Event Handling in React

Introduction

Events make the **web app interactive and responsive** to the user. Most of the time, you don't have static UIs; you need to build elements that are smart enough to respond to user actions. So far in the examples we've seen, we brought in a certain level of user interaction such as performing some action on a button click. By now, you would have realized that handling events with React elements is very similar to handling events on DOM elements.

React Event Handling vs DOM Event handling

1. With JSX in ReactJs, you pass a function as event handler and in DOM element we pass function as string

```
// event handling in ReactJs element
<input id="inp" name="name" onChange={onChangeName} />
```

```
// event handling in DOM element
<input id="inp" name="name" onchange="onChangeName()"/>
```

2. In DOM elements the event name is in lowercase while in ReactJs it is in camalCase. List of events are as follows.

Mouse

- onClick
- onContextMenu
- onDoubleClick
- onMouseDown
- onMouseEnter
- onMouseLeave
- onMouseMove
- onMouseOut
- onMouseOver
- onMouseUp

Image

- onLoad
- onError

Selection

- onSelect

Form

- onChange
- onInput
- onSubmit

Focus

- onFocus
- onBlur

Keyboard

- onKeyDown
- onKeyPress
- onKeyUp

UI

- onScroll

3. Cannot return false to prevent default behavior in React. Must call preventDefault explicitly.

HTML

```
<form onsubmit="console.log('You clicked submit.'); return false">
  <button type="submit">Submit</button>
</form>
```

React

```
function Form() {
  function handleSubmit(e) {
    e.preventDefault();
    console.log('You clicked submit.');
      }
  return (<form onSubmit={handleSubmit}>
   <button type="submit">Submit</button>
   </form>);
        }
```

Event Registration

It tells the browser that a particular function should be called whenever a definite event occurs i.e whenever an event is triggered, the function which is bound to that event should be called. Essentially, it allows you to add an event handler for a specified event. This can be used to bind to any event, such as keypress , mouseover or mouseout . Since class methods are not bound by default, it's necessary to bind functions to the class instance so that the this keyword would not return "undefined".

Synthetic Event Objects

React event handling system is known as Synthetic Events. The event object passed to the event handlers are SynthticEvent Objects. It is a wrapper around the DOMEvent object. The event handlers are registered at the time of rendering. Whenever you call an event handler within ReactJS, they are passed an instance of SyntheticEvent. A SyntheticEvent event has all of its usual **properties and methods**. These include its **type, target, mouse coordinates,** and so on. React defines these synthetic events according to the W3C spec to take care of **cross-browser compatibility.**

SyntheticEvent that wraps a **MouseEvent will have access to mouse-specific properties** such as the following:

```
boolean altKey                          boolean metaKey
number button                           number pageX
number buttons                          number pageY
number clientX                          DOMEventTarget relatedTarget
number clientY                          number screenX
boolean ctrlKey                         number screenY
boolean getModifierState(key)           boolean shiftKey
```

A SyntheticEvent that wraps a **KeyboardEvent will have access to keyboard-related properties** such as the following:

```
boolean altKey                          string locale
number charCode                         number location
boolean ctrlKey                         boolean metaKey
boolean getModifierState(key)           boolean repeat
string key                              boolean shiftKey
number keyCode                          number which
```

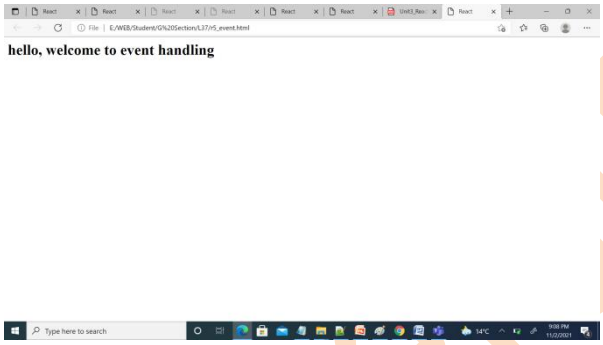
Coding example 1: Simple code to demo event handling

```
<body>
        <div id="root"></div>
        <script type = "text/babel">
                class NewOne extends React.Component    {
                        constructor()
                        {       super();
                                this.state = {content:"hello, welcome to event handling"}
                        }
                        render()
                        {return <h1 onClick = {this.fun1}>{this.state.content}</h1>  }
                        fun1=()=>
                        {       this.setState({content:"new text"})             }
```

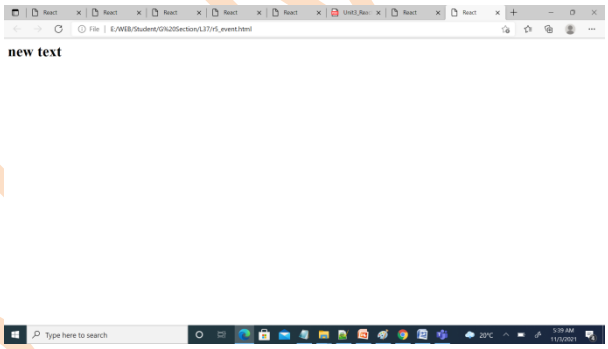
```
        }
        ReactDOM.render(<NewOne/>,document.getElementById("root"))
    </script>
<body>
```

