React useEffect Hooks

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

Some examples of side effects are: fetching data, directly updating the DOM, and timers.

useEffect accepts two arguments. The second argument is optional.

useEffect(<function>, <dependency>)

1. No dependency passed: it works componentDidMount

useEffect(() => {

//Runs on every render

});

2. An empty array: it works componentDidMount

useEffect(() => {

//Runs only on the first render

}, []);

3. Props or state values: works componentDidUpdate

useEffect(() => {

//Runs on the first render

//And any time any dependency value changes

}, [prop, state]);



Effect Cleanup

Some effects require cleanup to reduce memory leaks.

Timeouts, subscriptions, event listeners, and other effects that are no longer needed should be disposed.

We do this by including a return function at the end of the useEffect Hook.

