Handling events with React elements is very similar to handling events on DOM elements.

React events are named using camelCase, rather than lowercase.

For example, the HTML:

<button onclick="activateLasers()">

Activate Lasers

</button>

is slightly different in React:

<button onClick={activateLasers}>

Activate Lasers

</button>

In React 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:

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

Click me

</a>

In React, this could instead be:

function ActionLink() {

function handleClick(e) { e.preventDefault(); console.log('The link was clicked.'); }

return (

<a href="#" onClick={handleClick}> Click me

</a>

);

}

Note: Here, e is a synthetic event. React defines these synthetic events according to the [W3C spec](https://www.w3.org/TR/DOM-Level-3-Events/), so you don’t need to worry about cross-browser compatibility. React events do not work exactly the same as native events.