Output:

Case 1: When page is loaded



Case 2: When text is clicked



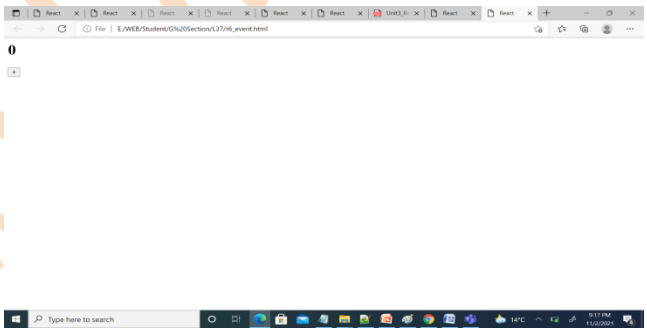
- **Coding example 2: Requirement is to click on the + button, the value of counter must be incremented by one**

```
        <body>
            <div id="root"></div>
            <script type = "text/babel">
                class NewOne extends React.Component    {
                constructor()
                {  super();              this.state = {counter:0}        }
```

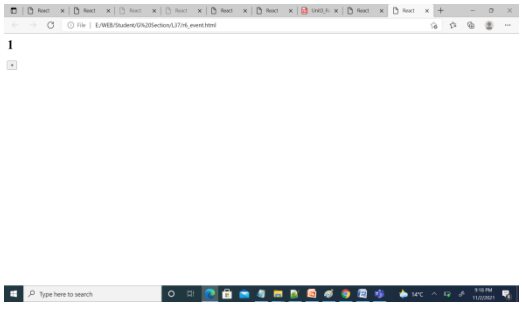
```
            render()        {
                return (<div>
                            <h1 >{this.state.counter}</h1>
                            <button onClick = {this.fun1}>+</button>
                            </div>)

            }
            fun1=()=>
            {       //this.setState(counter:this.state.counter+1})
                    this.setState((prevState) => ({counter:prevState.counter+1}))
            }
        }
        ReactDOM.render(<NewOne/>,document.getElementById("root"))
    </script>
<body>
```

Output:

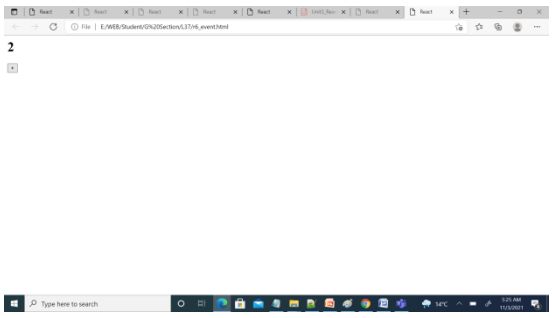
Case 1: when the page is loaded



Case 2: When + button is clicked once

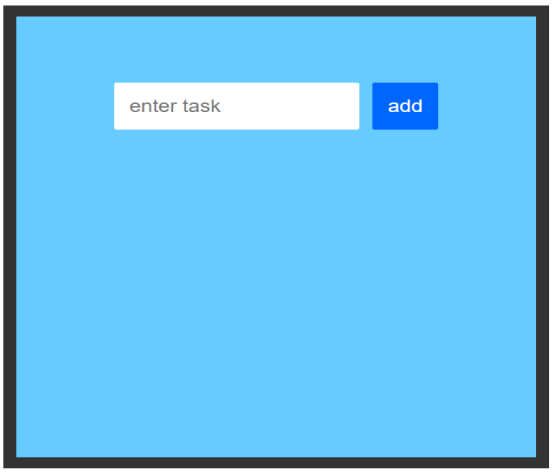


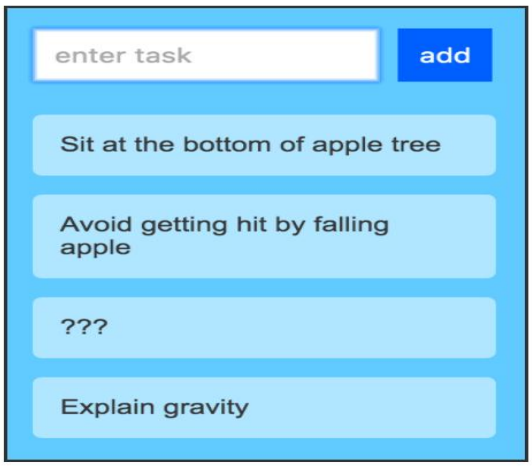
Case 3: When + button is clicked again



Practice Programs

1. Build an awesome todo list as shown below. Type the task in the input box provided and click on add button. This must create an element below the input box with a new background color for this. Clicking on the task directly, must delete the element from the list.





2. Simulate the below to obtain the current date string on the click of a button.

Current Time:

Sat Oct 16 2021 17:10:00 GMT+0530 (India Standard Time)

Get Current Time!