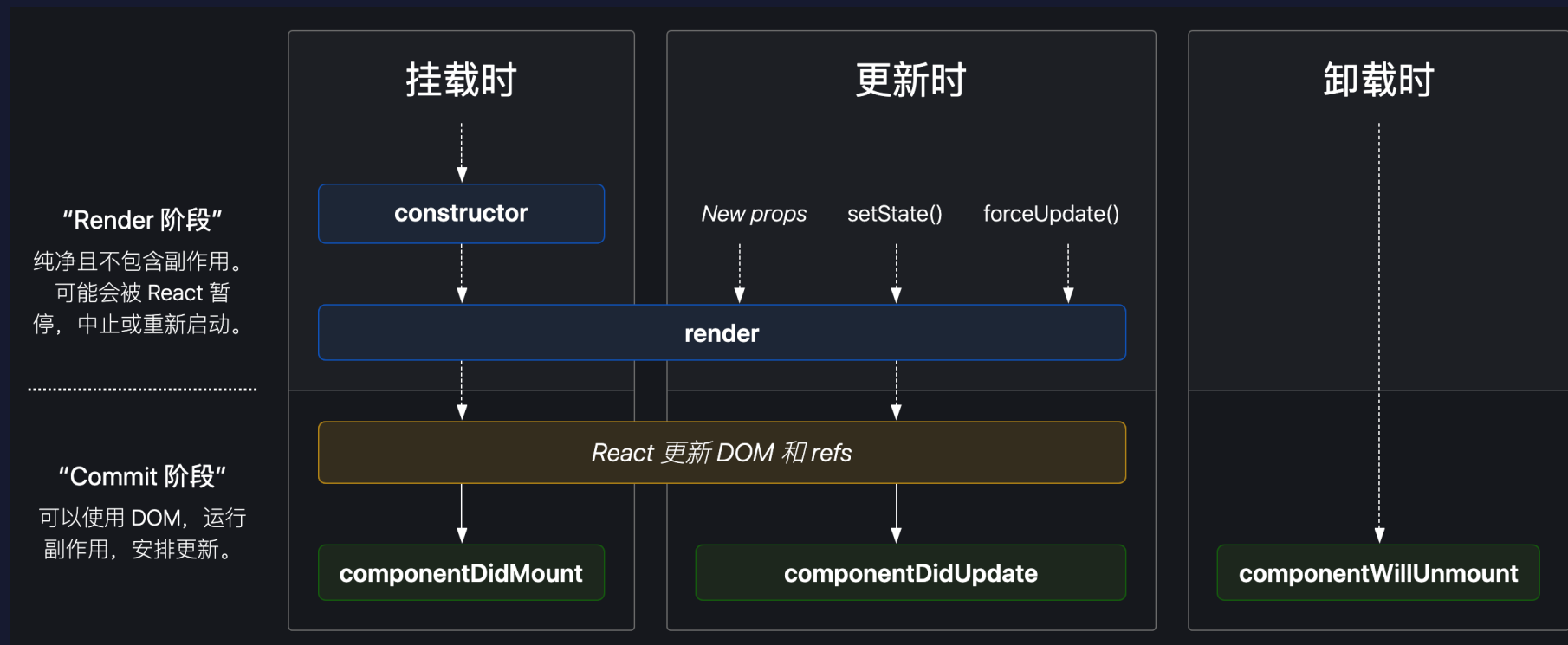


# 生命周期方法

React 元件在網頁上的各個執行階段



# 生命周期方法

## 使用注意事項

- > 生命周期方法是React設計給”**類別型元件**”使用的，不是函式型元件，函式型元件可以用useEffect勾子來模擬，但要注意行為不是100%相同
- > `componentDidMount` 代表元件”**已經**”出現在網頁上，這個方法中可以使用直接DOM處理，或向伺服器要初始化資料的JS程式碼
- > `componentDidUpdate` 代表元件”**已經**”更新完成(真實DOM)，這個方法中可以得到最後更新的狀態值 (只有State更新，或接收到新的props)
- > `componentWillUnmount` 代表元件”**即將**”被移除到DOM外(ps. 不是隱藏)

constructor, render也是生命周期方法，除了這五個外，其它生命周期方法都是最佳化/除錯用進階使用

# useEffect勾子

有兩個傳入參數：

- Callback函式
- 相依變數(狀態/屬性)陣列

```
1 // 模擬 componentDidMount
2 useEffect(() => {
3   // 程式碼寫在這
4 }, [])
5
6 // 模擬 componentDidUpdate
7 // componentDidMount時也會執行一次
8 useEffect(() => {
9   // 程式碼寫在這
10 }, [total])
11
12 // 模擬 componentWillUnmount
13 useEffect(() => {
14   return () => {
15     // 程式碼寫在這
16   }
17 }, [total])
```

完全沒寫代表每次  
render都會執行，寫  
為空白陣列為元件初  
次render時執行一次