
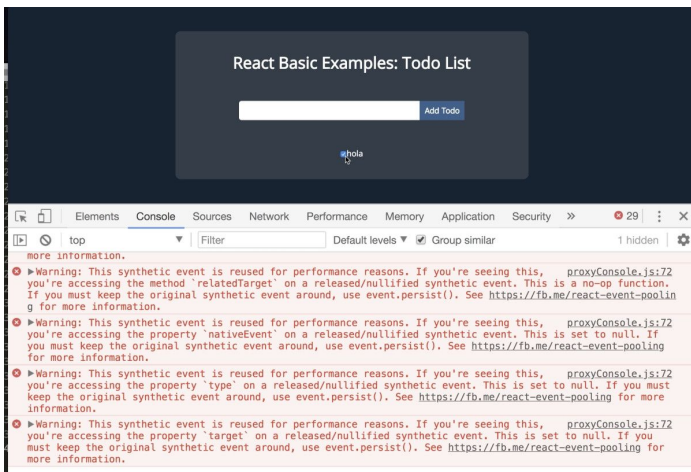


2nd June 2020	REACTJS→ Sundeep Slr	Woodpecker 
Revised Basics → This, Call, apply and bind,map Discussed State in React, State v/s Props Events and Synthetics Events	Lecture flow: <ul style="list-style-type: none"><li>• Event and Synthetic Event</li><li>• Passing fn as prop</li><li>• This anomaly and how to bind/arrow fn</li><li>• Usage of setState</li></ul>	
Topics & Explanation: Github link for Class Codes		Reference
<u>DOM Event and React Syn. Event :</u> <ul style="list-style-type: none"><li>• In a regular DOM event,even after the event propagation,the value will still be located.So we still have a Reference</li><li>• But ,in the case of <u>Synthetic Event in React</u>,</li></ul> <p><u>DOM event:</u></p> <pre>46 const button = document.querySelector("button"); 47 let storeEve = null; 48 button.addEventListener("click", function (event) { 49   storeEve = event; 50   console.log("Event happening", storeEve.target); 51   setTimeout(function () { 52     console.log("Event finished", storeEve.target); 53   }, 1000); 54 });</pre> <p><u>React Synthetic event: (on toggle component,refer code)</u></p> <pre>21 let storeEve = null; 22 const handleClick = () =&gt; { 23   storeEve = event; 24   console.log("Event happening", storeEve.target); 25   setTimeout(function () { 26     console.log("Event finished", storeEve.target); 27   }, 1000); 28   customOnClick(isSwitchedOn); 29 }; 30 31 let innerCircleComponent = null; 32 if (isSwitchedOn === true) { 33   innerCircleComponent = &lt;div style={innerCircleStyle}&gt;&lt;/div&gt;; 34 } 35 return ( 36   &lt;div onClick={handleClick} style={outerCircleStyle}&gt; 37     {innerCircleComponent} 38   &lt;/div&gt;</pre>		<p>Click me!</p> <div> React JS</div> <p><a href="#">Handling Events – React</a></p> <p><a href="#">Understanding ReactJS — setState</a></p> <p><a href="#">Warning: This synthetic event is reused for performance reasons happening</a></p> <p><a href="#">SyntheticEvent - React</a></p>
<u>Gives this kind of Error in Synthetic Event:</u>		



## Handling Events – React

## What is 'this' in React?

## Passing Functions to Components – React

## Understanding React 'setState' |

## Synthetic Events in React:

- Used for performance Reason
- Helps in clearing the memory
- These are not stored like we do in Regular DOM.

