

30 Days Of React: Events



LinkedIn



Follow @asabeneh

776

Author: [Asabeneh Yetayeh](#)

October, 2020

[<< Day 10](#) | [Day 12 >>](#)

- [Events](#)
 - [What is an event?](#)
- [Exercises](#)
 - [Exercises: Level 1](#)
 - [Exercises: Level 2](#)
 - [Exercises: Level 3](#)

Events

What is an event?

An event is an action or occurrence recognized by a software. To make an event more clear let's use the daily activities we do when we use a computer such as clicking on a button, hover on an image, pressing a keyboard, scrolling the mouse wheel and etc. In this section, we will focus only some of the mouse and keyboard events. The react documentation has already a detail note about [events](#).

Handling events in React is very similar to handling elements on DOM elements using pure JavaScript. Some of the syntax difference between handling event in React and pure JavaScript:

- React events are named using camelCase, rather than lowercase.

- With JSX you pass a function as the event handler, rather than a string.

Let's see some examples to understand event handling.

Event handling in HTML

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <title>30 Days Of React App</title>
  </head>
  <body>
    <button onclick="greetPeople()">Greet People</button>
    <script>
      const greetPeople = () => {
        alert('Welcome to 30 Days Of React Challenge')
      }
    </script>
  </body>
</html>
```

In React, it is slightly different

```
import React from 'react'
// if it is functional components
const App = () => {
  const greetPeople = () => {
    alert('Welcome to 30 Days Of React Challenge')
  }
  return <button onClick={greetPeople}> </button>
}
```

```
import React, { Component } from 'react'
// if it is functional components
class App extends Component {
  greetPeople = () => {
    alert('Welcome to 30 Days Of React Challenge')
  }
  render() {
    return <button onClick={this.greetPeople}> </button>
  }
}
```

Another difference between HTML and React event is that you cannot return false to prevent default behavior in React. You must call preventDefault explicitly. For example, with plain HTML, to prevent the default link behavior of opening a new page, you can write:

Plain HTML

```
<a href="#" onclick="console.log('The link was clicked.');" return false">
  Click me
</a>
```

However, in React it could be as follows:

```
import React, { Component } from 'react'
// if it is functional components
class App extends Component {
  handleClick = () => {
    alert('Welcome to 30 Days Of React Challenge')
  }
  render() {
    return (
      <a href="#" onClick={this.handleClick}>
        Click me
      </a>
    )
  }
}
```

Event handling is a very vast topic and in this challenge we will focus on the most common event types. We may use the following mouse and keyboard events. *onMouseMove*, *onMouseEnter*, *onMouseLeave*, *onMouseOut*, *onClick*, *onKeyDown*, *onKeyPress*, *onKeyUp*, *onCopy*, *onCut*, *onDrag*, *onChange*, *onBlur*, *onInput*, *onSubmit*

Let's implement some more mouse and keyboard events.

```
// index.js
import React, { Component } from 'react'
import ReactDOM from 'react-dom'

class App extends Component {
  state = {
    firstName: '',
    message: '',
    key: ''
  }
  handleClick = (e) => {
    // e gives an event object
    // check the value of e using console.log(e)
    this.setState({
      message: 'Welcome to the world of events',
    })
  }
}
// triggered whenever the mouse moves
```

```
handleMouseMove = (e) => {
  this.setState({ message: 'mouse is moving' })
}
// to get value when an input field changes a value
handleChange = (e) => {
  this.setState({
    firstName: e.target.value,
    message: e.target.value,
  })
}

// to get keyboard key code when an input field is pressed
// it works with input and textarea
handleKeyPress = (e) => {
  this.setState({
    message:
      `${e.target.value} has been pressed and the keycode is` + e.charCode,
  })
}

// Blurring happens when a mouse leave an input field
handleBlur = (e) => {
  this.setState({ message: 'Input field has been blurred' })
}

// This event triggers during a text copy
handleCopy = (e) => {
  this.setState({
    message: 'Using 30 Days Of React for commercial purpose is not allowed',
  })
}

render() {
  return (
    <div>
      <h1>Welcome to the World of Events</h1>

      <button onClick={this.handleClick}>Click Me</button>
      <button onMouseMove={this.handleMouseMove}>Move mouse on me</button>
      <p onCopy={this.handleCopy}>
        Check copy right permission by copying this text
      </p>

      <p>{this.state.message}</p>
      <label htmlFor=''> Test for onKeyPress Event: </label>
      <input type='text' onKeyPress={this.handleKeyPress} />
      <br />

      <label htmlFor=''> Test for onBlur Event: </label>
      <input type='text' onBlur={this.handleBlur} />

      <form onSubmit={this.handleSubmit}>
        <div>
          <label htmlFor='firstName'>First Name: </label>
          <input
            onChange={this.handleChange}
            name='firstName'
          />
        </div>
      </form>
    </div>
  )
}
```

```
        value={this.state.value}
      />
    </div>

    <div>
      <input type='submit' value='Submit' />
    </div>
  </form>
</div>
)
}
}

const rootElement = document.getElementById('root')
// we render the JSX element using the ReactDOM package
ReactDOM.render(<App />, rootElement)
```

Exercises

Exercises: Level 1

1. What is an event?
2. What is the different between an HTML element event and React event?
3. Write at least 4 keyboard events?
4. Write at least 8 mouse events?
5. What are the most common mouse and keyboard events?
6. Write an event specific to input element?
7. Write an event specific to form element?
8. Display the coordinate of the view port when a mouse is moving on the body?
9. What is the difference between onInput, onChange and onBlur?
10. Where do we put the onSubmit event ?

Exercises: Level 2

Implement the following using onMouseEnter event

30 Days Of React



Exercises: Level 3

Coming

🎉 CONGRATULATIONS ! 🎉

[<< Day 10](#) | [Day 12 >>](#)