



# CHANGING COMPLEX STATE



# React dev tools

- Installing and usages - <https://www.youtube.com/watch?v=rb1GWqCJid4>
- Google chrome externsion - <https://chrome.google.com/webstore/detail/react-developer-tools/fmkadmapgofadopljbjfkapdkoienihi>

# Learning objective

- Complex state – managing [the state of JS object](#) by retrieving the previous value of the object properties.
- CodeSandbox - <https://codesandbox.io/s/students-changing-complex-state-forked-8xyppm?file=/src/index.js>

# Initial code -

```
App.jsx x
1 import React from "react";
2
3 function App() {
4   return (
5     <div className="container">
6       <h1>Hello</h1>
7       <form>
8         <input name="fName" placeholder="First Name" />
9         <input name="lName" placeholder="Last Name" />
10        <button>Submit</button>
11      </form>
12    </div>
13  );
14 }
15
16 export default App;
17
```

Browser Tests

https://mcvw7j.csb

Hello

First Name

Last Name

Submit

Challenge: How would you solve it?

To solve this, we need to have a variable that can hold the both input values.

```
1 import React, { useState } from "react";
2
3 function App() {
4   const [fName, setFName] = useState("");
5   const [lName, setLName] = useState("");
6
7   function updateFName(event) {
8     const firstName = event.target.value;
9     setFName(firstName);
10  }
11  function updateLName(event) {
12    const lastName = event.target.value;
13    setLName(lastName);
14  }
15
16  return (
17    <div className="container">
18      <h1>
19        Hello {fName} {lName}
20      </h1>
21      <form>
22        <input
23          name="fName"
24          onChange={updateFName}
25          placeholder="FirstName"
26          value={fName}
27        />
28        <input
29          name="lName"
30          onChange={updateLName}
31          placeholder="LastName"
32          value={lName}
33        />
34        <button>Submit</button>
35      </form>
36    </div>
37  );
38 }
```

1

3

2

https://mcvw7j.csb.app/

Hello Shoaib Ahamed

Shoaib

Ahamed

Submit

Problems 0

Console 0

React DevTools 0

It does the job but there're multiple functions, variables, states. How can we optimize it? Can you manage an object as state?



# Steps -

- In useState instead of storing a single value (“”), we can store [an object](#).
- Let's delete the duplicated codes for the last name.
- Change the initial state fName into fullName. fullName is storing [an object](#).
- Change setFName as setFullName. SetFullName is setting the property values of the object.
- Set your initial state as an object, fName property being “” and lName property being “”.

App.jsx

```
1 import React, { useState } from "react";
2
3 function App() {
4   const [fullName, setFName] = useState({
5     fName: "",
6     lName: ""
7   });
8   // 1
9
10  function handleChange(event) {
11    event.target.value;
12  }
13
14  // 3
15
16  return (
17    <div className="container">
18      <h1>
19        Hello {fullName.fName} {fullName.lName}
20      </h1>
21      { /* 2      */ }
22      <form>
23        <input
24          name="fName"
25          onChange={handleChange}
26          // 4
27          placeholder="FirstName"
28          value={fullName.fName}
29        />
30        <input
31          name="lName"
32          onChange={handleChange}
33          // 4
34          placeholder="LastName"
35          value={fullName.lName}
36        />
37
38        <button>Submit</button>
```

Browser Test

https

Problems 1

- In the both input now, will call the same `function handleChange`.
- We need to get hold the previous value of `fullName` object, so we can add the part of it that changed based on user input. Other input value should remain as it was.
- But the challenge is since it's calling the same function, how to handle the `fName` or `lName` while keeping the previous value.
- How to know which input triggered the handle change?

