

# </talentlabs>

## useEffect Hook

### What is useEffect Hook?

useEffect Hook is a function that's being triggered every time the component is being updated or re-rendered. You can use it for doing some tasks before the component is being loaded, updated or refreshed.

### Simplest use

In the example below, you can see a simplest usage of the useEffect hook. It simply means that everytime the component is reloaded, the page title would get updated to "You clicked X time".

```
useEffect(() => {  
  // Update the document title using the browser API  
  document.title = `You clicked ${count} times`;  
});
```

### Only refresh when a particular variable is changed

In this example below, you can see that we have added `[time]` as the second parameter in the useEffect hook. That means that this useEffect hook would only run when the component is updated/refreshed due to an update in the `time` variable. This also applies to the first load.

```
useEffect(() => {  
  setInterval(showDate, 1000)  
}, [time])
```

# </talentlabs>

## Cleaning up

Sometimes, the action that you do in the `useEffect` hook would be running for a long time or even running forever (e.g. refreshing the clock every second by using `setInterval`). In this case, you would want to make sure that React would clean up the “long running task” when it refreshed the component. Otherwise, you would be setting tons of “long running tasks” in the background and use up your whole computer’s memory.

To do this, you will need to return a “clean up function” from the `useEffect` hook. And everytime when the component is refreshed, React will automatically use that function to clean up the long running task. You should always provide a clean up function if you have included a long running task in `useEffect` hook.

```
useEffect(() => {  
  const interval = setInterval(showDate, 1000)  
  
  return () => {  
    clearInterval(interval)  
  }  
}, [time])
```

In the example above, you can see that we have returned an arrow function which called “`clearInterval`” to do the clean up for us.

If you are not 100% clear about this, don’t worry too much about it. We seldom use it in actual coding.