

# Events in React

## Handling Click Events

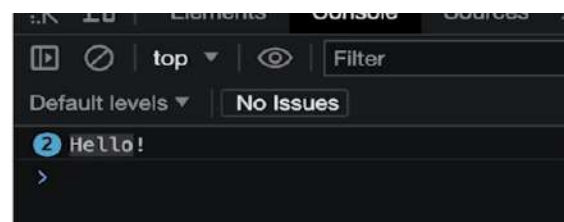
```
function doSomething() {  
  console.log("button was clicked");  
}  
  
export default function Button() {  
  return (  
    <div>  
      <button onClick={doSomething}>Click me!</button>  
    </div>  
  );  
}
```

Button.jsx • App.jsx

src > Button.jsx > Button

```
1  function printHello() {
2    | console.log("Hello!");
3  }
4
5  export default function Button() {
6    | return (
7    |   <div>
8    |   | <button onClick={printHello}>Click me!</button>
9    |   </div>
10   | );
11   }
12
```

Click me!



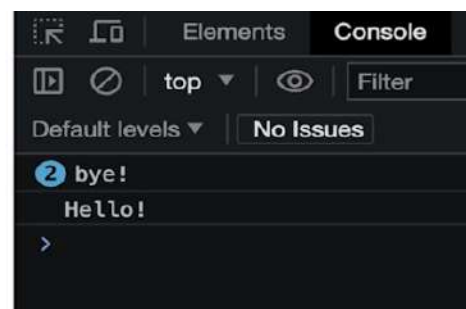
Button.jsx × App.jsx

src > Button.jsx > Button

```
1  function printHello() {
2    |   console.log("Hello!");
3  }
4
5  function printBye() {
6    |   console.log("bye!");
7  }
8
9  export default function Button() {
10   |   return (
11   |     <div>
12   |       <button onClick={printHello}>Click me!</button>
13   |       <p onClick={printBye}>this parah is for event demo</p>
14   |     </div>
15   |   );
16  }
17
```

Click me!

this parah is for event demo



# Events in React

---

## Handling Non-Click Events

```
function handleHover() {  
  console.log("hover");  
}  
  
export default function Button() {  
  return (  
    <div>  
      <button onMouseOver={handleHover}>Hover Over me!</button>  
    </div>  
  );  
}
```

```
function printBye() {  
  console.log("bye!");  
}
```

```
export default function Button() {  
  return (  
    <div>
```

```
      <button onClick={printHello}>Click me!</button>
```

```
      <p onMouseOver={printBye}>
```

Lorem ipsum dolor sit amet consectetur adipisicing elit. Dolorum soluta  
 alias facilis eaque mollitia, distinctio corporis vel exercitationem  
 itaque, quidem dignissimos odit tenetur, tempora cupiditate  
 necessitatibus. Adipisci praesentium delectus nesciunt.

```
      </p>
```

```
    </div>
```

```
  );  
}
```



Click me!

Lorem ipsum dolor sit amet consectetur adipisicing elit. Dolorum soluta alias facilis eaque mollitia, distinctio corporis vel exercitationem itaque, quidem dignissimos odit tenetur, tempora cupiditate necessitatibus. Adipisci praesentium delectus nesciunt.

top ▾  
Default levels ▾ No  
Hello!  
bye!  
>



src >  Button.jsx >  handle

```
1  function handleClick() {
2    console.log("Hello!");
3  }
4  
5  function handleMouse() {
6    console.log("bye!");
7  }
8
9  export default function Button() {
10    return (
11      <div>
12        <button onClick={handleClick}>Click me!</button>
13        <p onMouseOver={printBye}>
14          Lorem ipsum dolor sit amet consectetur adipisicing elit. Dolor
15          alias facilis eaque mollitia, distinctio corporis vel exercita
16          itaque, quidem dignissimos odit tenetur, tempora cupiditate
17          necessitatibus. Adipisci praesentium delectus nesciunt.
18        </p>
```

## State in React

The state is a built-in React object that is used to contain data or information about the component. A component's state can change over time; whenever it changes, the component re-renders.