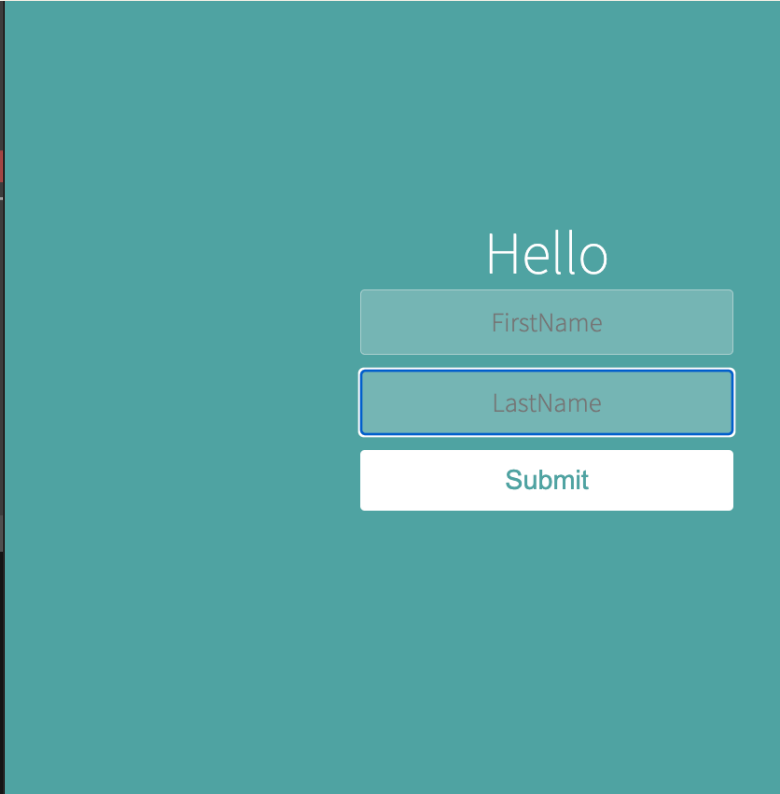
```
const [fullName, setFName] = useState({  
  fName: "",  
  lName: ""  
});
```

- Try this : In the input, we can use the name attribute to check which input was changed.

```
function App() {
  const [fullName, setFName] = useState({
    fName: "",
    lName: ""
  });
  // 1

  function handleChange(event) {
    const newValue = event.target.value;
    const inputName = event.target.name;
    console.log(newValue);
    console.log(inputName);
  }
  // 3

  return (
    <div className="container">
      <h1>
        Hello {fullName.fName} {fullName.lName}
      </h1>
      { /* 2 */ }
      <form>
        <input
          name="fName"
          onChange={handleChange}
          // 4
          placeholder="FirstName"
          value={fullName.fName}
        />
        <input
```



```

      </form>
    </div>
  );
}
```

Problems 1 Console 0 React DevTools 0

Console was cleared

s
fName
a
lName

Why what we type is not shown in the  
input field UI?

Because our **input fields** are set  
value={fullName.fName}  
value={fullName.lName}

and setName is not defined yet, so it's  
showing "" in the UI.

These are controlled components. If you  
comment out these two lines it will show  
input in the fields.

Does using a conditional solves our  
problem of first name vs last Name  
setting via setFullName?

# In the handleChange function after the two const -

```
if (inputName === "fName") {  
  setFullName({ fName: newValue });  
} else if (inputName === "lName") {  
  setFullName({ lName: newValue });  
}  
  
// 3  
  
return (  
  <div className="container">  
    <h1>  
      Hello {fullName} {fName} {lName}
```

Submit

Problems 4 Console 0 React DevTools 0 Filter

Console was cleared

s

fName

sh

fName

s



It Prints **either first or last name**. That's not ideal. Each time we type either field the value of the other field disappears. It happens because of the object thrown after typing.

We need to grab the previous value instead in order to keep the both values.

In `setFullName` we can **pass a function (not only single value)** which has a `prevValue` object in its argument. If you console log `prevValue`, you can see it has `fName` and `lName` properties. This is the key to the solution.

```
setFullName((prevValue) => {
  console.log(prevValue);
});

return (
  <div className="container">
    <h1>
      Hello {fullName.fName} {fullName.lName}
    </h1>
    <form>
      <input
        name="fName"
        onChange={handleChange}
```

```
22 |     </h1>
23 |     <form>
24 |       <input
```

View compiled

This screen is visible only in development. It will not appear if the app crashes in production. Open your browser's developer console to further inspect this error. This error overlay is powered by `react-error-overlay` used in `create-react-app`.

