Simple Counter using useEffect Hook

we will be creating a simple counter app with help of useEffect Hook.

The useEffect hook allow us to perform side effects to our component. UseEffect runs on every render. So in our project we want to render our component change every time count changes.

UseEffect has two arguments. A function to be execute and second parameter is optional. It is dependencies to useEffect function.

So lets start creating a simple counter using useEffect Hook.

So first we will create a state variable using useState. Now this will be count with initial value 0.

```
const [count, setCount] = useState(0);
```

Now we will also create another variable which will allow us to understand function and its dependency in useEffect hook.

```
const [name, setName] = useState('Rahul');
```

Now we will use useEffect hook.
It is important to note by using this Hook, we tell React that our component needs to do something after render. React will remember the function we passed and call it later after performing the DOM updates.

So want to show both state variables as title of our browser tab.
We can also pass more than one function.
So we will create a if statement to give us alert when our count reaches at multiple of 5.

```
useEffect(() => {
document.title = ` ${name} ${count} `;
if (count % 5 === 0) {
alert('Number of clicks: ' + count);
}
else {
}
};
```

Now we will pass dependency as our second argument of our useEffect hook.
If none of dependency is declared, this will run the function everytime component get rendered.
If we pass empty array, it will run function only first time when browser loads.
We want function of our useEffect hook to execute only when there is change in state variable count.

So we will have

```
useEffect(() => {
document.title = ` ${name} ${count} `;
if (count % 5 === 0) {
alert('Number of clicks: ' + count);
}
else {
}
}, [count]);
```

Now we will create an UI such a way, it has buttons for increment and decrement along with a text input. We will display our text input. So our text imput will be stored in state variable name.

```
<div className="container">
<input value={name} onChange={(e) => setName(e.target.value)} />
<div className="number">
<p>{name}!</p>
<p>Count: {count}</p>
<div className="allbuttons">
<button class="button" onClick={() => setCount(count + 1)}>INCR</button>
<button class="button" onClick={() => setCount(count - 1)}>DECR</button>
</div>
</div>
</div>
</div>
```

Now if we can see name and count in Browser title. Now if we change value of our count, we can see our useEffect hook executes its function. But if we change text, there is no change in browser title untill we change the count variable. This is how we control the rendering of our component with useEffect hook.