leD

1

f

2

3

}

4

5

f

6

7

}

8

9

f

10

1

11

}

12

13

ex

14

15

src > Button.jsx > Button

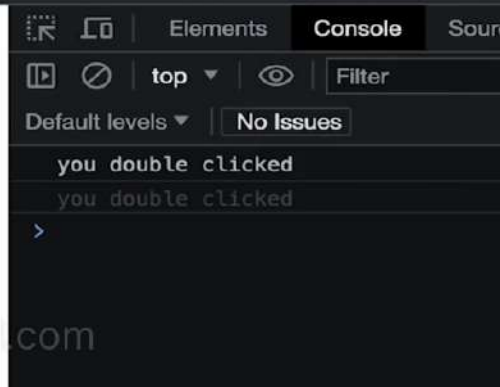
```
14   return (  
15     <div>  
16       <button onClick={handleClick}>Click me!</button>  
17       <p onMouseOver={handleMouseOver}>  
18         Lorem ipsum dolor sit amet consectetur adipisicing elit. Dolorum so  
19         alias facilis eaque mollitia, distinctio corporis vel exercitatione  
20         itaque, quidem dignissimos odit tenetur, tempora cupiditate  
21         necessitatibus. Adipisci praesentium delectus nesciunt.  
22       </p>  
23       <button onClick={handleDblClick()}>double click me!</button>  
24     </div>  
25   );  
26 }  
27
```

Click me!

Lorem ipsum dolor sit amet consectetur adipisicing elit. Dolorum soluta alias facilis eaque mollitia, distinctio corporis vel exercitationem itaque, quidem dignissimos odit tenetur, tempora cupiditate necessitatibus. Adipisci praesentium delectus nesciunt.

double click me!

snehagupta7385@gmail.com



src > Button.jsx > ...

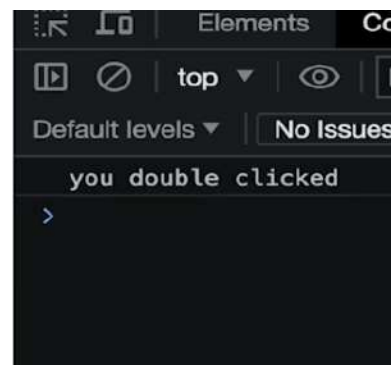
```
14   return (  
15     <div>  
16       <button onClick={handleClick}>Click me!</button>  
17       <p onMouseOver={handleMouseOver}>  
18         Lorem ipsum dolor sit amet consectetur adipisicing elit. Dolorum solu  
19         alias facilis eaque mollitia, distinctio corporis vel exercitationem  
20         itaque, quidem dignissimos odit tenetur, tempora cupiditate  
21         necessitatibus. Adipisci praesentium delectus nesciunt.  
22       </p>  
23       <button onDoubleClick={handleDbClick}>double click me!</button>  
24     </div>  
25   );  
26 }  
27
```

Click me!

Lorem ipsum dolor sit amet consectetur adipisicing elit. Dolorum soluta alias facilis eaque mollitia, distinctio corporis vel exercitationem itaque, quidem dignissimos odit tenetur, tempora cupiditate necessitatibus.

Adipisci praesentium delectus nesciunt.

double click me!



# Event Object

We can access the Event Object in Handler

```
function handleFormSubmit(event) {  
  event.preventDefault();  
  console.log("form was submitted");  
}  
  
export default function Form() {  
  return (  
    <form onSubmit={handleFormSubmit}>  
      <button>Submit</button>  
    </form>  
  );  
}
```



Counter.jsx × App.jsx

src > Counter.jsx > Counter

```
1  export default function Counter() {
2    let count = 0;
3
4    function incCount() {
5      count += 1;
6      console.log(count);
7    }
8
9    return (
10     <div>
11       <h3>Count = {count}</h3>
12       <button onClick={incCount}>Increase Count</button>
13     </div>
14   );
15 }
16
```

