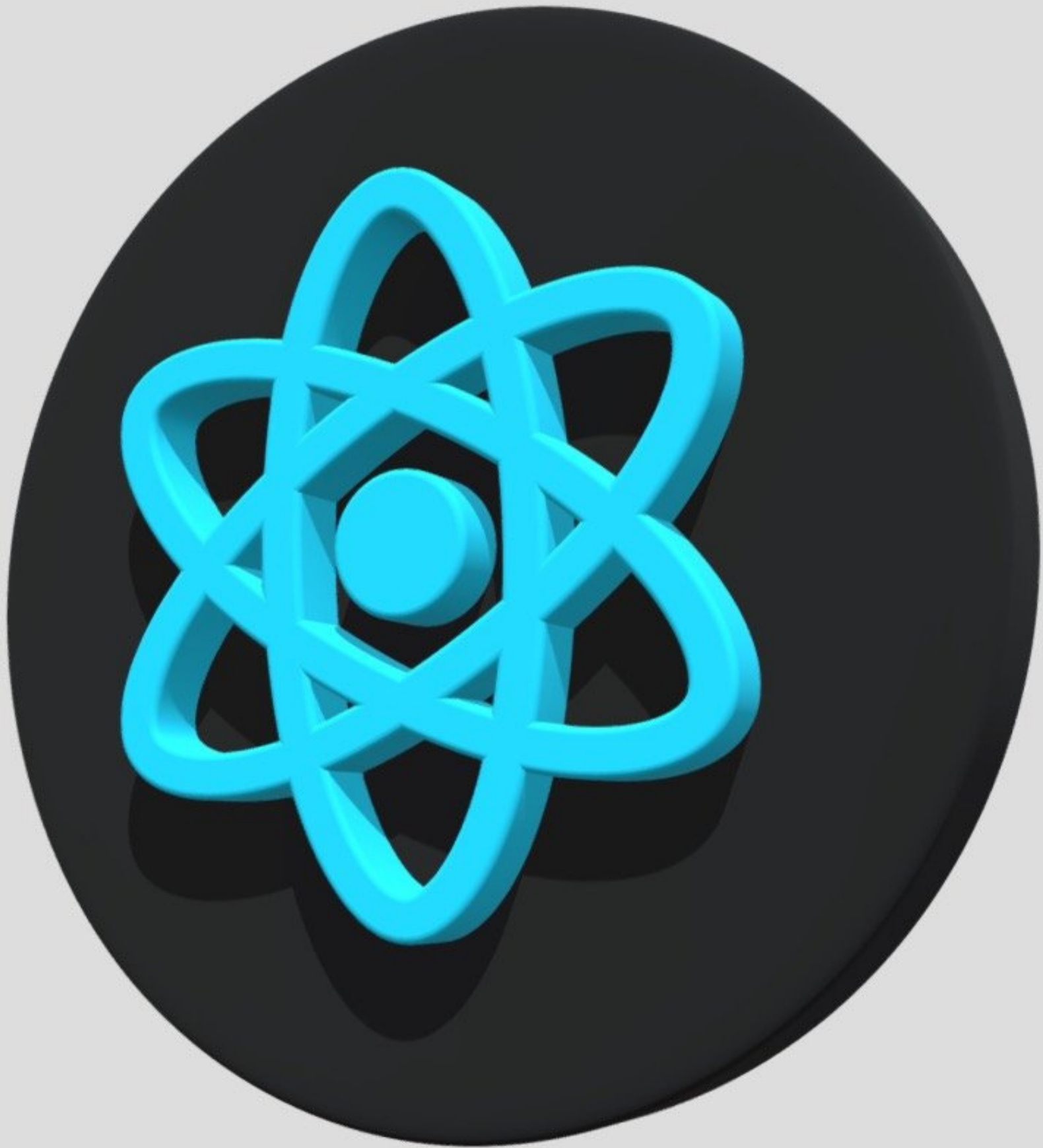


# The useEffect Hook



- **"useEffect"** is a hook that allows you to perform side effects in your functional components.
- **Side effects** are anything that affects something outside of the component, such as **fetching data, setting a timer, or updating the DOM.**

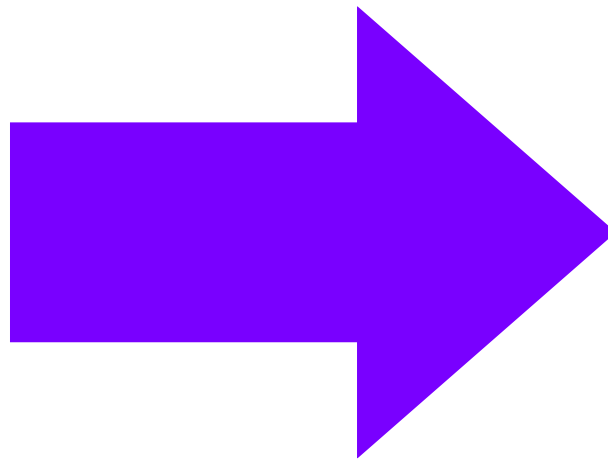
See the example



```
1 import React, { useState, useEffect } from 'react';
2 import axios from 'axios';
3
4 function MyComponent() {
5   const [data, setData] = useState(null);
6
7   useEffect(() => {
8     async function fetchData() {
9       const response = await
10  axios.get('https://api.example.com/data');
11     }
12     fetchData();
13   }, []);
14
15   return (
16     <div>
17       {data ? ( <ul>
18         {data.map((item) => (
19           <li key={item.id}>{item.name}</li>
20         ))}
21       </ul> ) : (
22         <p>Loading...</p>
23       )}
24     </div>
25   );
26 }
```

- Here, we use **useEffect** to fetch data from an API when the component mounts.
- We pass an **empty array ([])** as the second argument to **useEffect**, which means the function will only run once when the component mounts.
- **This ensures that we don't make multiple API calls unnecessarily.**
- When the API call is complete, we render the data in a list.

**USE EFFECT**



**IN DETAIL**



```
useEffect (()⇒{},[]);
```

There are 2 items :

1. `()⇒{}` - is a callback function
2. `[]` - is a dependency array

//use of useEffect in react

1. For every re-render

```
useEffect(()⇒{  
  console.log("As there is no dependency array, this will  
    run every time the component re-renders")  
});
```

2. For onMount - runs only one time

```
useEffect(()⇒{  
  console.log("As there is an empty dependency array, this will  
    run eonce only when the component gets mounted")  
},[]);
```



//use of useEffect in react

### 3. Condition based

```
useEffect(() => {  
  console.log("As there is a condition in dependency array,  
this will run every time the condition changes")  
},[condition])
```

### 4. Condition based with "clean-up"

```
useEffect(() => {  
  console.log("As there is a condition in dependency array,  
this will run every time the condition changes")  
  
  return ()=>{  
    console.log("Use this return as a clean-up tool, this  
runs before the actual code")  
  }  
},[condition]);
```

**Thankyou**

**Follow Ashraya KK**

**<https://www.linkedin.com/in/ashrayakk/>**