



Angular

3.3. Event Binding

1. Event binding

- Vamos a añadir un botón nuevo y vamos a asociarle un evento en TypeScript cuando hagamos clic. Este evento va colocado entre paréntesis y se asigna a un método de la clase TS.

`<button (click)="onSave()">Save</button>`


**Tipo de
evento**


**Método
en TS**

1. Event binding

- Tenemos un ejemplo del evento click asociado a un método.

<> app.component.html X

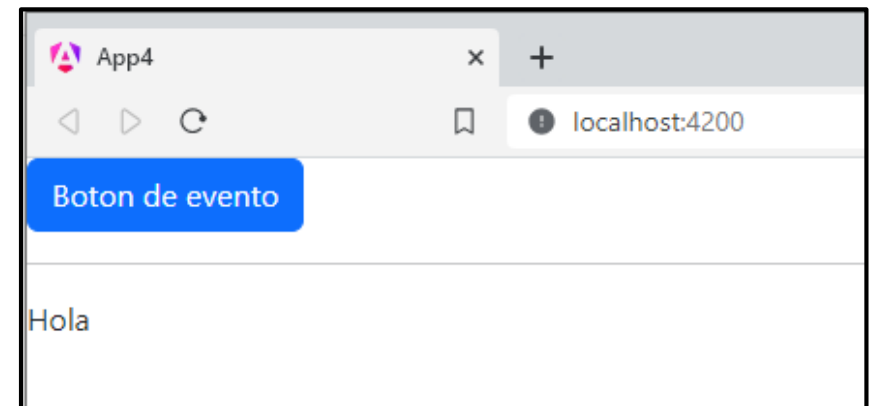
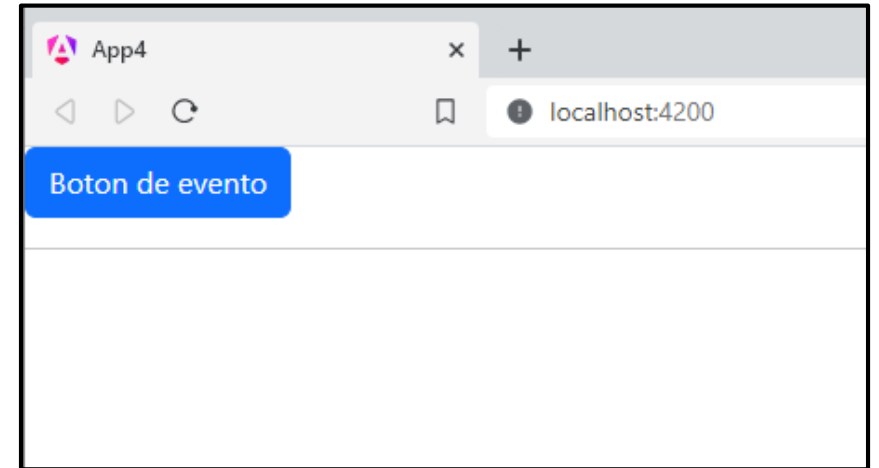
src > app > <> app.component.html > ...

Go to component

```
1  <button class="btn btn-primary" (click)="evento()">Boton de evento
    </button>
2  <hr>
3  {{ mensaje }}
```

1. Event binding

```
export class AppComponent {  
  title = 'app4';  
  mensaje="";  
  
  tabnine: test | explain | document | ask  
  evento() {  
    this.mensaje = "Hola";  
  }  
}
```




2. Event binding

- Angular incluye **\$event** que contiene la información sobre un evento.
- El tipo de **\$event** depende del evento de destino, por ejemplo, si el evento de destino es un evento de elemento DOM nativo, entonces es un objeto.

2. Event binding

- Vamos a ver un ejemplo de este tipo:

```
<> app.component.html ×   
src > app > <> app.component.html > ...  
Go to component  
1 <input type="text" placeholder="Introduce texto" name="texto" (input)  
  ="evento2($event)">  
2  
3 <hr>  
4  
5 {{ mensaje }}  
6 |
```

2. Event binding

```
export class AppComponent {  
  title = 'app4';  
  mensaje="";  
  
  tabnine: test | explain | document | ask  
  evento2(event:Event) {  
    this.mensaje = (<HTMLInputElement>event.target).value;  
  }  
}
```

2. Event binding

- Cuando escribimos ahora en el input text se ejecuta un evento (input) que llama al metodo EventoClick .
- Es necesario pasar el evento como parámetro para que TypeScript sea capaz de procesarlo.