Button.jsx • App.jsx

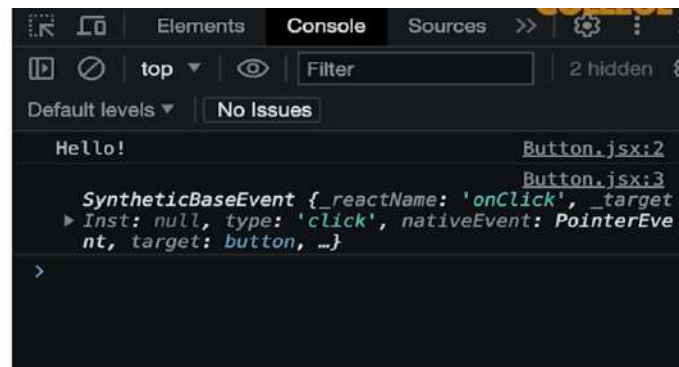
src > Button.jsx > handleClick

```
1  function handleClick(event) {
2    console.log("Hello!");
3    console.log(event);
4  }
5
6  function handleMouseOver() {
7    console.log("bye!");
8  }
9
10 function handleDbClick() {
11   console.log("you double clicked");
12 }
13
14 export default function Button() {
15   return (
16     <div>
17       <button onClick={handleClick}>Click me!</button>
18       <p onMouseOver={handleMouseOver}>
19         Lorem ipsum dolor sit amet consectetur adipisicing
20         alias facilis eaque mollitia, distinctio corp
```

Click me!

Lorem ipsum dolor sit amet consectetur adipisicing elit. Dolorum soluta alias facilis eaque mollitia, distinctio corporis vel exercitationem itaque, quidem dignissimos odit tenetur, tempora cupiditate necessitatibus. Adipisci praesentium delectus nesciunt.

double click me!



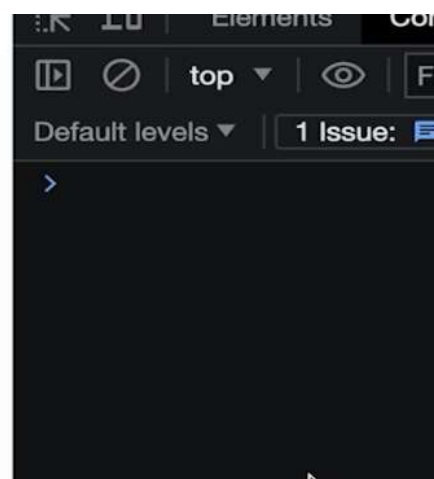
Button.jsx

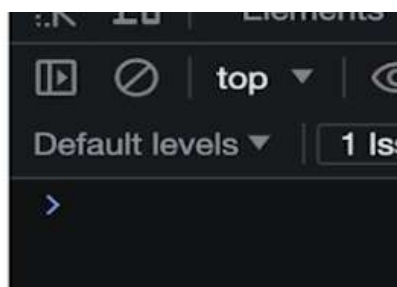
Form.jsx



App.jsx

src > Form.jsx > Form

```
1  function handleFormSubmit() {  
2    |   console.log("form was submitted");  
3  }  
4  
5  export default function Form() {  
6    |   return (  
7      |     <form>  
8      |       |   <input placeholder="write something" />  
9      |       |   <button onClick={handleFormSubmit}>Submit</button>  
10     |     </form>  
11     |   );  
12   }  
13
```





  | top

Default levels ▼

>

Button.jsx

Form.jsx

App.jsx

src > Form.jsx > handleFormSubmit

```
1  function handleFormSubmit(event) {
2    event.preventDefault()
3    console.log("form was submitted");
4  }
5
6  export default function Form() {
7    return (
8      <form>
9        <input placeholder="write something" />
10       <button onClick={handleFormSubmit}>Submit</button>
11     </form>
12   );
13 }
14
```



Elements Console

top Filter

Default levels ▾ 1 Issue: 1

form was submitted

>

Button.jsx

Form.jsx



App.jsx

shelagupta7385@gmail.com

src > Form.jsx > handleFormSubmit

```
1  function handleFormSubmit(event) {
2    event.preventDefault();
3    console.log("Form was submitted");
4  }
5
6  export default function Form() {
7    return (
8      <form onSubmit={handleFormSubmit}>
9        <input placeholder="write something" />
10       <button>Submit</button>
11     </form>
12   );
13 }
14
```

## State in React

The state is a built-in React object that is used to contain data or information about the component. A component's state can change over time; whenever it changes, the component re-renders.

