React useEffect Hook - Complete Guide

Complete Guide to useEffect in React

What is useEffect?

useEffect is a Hook in React that lets you perform side effects in function components.

Side effects include: fetching data, timers, subscriptions, DOM manipulations, etc.

In class components: componentDidMount, componentDidUpdate, componentWillUnmount.

In function components: all in one place via useEffect.

Basic Syntax:

```
React.useEffect(() => {
    // code after render
}, [dependencies])
```

Behavior of useEffect:

1. No Dependency Array:

```
useEffect(() => {
    console.log("Runs after every render")
})
```

- Runs after every render (initial + updates)

2. Empty Dependency Array []:

```
useEffect(() => {
    console.log("Runs once on mount")
}, [])
```

- Runs only once (on first render)

3. With Dependencies [var1, var2]:

```
useEffect(() => {
    console.log("Runs when 'count' changes")
}, [count])
```

- Runs after first render and when 'count' changes

Example:

```
const [count, setCount] = useState(0)
const [counter, setCounter] = useState(0)
useEffect(() => {
    console.log("Effect runs when counter changes")
}, [counter])
```

Dependency Array Content:

Can include:

- useState variables
- Props
- Context values

Avoid expressions like count + 1

Cleanup Function:

```
useEffect(() => {
  const id = setInterval(() => {
    console.log("Interval running")
  }, 1000)
  return () => {
    clearInterval(id)
    console.log("Interval cleared")
  }
}, [])
```

Summary Table:

React useEffect Hook - Complete Guide

Dependency Array	Effect runs when...
Not provided	After every render
[]	After first render only
[count]	After first render and when count changes
[counter]	After first render and when counter changes
[count, counter]	After first render and when count or counter changes

Final Recap:

"useEffect runs automatically after render. You control when it runs using the dependency array. It helps trigger code at specific lifecycle stages."