## This keyword Anomaly, State and the Toggle Component:

- If we can pass functions as props in React.

```

toggle: false
};
// const appInstance = this;
// this.handleToggleState = this.handleToggleState.bind(appInstance);
}

handleToggleState = toggleState => {
  this.setState({ toggle: !toggleState });
};

render() {
  return (
    <div className="App">
      <Form />
      <Toggle
        isSwitchedOn={this.state.toggle}
        customOnClick={this.handleToggleState}
      />
    </div>
  );
}
  
```

Diagram annotations:

- A red arrow points from the text 'adding in state' to the `toggle: false` line.
- A red arrow points from the text 'func to change state' to the `handleToggleState` function definition.
- A red arrow points from the text 'passing func as prop' to the `customOnClick={this.handleToggleState}` prop in the `<Toggle>` component.

## Steps in Toggle passing fn into the toggle component:

1. Passed the function into the toggle component as a prop
2. Went inside the toggle component
3. Called that func whenever we are having a handleClick event

```

let storeEve = null;
const handleClick = () => {  2
  // storeEve = event;
  // console.log("Event happening", storeEve.target);
  // setTimeout(function () {
  //   console.log("Event finished", storeEve.target);
  // }, 1000);
  customOnClick(isSwitchedOn);
};

let innerCircleComponent = null;
if (isSwitchedOn === true) {
  innerCircleComponent = <div style={innerCircleStyle}></div>;
}
return (  3
  <div onClick={handleClick} style={outerCircleStyle}>
    {innerCircleComponent}
  </div>
);
};

```

We get an Error because we haven't fixed "this" keyword anomaly!!!

- Normal functions work dynamically, don't care where it is called.
- It would give undefined because we are working with virtual DOM and the context is regular DOM
- SO, we voluntarily need to "bind" it with the Class component (use constructor fn)
- Or, simply use an arrow function. because arrow have lexical scope and don't have their this keyword. In other words, the value of this is determined where it is defined

```

toggle: false
};
// const appInstance = this;
// this.handleToggleState = this.handleToggleState.bind(appInstance);
}

```

```

handleToggleState = toggleState => {
  this.setState({ toggle: !toggleState });
};

```



or

```

  }
],
toggle: false
};
const appInstance = this;
this.handleToggleState = this.handleToggleState.bind(appInstance);
}

```

Bind it within Constructor Function

## Set Change:

## Understanding React 'setState'

```
// Wrong
this.state.fullName = 'eduardo';

//Correct
this.setState({
  fullName: 'eduardo',
});
```

## Last 30 mins → Discussed about the above topics by implementing it to a Form Component

```
1  import React, { Component } from "react";
2
3  class Form extends Component {
4    state = {
5      name: "",
6      email: "",
7      password: ""
8    };
9
10   handleChange = event => {
11     this.setState({ [event.target.name]: event.target.value });
12   };
13
14   handleSubmit = event => {
15     event.preventDefault();
16     console.log(this.state.name, this.state.password, this.state.email);
17   };
18 }
```

```

19   render() {
20     console.log("Rendering");
21     return (
22       <form onSubmit={this.handleSubmit}>
23         <input
24           onChange={this.handleChange}
25           type="text"
26           name="name"
27           value={this.state.name}
28         />
29         <input
30           onChange={this.handleChange}
31           type="email"
32           name="email"
33           value={this.state.email}
34         />
35         <input
36           onChange={this.handleChange}
37           type="password"
38           name="password"
39           value={this.state.password}
40         />
41         <input type="submit" name="" value="Sign up" />
42       </form>
43     );
44   }
45 }
46
47 export default Form;
48

```

## [Useful Links:](#)

[Handling Events – React](#)

[Understanding ReactJS — setState](#)

[Warning: This synthetic event is reused for performance reasons happening](#)

[SyntheticEvent - React](#)

[What is 'this' in React?](#)

[Passing Functions to Components – React](#)

[Understanding React `setState` |](#)

Link to the lecture video:- [DASHBOARD](#) : [Github link for Class Codes](#)

**Summary:** →ReactJS→ Event and Synthetic Event→  
Passing fn as prop→  
This anomaly and how to bind/arrow fn→  
Usage of setState