App.jsx U Counter.jsx U X

Learning > FRONTEND > Apna College > React > React Lecture-3 > React-State-Lecture > src >

Complexity is 6 It's time to do something...

```
1  export default function Counter(){
2      let count=0;
3
4      function increaseCount(){
5          count++;
6          console.log(count);
7      }
8
9      return(
10         <div>
11             <h3>Count: {count}</h3>
12             <button onClick={increaseCount}>Increase Count</button>
13         </div>
14     )
15 }
```

```
1  import './App.css';
2  import Counter from './Counter.jsx';
3
4  Complexity is 4 Everything is cool!
5  function App() {
6      return (
7          <>
8              <h1>States in React</h1>
9              <Counter/>
10             </>
11         );
12     }
13
14     export default App;
15
```

# States in React

Count = 0

Increase Count

top

Default levels

No Iss

1

2

3

4

5

6

7

8

9

>

## State in React

The state is a built-in React object that is used to contain data or information about the component. A component's state can change over time; whenever it changes, the component re-renders.



# Hooks

Hooks were a new addition in React 16.8.

They let you use state and other React features without writing a class.

React  
↓  
class components

↓  
functional components



# Hooks

Hooks were a new addition in React 16.8.

They let you use state and other React features without writing a class.

React  
↓  
class components

↓  
functional components  
+  
Hooks

# Hooks



Hooks were a new addition in React 16.8.

They let you use state and other React features without writing a class.

React



class components



functional components

# Hooks

Hooks were a new addition in React 16.8.

They let you use state and other React features without writing a class.

React



class components



functional components

5  
15

## useState( )

useState is a React Hook that lets you add a state variable to your component.

```
const [state, setState] = useState(initialState);
```

useState returns an array with exactly two values:

1. The current state. During the first render, it will match the `initialState` you have passed.
2. The `set` function that lets you update the state to a different value and trigger a re

## useState( )

useState is a React Hook that lets you add a state variable to your component.

```
const [state, setState] = useState(initialState);
```

*update function*

useState returns an array with exactly two values:

1. The current state. During the first render, it will match the `initialState` you have passed.
2. The `set` function that lets you update the state to a different value and trigger a re

Counter.jsx • App.jsx

src > Counter.jsx > Counter > incCount

```
1  import { useState } from "react";
2
3  export default function Counter() {
4    let [count, setCount] = useState(0);
5
6    let incCount = () => {
7      setCount(count+1);
8      console.log(count);
9    };
10
11    return (
12      <div>
13        <h3>Count = {count}</h3>
14        <button onClick={incCount}>Increase Count</button>
15      </div>
16    );
17  }
18
```

# States in React

Count = 7

Increase Count

Counter.jsx • App.jsx

src > Counter.jsx > Counter > incCount

```
1  import { useState } from "react";
2
3  export default function Counter() {
4    let [count, setCount] = useState(0);
5
6    let incCount = () => {
7      setCount(count + 1);
8      console.log(count);
9    };
10
11    return (
12      <div>
13        <h3>Count = {count}</h3>
14        <button onClick={incCount}>Increase Count</button>
15      </div>
16    );
17  }
18
```

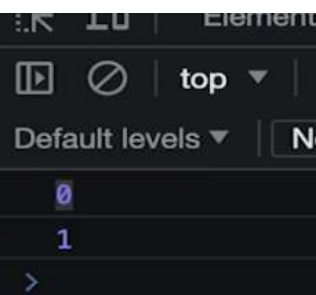


# States in React

Count = 2

Increase Count

snehagupta7385@gmail.com



App.jsx U Counter.jsx U X

Apna College > React > React Lecture-3 > React-State-Lecture > src > Counter.jsx > Co

```
1  import { useState } from "react";  
    Complexity is 6 It's time to do something...  
2  export default function Counter(){  
3      let [count, setCount]=useState(0);  
4  
5      let increaseCount= ()=>{  
6          setCount(count+1);  
7          console.log(count+1);  
8      }  
9      return(  
10         <div>  
11             <h3>Count: {count}</h3>  
12             <button onClick={increaseCount}>Increase Count</button>  
13         </div>  
14     )  
15 }
```

