

# Data Fetching & Other Side Effects

# AGENDA

Pure functions

Side effects

The useEffect hook

Rules of hooks

Examples of the useEffect hook


**ONE DOES NOT SIMPLY**

**MUTATE STATE**

# Pure Functions

slido

What is a pure function? Select all that are true.

 Start presenting to display the poll results on this slide.

slido

# Examples of side effects



 Start presenting to display the poll results on this slide.

# Side-Effects

A **secondary effect** (an interaction with the outside of the function) when running a function:

- Setting timers or intervals
- Modifying DOM elements not controlled by React
- Network request
- Connection to a socket server
- Adding and removing event listeners
- Logging to the console

# Side-Effects

 Side-effects are generally undesirable because they can introduce a **\*\*lot of bugs\*\*** 

In React, they can disrupt the normal component rendering 



# The useEffect Hook

To better handle side-effects, React encapsulates them in a `useEffect` Hook.

🔥 It's important to understand **when** the `useEffect` happens

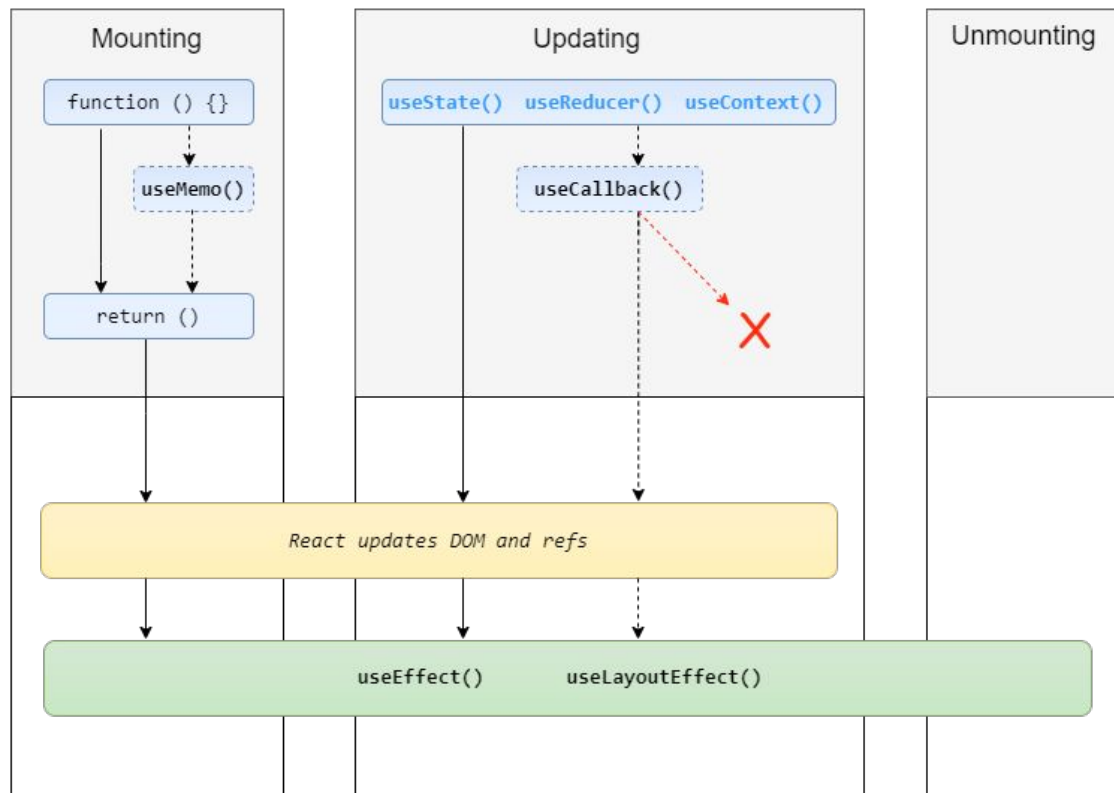
Let's look at the hooks lifecycle 👁👁



# React Hooks Lifecycle

## "Render phase"




Pure and has no side effects. May be paused, aborted or restarted by React



## "Commit phase"

Can work with DOM, run side effects, schedule updates.

# The useEffect has 3 potential actions

1. adding a side effect 
2. reinvoking the side effect (or not) 
3. cleaning up the side effect 

# Recap of The Rules of Hooks

- 1 Don't call Hooks inside loops, conditions, or nested functions.
- 2 Only call Hooks from the top-level of a function component or a custom Hook.
- 3 The effect method that we pass to `useEffect` must either return `undefined` or a `function`.

The easiest way to avoid issues with this rule is always to declare your effect as a **multiline function**.

# useEffect Syntax

```
useEffect(() => {  
    // execute side effect here  
    return () => {  
        // clean up any side effect that was added (ex. removing event  
listeners)  
    };  
}, [] // dependency array);
```

# Examples

# Questions?