

# Angular

## Events & Binding



# Flujo de datos

---

Hay 3 formas

## 1. Javascript a HTML

Como venimos haciendo, por ejemplo un título o las cervezas.

## 2. HTML a Javascript

Como un mouse click, hover, input, key press.

## 3. De las dos formas (***Two-Way data binding***)

Por ejemplo un input que tiene que estar sincronizado con el dato del componente.



**Que creen que es el Event Binding?**



# Event Binding

---

El event binding es la forma de comunicar lo que pasa en HTML a Javascript.


- Mouse click,
- hover,
- etc.



# Agregar la cantidad a encargar

---

- Vamos a agregar dos botones para seleccionar la cantidad de cervezas que queremos.
- Vamos a mostrar cuantas cervezas de este tipo se están encargando

	Nombre	Tipo	Precio	Stock	
	Negra Fuerte	Porter	\$20.00	✓	- 0 +

Después vamos a poner un boton “agregar al carrito”.

# Que tenemos que modificar?

---

- Template (HTML)
- Component (beer-list)



# Actualizamos el Template

— — —



```
<td>
  <div class="form-group">
    <button type="button" class="btn btn-default btn-xs">
      <span class="glyphicon glyphicon-minus" aria-hidden="true"></span>
    </button>
    <span>{{beer.quantity}}</span>
    <button type="button" class="btn btn-default btn-xs">
      <span class="glyphicon glyphicon-plus" aria-hidden="true"></span>
    </button>
  </div>
</td>
```





# Función de incrementar quantity

- Creamos una función en el componente: `upQuantity(beer: Beer)`
- La bindeamos en el `click` de boton “+”

```
export class BeerListComponent implements OnInit {  
  beers : Beer[];  
  
  constructor() { }  
  
  ngOnInit() {  
    this.beers = BEERS;  
  }  
  
  upQuantity(beer){  
    beer.quantity++;  
  }  
}
```

```
<button type="button" class="btn btn-default btn-xs" (click)="upQuantity(beer)">  
  <span class="glyphicon glyphicon-plus" aria-hidden="true"></span>  
</button>  
</div>
```







**Cómo hacemos para decrementar la cantidad?**



# Función de decrementar quantity

---

- Creamos una función `downQuantity(beer: Beer);`
- La bindeamos en el click del boton “-”

```
<button type="button" class="btn btn-default btn-xs" (click)="downQuantity(beer)">  
  <span class="glyphicon glyphicon-minus" aria-hidden="true"></span>  
</button>
```

¿Dónde vamos almacenando la cantidad de cervezas elegida de cada tipo?

```
downQuantity(beer){  
  beer.quantity--;  
}
```



# Bugfixing



- No podemos incrementar más la cantidad de las que hay en stock.
- No podemos encargar menos de 0 cervezas.



# Two-Way Data Binding



---

Nos permite compartir información entre el componente y el template en ambas direcciones.

- Mantenemos sincronizados (sync) el template y el componente.
- Se utiliza mediante el uso de la sintaxis `[( )]`, conocida como “Banana in a Box”, Este tipo de binding se usa en inputs, donde queremos matener en sync el input con la propiedad en el componente.



# Importar Forms Module

---

src/app/app.module.ts

```
import { FormsModule } from '@angular/forms'
```

...

```
imports: [  
  BrowserModule,  
  AppRoutingModule,  
  FormsModule,  
],
```



# Usemos un input

---

- En lugar del span que estamos usando para mostrar la cantidad de cervezas que queremos comprar, usemos un input.
- Usando `[(ngModel)]="beer.quantity"` estamos usando two-way data binding con la **quantity** de las cervezas.

```
<span class="glyphicon glyphicon-minus" aria-hidden="true"></span>
</button>
<input class="smallInput" type="number" [(ngModel)]="beer.quantity"/>
<button type="button" class="btn btn-default btn-xs" (click)="upQuantity(beer)">
  <span class="glyphicon glyphicon-plus" aria-hidden="true"></span>
```



# Bugfixing



- Que pasa si ponemos un número muy grande en el input?
- Que pasa si ponemos un número negativo en el input?
- Como lo arreglamos?





# Obtener más datos del evento

---

## `$event`

- En caso de necesitar más datos del evento:
  - Que tecla se presionó?
  - Donde hizo click el usuario?
- Podemos pasarle a nuestra función el evento usando la variable **`$event`** desde el template.

*Note: Ese **`event`**, también se puede usar para llamar al `event.preventDefault()`*



# Debate

— — —

CSS y TS por componentes

Mezcla HTML y TS

Ventajas?

Desventajas?

# Referencias

— — —

- [Angular.io - Getting Started](#) - Documentación Oficial
- <https://angular.io/guide/two-way-binding>
- [Branch en el Repositorio](#) (2017, Angular 2)