

7.4. El objeto \$event

En JavaScript, cuando se produce un evento, el navegador crea automáticamente un objeto el cual guardan varios datos muy útiles (como la posición del ratón, si hay teclas pulsadas, el elemento que ha producido el evento, etc). Este objeto lo recogíamos de la siguiente forma: `miFuncion ($event){...}`.

En Angular, esto lo podemos hacer con el objeto `$event`. Igual que en JavaScript, podemos recoger por ejemplo, el `target` que es una referencia al elemento que causó el evento.

Vamos a hacer un pequeño ejemplo para que veamos como funciona:

Dentro de la clase del componente `zapatillas` vamos a crear un método llamado `mostrarEvento()` el cual va a contener un parámetro (le llamaremos `event`) que va a contener la información del evento que vamos a crear en la vista:

```
mostrarEvento(event: any) {  
  console.log(event.target.textContent);  
}
```

ACLARACIÓN:

Si ponemos a `event` de tipo `EVENT` no podemos usar ni `innerHTML`, ni `TextContent`, tendríamos que ponerlo de tipo `any`. Si queremos ponerlo tipo `event` y queremos usar por ejemplo `textContent`, haríamos lo siguiente:

```
let texto = (event.target as HTMLElement).textContent?.trim();  
console.log(texto);
```

De esta manera, debemos decirle a que consideremos a `event.target` como un elemento `HTML`, que sí que tiene la propiedad `textContent`.

La función `trim()` se usa para eliminar cualquier espacio en blanco extra.

Ahora, en la vista del componente vamos a añadir un botón que cuando le hagamos click, se ejecute el evento **mostrarEvento()** el cual imprimirá por la consola el texto que esté escrito en el botón (en este caso ‘*Hazme click y abre la consola*’):

```
<button (click)="mostrarEvento($event)">Hazme click y  
abre la consola</button>
```

Ahora, si nos vamos a nuestro componente **zapatillas** y pulsamos dicho botón, podemos ver que en la consola nos aparece el texto del botón, como hemos dicho anteriormente.