# States in React

Count: 18

Increase Count

```
9      Counter.jsx:7
10     Counter.jsx:7
11     Counter.jsx:7
12     Counter.jsx:7
13     Counter.jsx:7
14     Counter.jsx:7
15     Counter.jsx:7
16     Counter.jsx:7
17     Counter.jsx:7
18     Counter.jsx:7
>
```

## Create LikeButton

`useState` is a React Hook that lets you add a state variable to your component.

Complexity is 0 it's time to do something.

```
export default function LikeButton(){  
  function Click(){  
    console.log("like")  
  }  
  return(  
    <div>  
      <p onClick={Click}>  
        <i className="fa-regular fa-heart"></i>  
      </p>  
    </div>  
  );  
}
```

States in React



Element

top ▾

No Issues | 2 hidden

11 like

>

```
import { useState } from "react";
```

Complexity is 8 It's time to do something...

```
export default function LikeButton(){  
  let [Islike, setIslike]=useState(false);
```

```
  let toggleLike= ()=>{  
    setIslike(!Islike);  
  }
```

```
  let likestyle={  
    color:"red"  
  };  
  return(  
    <div>
```

```
      <p onClick={toggleLike}>
```

```
        Islike ? <i className="fa-solid fa-heart" style={likestyle}></i> : <i className="fa-regular fa-heart"></i>
```

```
      </p>  
    </div>
```

```
  );  
}
```


# States in React





# States in React



src >  LikeButton.jsx > ...

```
1  import { useState } from "react";
2  let [isLiked, setIsLiked] = useState(false);
3
4  export default function LikeButton() {
5
6      
7      let toggleLike = () => {
8          setIsLiked(!isLiked);
9      };
10
11     let likeStyle = { color: "red" };
12
13     return (
14         <div>
15             <p onClick={toggleLike}>
16                 {isLiked ? (
17                     <i className="fa-solid fa-heart" style={likeStyle}>
18                 ) : (
19                     <i className="fa-regular fa-heart"></i>
20                 )}
```

Default levels **No issues**

- ✖ ▶ Error: Invalid hook call. Hooks can only be called inside of the body of a function component. This usually happens for one of the following reasons:
  1. You might have mismatching versions of React and the renderer (such as React DOM)
  2. You might be breaking the Rules of Hooks
  3. You might have more than one copy of React in the projectSee <https://reactjs.org/link/invalid-hook-call> for more info on how to debug and fix this problem.
  - at Object.throwInvalidHookError (react-dom.development.js:16227:9)
  - at useState (react.development.js:1622:21)
  - at LikeButton.jsx:2:29
- ✖ ▶ [hmr] Failed to reload /src/LikeButton.jsx. This could be due to syntax errors or modules. (see errors above)

>

```
let [isLiked, setIsLiked] = useState(false);
let [clicks, setClicks] = useState(0);

let toggleLike = () => {
  setIsLiked(!isLiked);
  setClicks(clicks + 1);
};

let likeStyle = { color: "red" };

return (
  <div>
    <p>Clicks = {clicks}</p>
    <p onClick={toggleLike}>
      {isLiked ? (
        <i className="fa-solid fa-heart" style={likeStyle}></i>
      ) : (
        <i className="fa-regular fa-heart"></i>
      )}
    </p>
  </div>
)
```

# States in React

Clicks: 5



# Closure

**Closure** : A closure is a feature in JavaScript where an inner function has access to the outer (enclosing) function's variables

```
function outer() {  
  var b = 10;  
  function inner() {  
    var a = 20;  
    console.log(a+b);  
  }  
  return inner;  
}
```

