

## \* useEffect Hook

The useEffect Hook allows you to manage side effects in your components.

A side effect is a change that affects something outside the component being rendered.

Ex - Fetching data, directly updating the DOM, and timers.

### Syntax-1

```
useEffect(() => {  
  // ...  
});
```

Every render

### Syntax-2

```
useEffect(() => {  
  // ...  
}, []);
```

First render

### Syntax-3

```
useEffect(() => {  
  // ...  
}, [  
  // Dependency  
]);
```

First render  
+  
when dependency changes

## Syntax - 4

useEffect ( () => {

Ⓐ [ // some code

Ⓑ [ return () => {  
 }  
});

To handle unmounting  
of a component

Here Ⓑ will execute before Ⓐ