Problems 5

Console 0

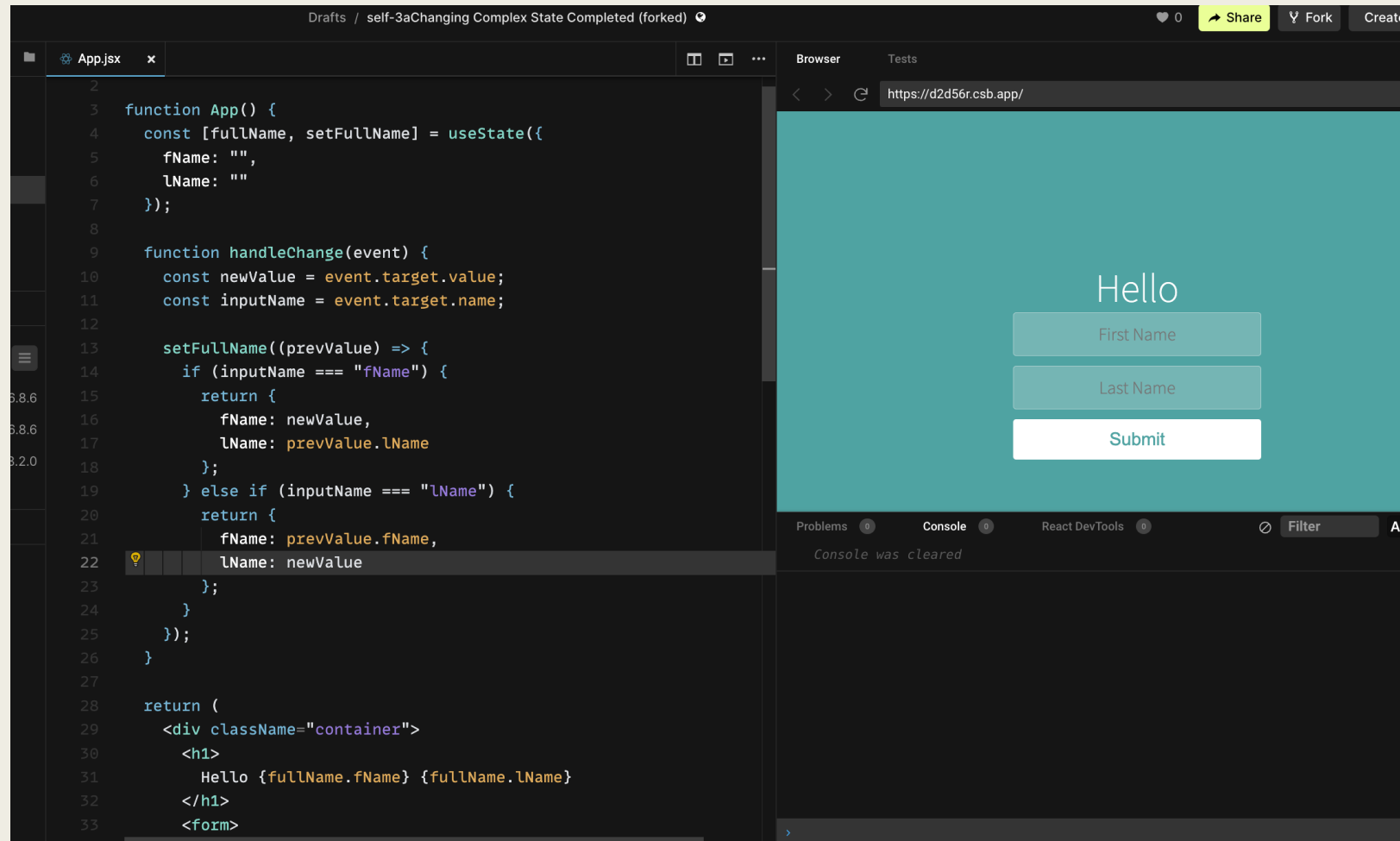
React DevTools 0

Console was cleared

▶ {fName: "", lName: ""}

! ▶ The above error occurred in the <App> component:

As soon as we type in the input field, React rerender the App component. App has state in the form of `fullName` object, and react remembers the value of this object. Now we can make use this remembered value as we update the first and last name.



We can also do object destructuring for our event to make our code cleaner.

```

5     fName: "",
6     lName: ""
7   });
8
9   function handleChange(event) {
10     const { value, name } = event.target;
11
12     setFullName((prevValue) => {
13       if (name === "fName") {
14         return {
15           fName: value,
16           lName: prevValue.lName
17         };
18       } else if (name === "lName") {
19         return {
20           fName: prevValue.fName,
21           lName: value
22         };
23       }
24     });
25   }

```

Hello

First Name

Last Name

Submit

Uncomment the value

```

20     lName: prevValue.lName,
21     lName: value
22   });
23   }
24   });
25   }
26
27   return (
28     <div className="container">
29       <h1>
30         Hello {fullName.fName} {fullName.lName}
31       </h1>
32       <form>
33         <input
34           name="fName"
35           onChange={handleChange}
36           placeholder="First Name"
37           value={fullName.fName}
38         />
39         <input
40           name="lName"
41           onChange={handleChange}
42           placeholder="Last Name"
43           value={fullName.lName}
44         />
45         <button>Submit</button>
46       </form>
47     </div>

```

https://5jj3v3.csb.app/

Hello

First Name

Last Name

Submit

- More about ES6 arrow function - [https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Functions/Arrow\\_functions](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Functions/Arrow_functions)
- Full code can be found - <https://codesandbox.io/s/self-3changing-complex-state-completed-forked-5jj3v3?file=/src/components/App.jsx:467-472>