**\*In JS, once a function completes its execution, any variables that were defined inside the function scope cease to exist.**



```
> function outer() {  
    let b = 10;  
    function inner() {  
        let a = 20;  
        console.log(a+b);  
    }  
}
```

```
< undefined
```

```
> function outer() {  
    let b = 10;  
    function inner() {  
        let a = 20;  
        console.log(a+b);  
    }  
    inner();  
}
```

```
< undefined
```

```
> outer();
```

```
30
```

```
< undefined
```

```
>
```

```
> function outer() {  
  let b = 10;  
  function inner() {  
    let a = 20;  
    console.log(a+b);  
  }  
}
```

```
< undefined
```

```
> function outer() {  
  let b = 10;  
  function inner() {  
    let a = 20;  
    console.log(a+b);  
  }  
  inner()  
}
```

```
< undefined
```

```
> outer();
```

```
30 VM3237:5
```

```
< undefined
```

```
> inner();
```

```
✖ ▶ Uncaught ReferenceError: inner is not defined VM3278:1  
  at <anonymous>:1:1
```

```
>
```



```
> function outer() {  
    let b = 10;  
    function inner() {  
        let a = 20;  
        console.log(a+b);  
    }  
    return inner;  
}
```

```
< undefined
```

```
> let inner = outer();
```

```
< undefined
```

```
> inner()
```



```
< undefined
```



```
>
```


# Re-render : How it works?

```
export default function Counter() {
  let [count, setCount] = useState(0);
  console.log("component is re-executed");
  console.log(`count = ${count}`);

  let incCount = () => {
    setCount(count + 1);
    console.log(`new value of count = ${count}`);
  };

  return (
    <div>
      <p>Count = {count}</p>
      <button onClick={incCount}>Increase Count</button>
    </div>
  );
}
```

src >  Counter.jsx >  Counter

```
1  import { useState } from "react";
2
3  export default function Counter() {
4     let [count, setCount] = useState(0); //initialization
5    console.log("Component is rendered!");
6
7    let incCount = () => {
8      setCount(count + 1);
9    };
10
11    return (
12      <div>
13        <h3>Count = {count}</h3>
14        <button onClick={incCount}>Increase Count</button>
15      </div>
16    );
17  }
18
```

States in React

**Count = 2**

Increase Count

Default levels ▾

No Issues

Component is rendered!

2 Component is rendered!

>

src >  Counter.jsx >  Counter >  incCount

```
1  import { useState } from "react";
2
3  export default function Counter() {
4    let [count, setCount] = useState(0); //initialization
5    console.log("Component is rendered!");
6    console.log(`count = ${count}`);
7
8    let incCount = () => {
9      setCount(count + 1);
10     console.log(`inside incCount, count = ${count}`);
11   };
12
13   return (
14     <div>
15       <h3>Count = {count}</h3>
16       <button onClick={incCount}>Increase Count</button>
17     </div>
18   );
19 }
20
```



States in React

**Count = 2**

Increase Count

Default levels ▾ No Issues

```
Component is rendered!  
count = 0  
inside incCount, count = 0  
Component is rendered!  
count = 1  
inside incCount, count = 1  
Component is rendered!  
count = 2  
>
```

src >  Counter.jsx >  Counter

```
1  import { useState } from "react";
2
3  export default function Counter() {
4    ⚡let [count, setCount] = useState(0); //initialization
5    // console.log("Component is rendered!");
6    // console.log(`count = ${count}`);
7
8    let incCount = () => {
9      setCount(count + 1);
10     console.log(`inside incCount, count = ${count}`);
11   };
12
13   return (
14     <div>
15       <h3>Count = {count}</h3>
16       <button onClick={incCount}>Increase Count</button>
17     </div>
18   );
19 }
20
```

States in React

**Count = 4**

Increase Count




top Filter  
Default levels No Issues  
inside incCount, count = 0  
inside incCount, count = 1  
inside incCount, count = 2  
inside incCount, count = 3  
>

snehagupta7385@gmail.com



# **Callback in Updater Function**

**How to change state when it depends on the current value**

src >  Counter.jsx >  Counter >  incCount

```
1  import { useState } from "react";
2
3  export default function Counter() {
4    let [count, setCount] = useState(0); //initialization
5
6    let incCount = () => {
7      setCount(count + 1);
8      setCount(count + 1);
9      setCount(count + 1);
10     setCount(count + 1);
11   };
12
13   return (
14     <div>
15       <h3>Count = {count}</h3>
16       <button onClick={incCount}>Increase Count</button>
17     </div>
18   );
19 }
20
```

