# React HooksuseState useEffect





HOOKS: A hook is a special function that lets you "hook into" React features.

It allows you to use features of class based components in function based components.

#### For example:

useState is a hook that lets you add React state to function based components.

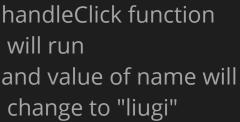


### **REACT** functional component:

```
const Home = ( ) => {
```

```
let name = "mario"; //Understanding the problem
```

REACT doesn't watch for changes, when value does change it doesn't trigger react to re-render the template with new value.







## Solution: using useState hook

```
import {useState} from 'react';
const Home = ( ) => {
  const [name, setName] = useState("mario");
                useState function
                returns an array
                with 2 elements
  const handleClick = ( ) => {
    setName( "luigi" );
  return (
   <div>
      <h1>{name}</h1>
      <but><br/><br/>dick = { handleClick } > Click me</br>
      </button>
   </div>
   );
```

## useEffect hook

useEffect hook lets you perform side effects in function components.

example of side effects are - data fetching, manually changing dom in react component ...

useEffect runs a function at every render of the component.





```
const Home = ( ) => {
  const [name, setName] = useState("mario");
  useEffect ( ( ) => {
     console.log("useEffect ran");
  }, [name]);
         useEffect will run function if value of name changes
  return and after first render
   <div>
      <h1>{name}</h1>
      <button onClick = { ( ) => setName('luigi') } > Change name
      </button>
   </div>
   );
```





For more such content follow me

