@REACTJS_WITH_AJITH

REACT

USE EFFECT

HOOK

ABOUT THIS POST

we may need to modify the UI with new data/details, based on API results, User actions, DOM updates, etc.., All these actions that can potentially be used to change the component data/state are called Side effects.

And useEffect helps to handles side effect

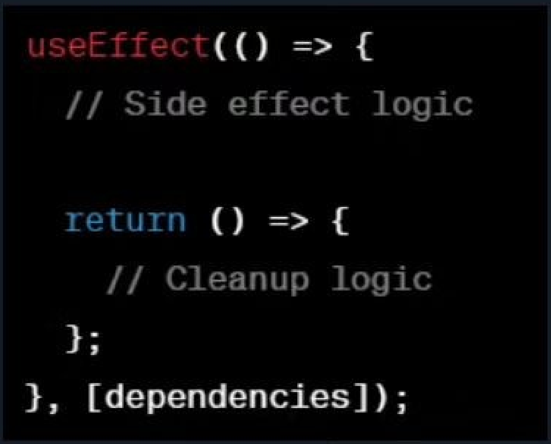
We have multiple ways of using useEffect in functional components. Let's see them today :)

USE EFFECT SYNTAX

useEffect hook in React takes a function as its first argument.

1st argument - This function can perform those discussed side effects. This function can also have an optional return function to write cleanup code

2nd argument - useEffect holds an optional array of dependencies as second argument



```
useEffect(() => {
  // Side effect logic

  return () => {
    // Cleanup logic
  };
}, [dependencies]);
```

@REACTJS_WITH_AJITH

RENDER & RERENDER

Before learning the behavior of useEffect, we ll see what is
1) render 2) re-render

Rendering - Whenever a code runs the first time, the component needs to render to show the UI with initial states and props

Re-rendering - Whenever there is an update, i.e., a change in state or prop, the component needs to re-render to show the UI with updated results

@REACTJS_WITH_AJITH

BEHAVIOR OF USEEFFECT

The useEffect exhibits 4 different behaviors based on its usage

1) useEffect(func)
2) useEffect(function, [])
3) useEffect(function , [dependencies])
4) useEffect(function with return function)

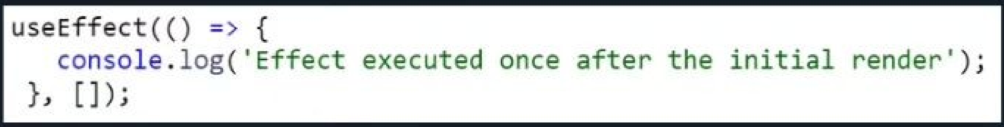
@REACTJS_WITH_AJITH

USEEFFECT(FUNC)

HOW	UseEffect takes just function as 1st argument. This function will execute on every render and re-render.
EXAMPLE	`useEffect(() => {` `console.log('Effect executed after every render');` `});`
WHY	Best place to console log to see the latest value on every render & rerender

@REACTJS_WITH_AJITH

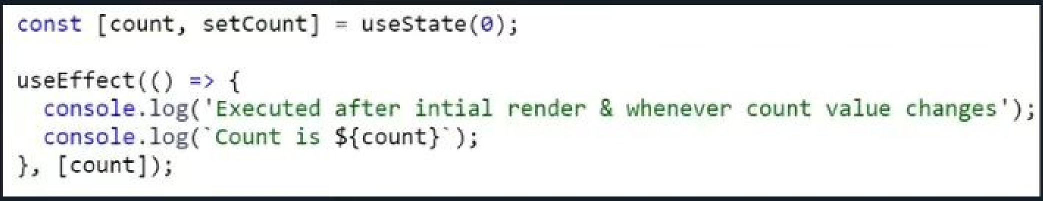
USEEFFECT(FUNCTION, [ ])

HOW	When an empty dependency is passed, The function executes only once during the initial render. (Does not execute on re-renders)
EXAMPLE	 `useEffect(() => {` `console.log('Effect executed once after the initial render');` `}, []);`
WHY	This is highly helpful as we might need to update or set data to a state only once when the component renders the first time.

@REACTJS_WITH_AJITH

USEEFFECT(FUNCTION , [DEPENDENCIES])

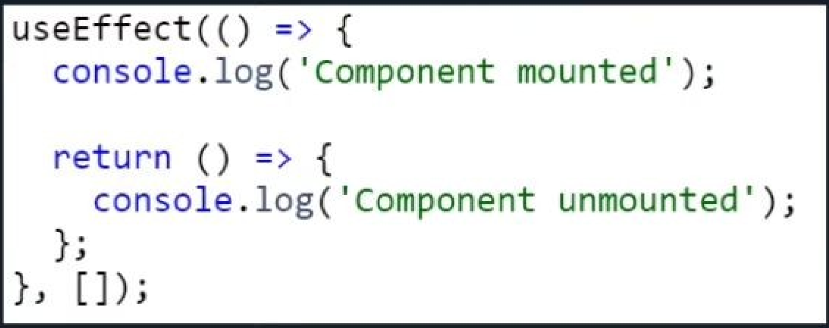
HOW	When a dependency is passed, The function executes once during the initial render and gets executed when witnesses a change in a dependent value

EXAMPLE	 `const [count, setCount] = useState(0);`
	
	`useEffect(() => {`
	`  console.log('Executed after intial render & whenever count value changes');`
	` console.log(`` `Count is ${count}` ``);`
	`}, [count]);`

WHY	This is highly helpful when we need to update the UI based on states or prop changes. Keep the prop or state in dependency. Write the update code in the function.

@REACTJS_WITH_AJITH

USEEFFECT(FUNC WITH RETURN FUNC)

HOW	Return function is added for clean-up purposes like removing event listeners, cancel subscriptions, etc., It executes at unmounting (when removing the components from UI)
EXAMPLE	
WHY	It is helpful when we need to do something like a DOM update, remove the subscription, etc., before removing the component from UI.

```
useEffect(() => {
  console.log('Component mounted');

  return () => {
    console.log('Component unmounted');
  };
}, []);
```

@REACTJS_WITH_AJITH