Course › Modul... › Event ... › Event ...

Audit Access Expires Apr 20, 2020
You lose all access to this course, including your progress, on Apr 20, 2020.

Event Handlers

Adding Event Handlers

Handling events in React is similar to handling events in HTML. To attach an event handler to a React Element, assign the event handler method to the appropriate event attribute.

Here are the main differences:

- One main difference in React is that you can use JSX brackets to specify the event handler function instead of declaring it as a string.
- The next difference is that React events are named using camelCase instead of being all lowercase. For example, onclick and onkeypress in HTML become onClick and onKeyPress in React respectively.

Click event in React:

```
<button onClick = {clickHandler} >Click Me</button>
```

Click event in HTML:

```
<button onclick = "clickHandler()" >Click Me</button>
```

Binding Event Handlers to Class Components

Event handlers can be defined as methods within a class Component.

In this example, the clickHandler method is defined within the Class Component and is assigned to the onClick attribute of the `<button>` element. The component renders a button that increments the number displayed inside it whenever it is clicked.

```

class Counter extends React.Component{
    constructor(props){
        super(props)
        this.state = {count:0}
        //binding is necessary to make `this` point to the correct
object
        this.clickHandler = this.clickHandler.bind(this)
    }
    clickHandler(){
      //increments the count of the state
      this.setState((prevState,props) => {
        return {count: prevState.count + 1}
      })
    }
    render(){
        //renders a button that displays the state count
        return <button onClick = {this.clickHandler}>
{this.state.count}</button>
    }
}

ReactDOM.render(
  <Counter/>,
  document.getElementById("root")
)
```

The bind() method is used to bind the clickHandler() method's `this` keyword to the component instance. Without binding the function, the function will have its `this` keyword point to an incorrect object and the setState() method will not work correctly.

Binding example:

```
this.clickHandler = this.clickHandler.bind(this)
```

An alternative to using bind() is to attach the event handler to the React Element using an ES6 arrow function. The arrow function will automatically have its `this` keyword point to the enclosing scope which happens to be the component instance.

Fat arrow example:

```

<button onClick = {{ () => this.clickHandler()}}>{this.state.count}
</button>
```