



Hooks

Setting a click event

Rolando Gonzalez, 2022

Content

- Events
- Hook
- useState
- useEffect

Events

- In React you have the same events as in vanilla JavaScript
- When you add event in your JSX you must use **(lower) camelCase**
 - `<MyComponent onClick={ functionName } />`

Code example onClick event

- In this code example the component Alert has two functions that are called by buttons.
- Note that you can use arrow functions or regular functions inside a component

```
src > components > JS Alert.js > [🔍] default
1  import React from 'react';
2
3  const Alert = () => {
4
5      const sayHello = () => alert("Hello! :)");
6
7      const sayHello2 = () => {
8          let greeting = "Good day!";
9          alert( greeting );
10     }
11
12     return (
13         <section>
14             <button onClick={ sayHello }>Say hello!</button>
15             <button onClick={ sayHello2 }>Say good day!</button>
16         </section>
17     );
18 }
```

Passing in argument

JS GoodDay.js X

src > components > JS GoodDay.js > [🔗] default

```
1  import React from 'react';
2
3  const GoodDay = () => {
4
5      const sayHello = ( name ) => {
6          alert(`Good day to you ${name}`);
7      }
8
9      return (
10         <button onClick={() => sayHello( "Jordan Walke" )}>Say good day</button>
11     );
12 }
13
14 export default GoodDay;
```

More on handling events in React

- <https://reactjs.org/docs/handling-events.html>

Hooks

What are Hooks and useState?

- Hooks are a set of techniques in React among other to maintain a state of a component, make a first event call when a component starts and so on.
- useState is a technique which keeps state of a variable and creates rerender of page

- You must import useState from 'react' in the component that uses it.

`import React, { useState } from 'react';`

- The template for making a useState hook:

`const [variable, functionForChangingVariable] = useState(optionalDefaultValue);`

Simple code example for a counter hook

- A variable named count is created.
- setCounter is a function for changing the variable count.
- count is set with a default value of 100 with useState

```
JS Incrementator.js X JS App.js
src > components > JS Incrementator.js > ...
1  import React, { useState } from 'react';
2
3  const Counter = () => {
4
5      const [ count, setCounter ] = useState( 100 );
6
7      return (
8          <section>
9              <button onClick={ () => setCounter (count - 100 ) }>-</button>
10             <span>{ count }</span>
11             <button onClick={ () => setCounter( count + 100 ) }>+</button>
12         </section>
13     )
14 }
15
16 export default Counter;
```

useEffect

- useEffect is a hook which among other can be used to do something, 1 time, when the component is mounted. In this code example make a http call.

```
4 import React, { useState, useEffect } from 'react';
5 import axios from 'axios';
6
7
8 function App() {
9
10   useEffect( () => {
11     axios.get(url)
12       .then( ( result ) => {
13         console.log(result);
14       })
15   }, []);
16
17   return (
18     <div>
19
20     </div>
21   );
22 }
23
24 export default App;
```

More on Hooks

- Hooks useState:
 - <https://reactjs.org/docs/hooks-state.html>
- Hooks useEffect:
 - <https://reactjs.org/docs/hooks-effect.html>