



2022 © Wawiwa Tech | Confidential



Introducción a Eventos

React puede ejecutar acciones basadas en los eventos del usuario, como los eventos HTML del DOM.

React tiene **los mismos** eventos que HTML: click, change, mouseover, etc.

Sintaxis:

Los eventos de React están escritos en sintaxis camelCase. Por ejemplo:

~~onclick~~ → **onClick** , ~~onchange~~ → **onChange**

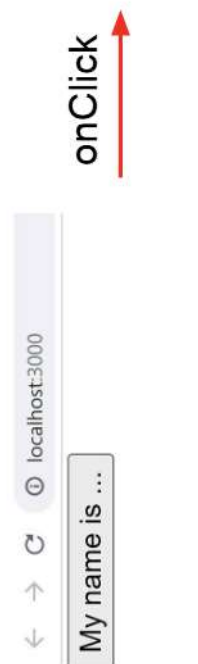
Los controladores de los eventos de React son escritos dentro de llaves. Por ejemplo:

~~onclick="myFunction()"~~ → **onClick={myFunction}**



Ejemplo de eventos

```
function App() {  
  
  const handleClick = () => {  
    alert("Adam");  
  }  
  return (  
    <>  
    <button onClick={handleClick}>My name is ...</button>  
    </>  
  )  
}
```





Ejemplo de eventos - sin pasar un argumento

If there are no arguments to pass, we simply write the name of the function without parentheses:

```
function App() {  
  
  const handleClick = () => {  
    alert("Adam");  
  }  
  
  return (  
    <>  
    <button onClick={handleClick}>My name is ...</button>  
    </>  
  )  
}
```





Ejemplo de eventos - pasando un argumento

Si se necesita pasar argumentos, usa una función de flecha:

```
function App() {  
  
  const handleClick = (name) => {  
    alert(name);  
  }  
  return (  
    <>  
    <button onClick={()=>handleClick("Adam")}>My name is ...</button>  
    </>  
  )  
}
```





Ejemplo de eventos - Con objeto "event"

Si queremos pasar el evento disparado (en este caso el evento "click"), usa una función de flecha:

```
function App() {  
  
  const handleClick = (name, e) => {  
    alert(name);  
  }  
  return (  
    <>  
    <button onClick={e=>handleClick("Adam", e)}>My name is ...</button>  
    </>  
  )  
}
```



useState - Ejemplo 1

Si queremos actualizar lo mostrado en la pantalla, llamamos la función “set” con un valor. En el siguiente ejemplo, queremos actualizar el estado cuando el botón es presionado.

```
import { useState } from "react";
function App() {
  const [name, setName] = useState("James")
  const handleClick = (newName) => {
    setName(newName);
  }
  return (
    <>
      <h3>My name is {name}</h3>
      <button onClick={()=>handleClick("Adam")}>change the name</button>
    </>
  )
}
export default App;
```



React va a guardar el siguiente estado, renderizará tu componente otra vez con los nuevos valores y actualizará el UI.



useState - Ejemplo 2

```
import React, {useState} from "react";
function App() {
  const [count, setCount] = useState(0);
  return (
    <>
      <p>{count}</p>
      <button onClick={()=>setCount(5)}>set count 5</button>
    </>
  )
}
export default App;
```

import

Set our state

0

set count 5

before after

0 5

set count 5

set count 5



Ejercicio 5

1. Asigna un controlador de eventos onClick al botón "increment" que aumente el valor de "count" en 1
2. Asigna un controlador de eventos onClick al botón "decrement" que reduzca el valor de "count" en 1.