States in React

**Count = 5**

Increase Count



top



Default levels

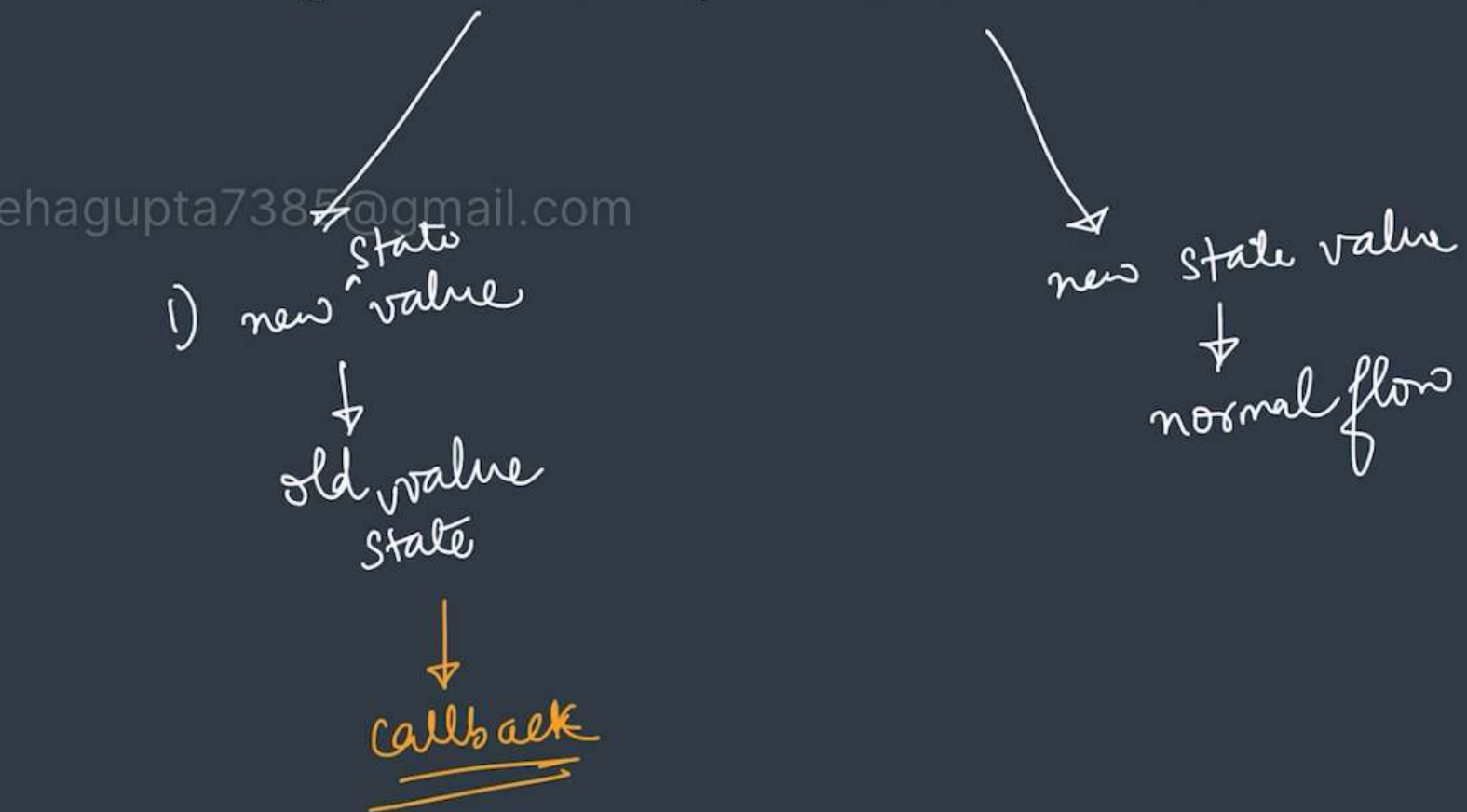


No Issues




# Callback in Updater Function

How to change state when it depends on the current value



src > Counter.jsx > Counter > incCount

```
2
3 export default function Counter() {
4   let [count, setCount] = useState(0); //initialization
5   
6   let incCount = () => {
7     setCount((currCount) => {
8       return currCount + 1;
9     });
10    setCount((currCount) => {
11      return currCount + 1;
12    });
13  };
14
15  return (
16    <div>
17      <h3>Count = {count}</h3>
18      <button onClick={incCount}>Increase Count</button>
19    </div>
20  );
21 }
22
```

