useEffect Basics

useEffect hook allows you to perform side effects in your components. Some examples of side effects are : fetching data, directly updating the DOM etc.

```
useEffect()=>{

   //do your magic here

}
```

things to keep in mind

1.WITHOUT THE ARRAY

here you can se we are not providing any array that array is called the dependency array

so anytime we dont specify the dependency array so in that situation the call back function will call on every single render

2. useEffect with conditional

-> you cannot prep your hook inside a conditional statement

eg.❌❌❌

```
 if(val>0){

   useEffect(()=>{

      clg("call useEffect)

      document.title = 'Increment (${val})'

   })

 }
```

-> if you want to use conditional statement you are going to have to use inside the useEffect hook

eg.✅✅✅

```
useEffect(()=>{
   if(val>0){
      clg("call useEffect)
      document.title = 'Increment (${val})'
   }
})
```

3. Any time we specify empty dependency array forward useEffect ot will only fire that function which is available inside our useEffect hook on the initial render like when our component first renders it will call their functions not when our component changes

eg.

```
useEffect(()=>{
   clg("call useEffect")
   if(val>0){
      document.title = "Increment(${val})'
   }
},[])
```

but if we create some sort of a state and we provide the value of that state so anytime that state changes that component will re render and that callback function will fire

Code:-

```
import React, { useEffect, useState } from 'react'

const App = () => {
 const [value,setValue] = useState(0)
 const [something,setSomething] = useState(0)
```

```
  useEffect(()=>{      // first argument it takes is a call back function and second argumnet is the dependancy array

    console.log("call useEffect")

    document.title = `Increment ${value}`

  },[value])  //if we do not give the dependency array, then it will run this code every single time whenever our component re renders

  //now you will think if we add value to dependacy array and do not making a dependency array gives the same result then we are wrong

  //=> when we secify value inside the dependency array and anytime when our component changes becaues of this state the call back function will gonna be fire

  // => but if we have a sort of some other state and that state is also rerendering then the call back function code will not gonna be fire

  return (

    <div>

      <h2>{value}</h2>

      <button onClick={()=>setValue(value+1)}>Add</button>

      <button onClick={()=>setSomething(something+1)}>Something add</button>

    </div>

  )

}


export default App
```