

React - components

JSX

How to return JSX

Props - both type of components

We sent --- number, string, arr, obj, function to props

Event Handling in React -----

Event handler ---

Javascript onclick

JSX onClick

Javascript calling the handler function onclick="f1()"

JSX register the handler function and it is called by React Framework
onClick={f1}

Scenario --- HOW WILL THE HANDLER FUNCTION DISTINGUISH BETWEEN DIFFERENT CALLS?

Javascript -----

```
<button id="b1" onclick="f1(1)">OK1</button>
<button id="b2" onclick="f1(2)">OK2</button>
```

//SOLUTION 1

```
function f1()
{
    document.getElementById("b1").clicked == true
    .....
    document.getElementById("b2").clicked == true
    ///////////////
}
```

//SOLUTION 2

```
function f1(b)
{
    If(b==1) { for button 1 ..... }
    If( b==2) { for button 2.....}
}
```

React --- the handler method can get a parameter from the framework
That parameter is the event object
The event object gives us all details about the target of the event
event.target.name !!!!!

```
<button name="b1" onClick={f1}>OK1</button>
<button name="b2" onClick={f1}>OK2</button>
```

//SOLUTION 1

```
function f1(event)
{
    If(event.target.name == "b1" )
    .....
```

```

        If(event.target.name == "b2" )
            ///////////////
    }
    -----
    <button name="b1" onClick={()=>{ f2(1)}}>OK1</button>
    <button name="b2" onClick={()=>{ f2(2)}}>OK2</button>

//SOLUTION 2
function f2(x)
{
    If(x==1 )
        //.....
    If(x==2)
        ///////////////
}

```

==	Compares the two values
===	Strictly compares types also , more type safe

Scenario ----- We have a text field . When we enter data that should be shown in a <p> tag below

Javascript

```

<input type="text" id ="ip1"  onChange="f1()" />

function f1()
{
    Str = getelebyId("ip1" ).value
    getElebyId("pId").innerHTML = str
}

```

React----

When the component is rendered-displayed for the First time ----- All the {} are evaluated and displayed

When we change the variable values during event firing ---- **RERENDERING does not HAPPEN !!!**

To solve this issue --- we have to use STATE !!!!

State is a special property in REACT

whenever the value of state changes REACT initiates RERENDERING !

In function components We use a HOOK = useState ! To set the state variable !!!

the handler function within the class does not get the "this" of the class !!!

Solutions

1. Bind the handler function to this
2. Use arrow function instead of named function