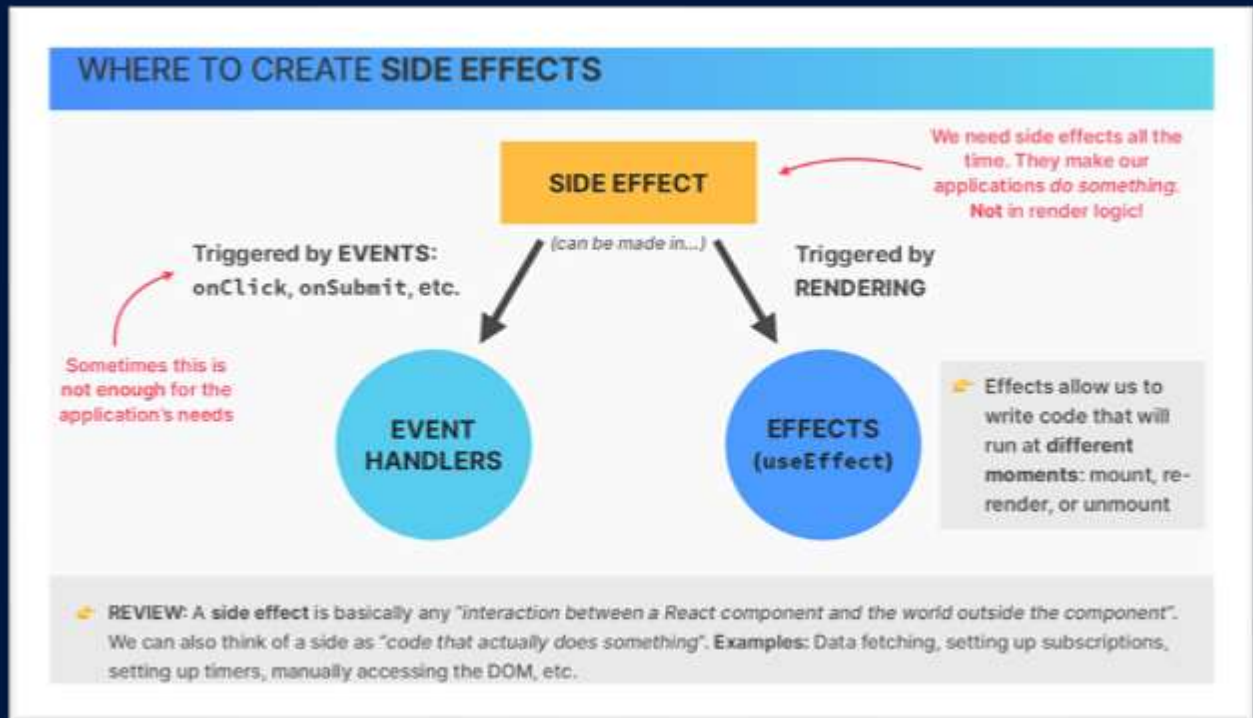




# EFFECTS





## EVENT HANDLERS VS. EFFECTS

### EVENT HANDLERS

```
function handleClick() {  
  fetch('http://www.omdbapi.com/?s=inception')  
    .then((res) => res.json())  
    .then((data) => setMovies(data.Search));  
}
```

- 👉 Executed when the **corresponding event** happens
- 👉 Used to **react** to an event

👉 **Preferred way of creating side effects!**

Produce the same result,  
but at different moments

### EFFECTS (useEffect)

```
useEffect(function () {  
  fetch('http://www.omdbapi.com/?s=inception')  
    .then((res) => res.json())  
    .then((data) => setMovies(data.Search));  
  return () => console.log('cleanup');  
}, []);
```

Effect

Cleanup  
function

Dependency array

🤔 When?

Thinking about  
synchronization,  
not lifecycles

- 👉 Executed **after the component mounts** (initial render), and **after subsequent re-renders** (according to dependency array)
- 👉 Used to keep a component **synchronized with some external system** (in this example, with the API movie data)

(We'll come back to all this after using `useEffect` in practice...)





```
import { useEffect, useState } from "react"

const KEY = '44397289';
const BASE = 'https://www.omdbapi.com'

const App = () =>
{
  const [movies, setMovies] = useState([]);

  useEffect(() =>
  {
    fetch(`${BASE}/?apikey=${KEY}&s=interstellar`)
      .then((res) => res.json())
      .then((data) => setMovies(data.Search));
  }, []);

  return(<div>Movies</div>);
}

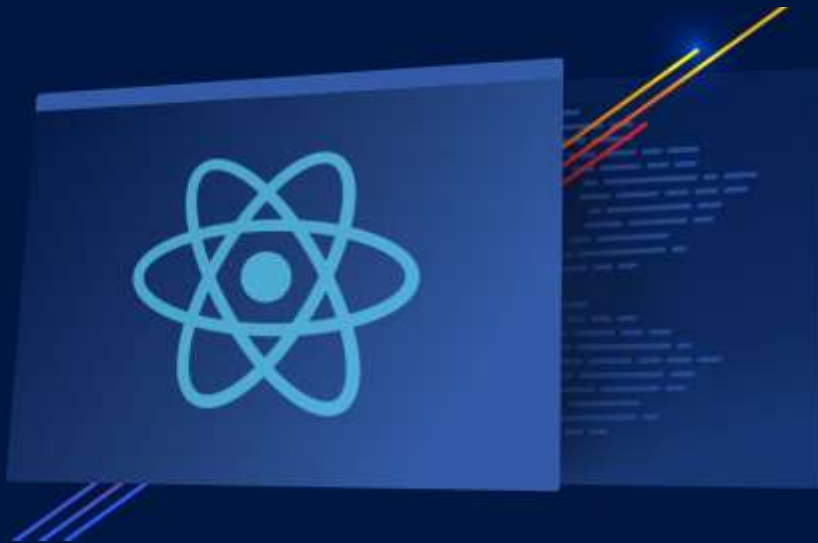
export default App;
```





# Do you find it helpful?

**Let me know down in the comments.**



**Click To Follow For More On LinkedIn**

