



Desarrollo y Programación de  
Aplicaciones Avanzadas con Angular

Formación online



## Módulo 4. Data Binding

## Módulo 4. Data Binding

### Interpolation (interpolación)

La interpolación de texto nos permite incorporar valores de cadena dinámicos en sus plantillas HTML.

Sintaxis:

```
 {{title}}
```

Donde “title” es una propiedad del componente, que está declarada en el fichero .ts del mismo.

## Módulo 4. Data Binding

### Event binding

Podemos definir un evento click, con event binding, para invocar una función que esté definida en el componente.

Sintaxis:

`(click)=“metodoToCall()”`

# Módulo 4. Data Binding

## One-way data binding

**Property Binding** o **one-way data-binding** nos permite **vincular propiedades** de un elemento HTML a una variable de nuestro contexto de componente del .TS  
(Los inputs con los corchetes [] son property binding)

```
export class PruebaComponent {  
  imageUrl = 'https://google ...';  
  inputName = 'Valor por defecto!';  
}
```

TS Prueba Component

```
<img [src]="imageUrl">  
  ↴  
<input type="text" [value]="inputName">
```

HTML Prueba Component

MODEL



VIEW

# Módulo 4. Data Binding

## Two-way data binding

**Two-way data-binding** crea un **enlace de datos que sincroniza valores de nuestro componente .TS y nuestra vista .HTML** – permitiendo ver los cambios en tiempo real

Hay una directiva en Angular que implementa two-way data-binding -> **ngModel**

- ✓ Si el valor de la variable inputName cambia, la vista se actualizará reflejando el valor
- ✓ Si desde el navegador cambiamos el valor del input en la vista, el valor de la variable cambiará

```
<input type="text" [(ngModel)]="inputName">
```

MODEL

VIEW



## Módulo 4. Data Binding

# Property Binding / Event Binding



# Módulo 4. Data Binding

## Forms Template Driven

Este tipo de formularios en angular, hacen uso del Two-way data binding: <https://angular.io/guide/forms>

### MODELO

```
export class Hero {  
  
  constructor(  
    public id: number,  
    public name: string,  
    public power: string,  
    public alterEgo?: string  
  ) {}  
  
}
```

### DECLARACIÓN DE UN ELEMENTO

```
const myHero = new Hero(42, 'SkyDog',  
  'Fetch any object at any distance',  
  'Leslie Rollover');  
console.log('My hero is called ' + myHero.name); // "My hero is called SkyDog"
```

### BINDING

```
<input type="text" class="form-control" id="name"  
      required  
      [(ngModel)]="model.name" name="name">
```

# LABORATORIO: Formulario de login

Realizar una pantalla de inicio de sesión (LOGIN), haciendo uso de Angular Material. Dicha pantalla debe tener:

- Toolbar.
- Email: <https://material.angular.io/components/form-field/overview>
- Contraseña: <https://material.angular.io/components/form-field/overview>
- Botón de inicio de sesión: <https://material.angular.io/components/button/examples>

NgModel in forms: <https://angular.io/guide/forms>

SE PIDE:

Haciendo uso de ngModel (two ways data binding) guardar el email y contraseña escrito en el formulario de login para mostrarlo por consola (console.log()), al hacer click en el botón de Login.