

The Effect Hook

What is the `useEffect` hook and how is it used to save states effect.

WE'LL COVER THE FOLLOWING ^

- Need for the Effect Hook
- Calling `useEffect`

Need for the Effect Hook

With class components, you've likely performed side effects such as logging, fetching data, or managing subscriptions.

These side effects may be called "effects" for short, and the effect hook, `useEffect` was created for this purpose.

How is it used?

Well, the `useEffect` hook is called by passing it a function within which you can perform your side effects.

Below is a quick example:

```
useEffect(() => {  
  // 🐛 you can perform side effects here  
  console.log("useEffect first timer here.")  
})
```



Calling `useEffect`

In `useEffect`, I've passed an anonymous function with some side effect called within it.

The next logical question is, when is the `useEffect` function invoked?

Well, remember that in class components you had lifecycle methods such as

Well, remember that in class components you had lifecycle methods such as `componentDidMount` and `componentDidUpdate`.

Since functional components don't have these lifecycle methods, `useEffect` *kind of* takes their place.

In the example above, the function within `useEffect`, also known as the effect function, will be invoked when the functional component mounts (`componentDidMount`) and when the component updates (`componentDidUpdate`).

Here's that in action.

By adding the `useEffect` call above to the counter app, we indeed get the log from the `useEffect` function.



```
function CounterHooks() {  
  const [count, setCount] = useState(0);  
  const [time, setTime] = useState(new Date())  
  // 🐞 look here.  
}
```



```
useEffect(() => {
  console.log("useEffect first timer here.")
}, [count])

const handleClick = () => {
  setCount(count + 1);
  setTime(new Date())
}
return (
  ...
);
}
```

By default, the `useEffect` function will be called after every render.

The `useEffect` hook isn't entirely the same as `componentDidMount` + `componentDidUpdate`. It can be viewed as such, but the implementation differs with some subtle differences.

In the next lesson, we'll discuss how the `useEffect` hook can be used for specific functionality, especially for array passing.