States in React

**Count = 4**

**Increase Count**

src >  Counter.jsx >  Counter >  incCount

```
1  import { useState } from "react";
2
3  export default function Counter() {
4    let [count, setCount] = useState(0); //initialization
5
6    let incCount = () => {
7      // setCount((currCount) => {
8      //   return currCount + 1;
9      // });
10     // setCount((currCount) => {
11     //   return currCount + 1;
12     // });
13      setCount(25);
14   };
15
16   return (
17     <div>
18       <h3>Count = {count}</h3>
19       <button onClick={incCount}>Increase Count</button>
20     </div>
21   );
```

## States in React

**Count = 25**

**Increase  Count**



**More about State**

---

src > Counter.jsx > Counter > incCount

```
1  import { useState } from "react";
2
3  export default function Counter() {
4    let [count, setCount] = useState(0); //initialization
5    console.log("component was re-rendered");
6
7    let incCount = () => {
8      // setCount((currCount) => {
9      //   return currCount + 1;
10     // });
11     // setCount((currCount) => {
12     //   return currCount + 1;
13     // });
14     setCount(25);
15   };
16
17   return (
18     <div>
19       <h3>Count = {count}</h3>
20       <button onClick={incCount}>Increase Count</button>
21     </div>
```

States in React

**Count = 25**

Increase Count

component was re-rendered  
2 component was re-rendered  
>

```
src > Counter.jsx > Counter > incCount
1  import { useState } from "react";
2
3  function init() {
4    return Math.random();
5  }
6
7  export default function Counter() {
8    let [count, setCount] = useState(init()); //initialization
9    console.log("component was re-rendered");
10
11    let incCount = () => {
12      setCount((currCount) => {
13        return currCount + 1;
14      });
15    };
16
17    return (
18      <div>
19        <h3>Count = {count}</h3>
20        <button onClick={incCount}>Increase Count</button>
21      </div>
```

States in React

**Count = 1.6408014564605962**

Increase Count

Elements Console

top Filter

Default levels No Issues

component was re-rendered

component was re-rendered

>

src > Counter.jsx > Counter

```
1  import { useState } from "react";
2
3  function init() {
4    console.log("init was executed");
5    return Math.random();
6  }
7
8  export default function Counter() {
9    let [count, setCount] = useState(init()); //initialization
10   console.log("component was rendered");
11
12   let incCount = () => {
13     setCount((currCount) => {
14       return currCount + 1;
15     });
16   };
17
18   return (
19     <div>
20       <h3>Count = {count}</h3>
21       <button onClick={incCount}>Increase Count</button>
```

## States in React


Count = 3.864717071287336

Increase Count

Default levels ▾ | No Issues

- init was executed
- component was rendered
- init was executed
- component was rendered
- init was executed
- component was rendered
- init was executed
- component was rendered

>

src >  Counter.jsx >  Counter

```
1  import { useState } from "react";
2
3  function init() {
4    console.log("init was executed");
5    return Math.random();
6  }
7
8  export default function Counter() {
9    let [count, setCount] = useState(init); // initialization
10   console.log("component was rendered");
11
12   let incCount = () => {
13     setCount((currCount) => {
14       return currCount + 1;
15     });
16   };
17
18   return (
19     <div>
20       <h3>Count = {count}</h3>
21       <button onClick={incCount}>Increase Count</button>
```



States in React

**Count = 9.134664339854154**

Increase Count

Elements Console

top Filter

Default levels No Issues

init was executed

component was rendered

9 component was rendered

>