



Desarrollo y Programación de
Aplicaciones Avanzadas con Angular

Formación online



Módulo 5. Angular Directives

Módulo 5. Angular Directives

Pipes

```
<p>The hero's birthday is {{ birthday | date:"MM/dd/yy" }} </p>
```

PIPES	DETAILS
DatePipe	Formats a date value according to locale rules.
UpperCasePipe	Transforms text to all upper case.
LowerCasePipe	Transforms text to all lower case.
CurrencyPipe	Transforms a number to a currency string, formatted according to locale rules.
DecimalPipe	Transforms a number into a string with a decimal point, formatted according to locale rules.
PercentPipe	Transforms a number to a percentage string, formatted according to locale rules.

Con un Pipe podemos transformar la salida de un dato mostrado en la vista del usuario.

Existe una lista de Pipes predefinidos, aunque también podemos crear nuestros propios pipes.

La sintaxis, es la que se muestra en el ejemplo.

<https://angular.io/guide/pipes>

Módulo 5. Angular Directives

Custom Pipe

```
import { Pipe, PipeTransform } from '@angular/core';

const apiKey = 'YOUR_API_KEY';

@Pipe({
  name: 'apiKey'
})
export class ApiKeyPipe implements PipeTransform {

  transform(value: string): string {
    return `${value}?api_key=${apiKey}`;
  }
}
```

Módulo 5. Angular Directives

Directivas estructurales y de estilo

Módulo 5. Angular Directives

ngIf

La directiva **ngIf**, permite determinar si un elemento sobre el que se define debe mostrar u ocultarse.

Sintaxis:

```
*ngIf="expression"
```

Puede ser que nos interese en algún caso, hacer uso de la directiva **ngIf**, sin que queramos que se incluyan ningún elemento HTML adicional en el DOM (Document Object Model). Para ello, hacemos uso de **<ng-container>**:

```
<p>
  I turned the corner
  <ng-container *ngIf="hero">
    and saw {{hero.name}}. I waved
  </ng-container>
  and continued on my way.
</p>
```

Módulo 5. Angular Directives

ngFor

- La directiva **ngFor**, nos permite dibujar *n veces* el elemento HTML sobre el que definamos la directiva. La sintaxis básica es:

```
<div *ngFor="let item of items">{{item.name}}</div>
```

- Donde **items** es un array de elementos a recorrer y definimos **let item**, como la variable que contiene el elemento actual del array que va a ser tratado.
- Existe la posibilidad de iterar sobre un componente hijo que queramos invocar:

```
<app-item-detail *ngFor="let item of items" [item]="item"></app-item-detail>
```

- En este caso, se utiliza el “item” actual, para introducirlo como parámetro @Input del componente app-item-detail. En el siguiente ejemplo, podemos ver de qué manera tener acceso al índice del elemento actual del array:

```
<div *ngFor="let item of items; let i=index">{{i + 1}} - {{item.name}}</div>
```

Módulo 5. Angular Directives

ngStyle

Set the font of the containing element to the result of an expression.

```
<some-element [ngStyle]="'font-style': styleExp">...</some-element>
```

Set the width of the containing element to a pixel value returned by an expression.

```
<some-element [ngStyle]="'max-width.px': widthExp">...</some-element>
```

Set a collection of style values using an expression that returns key-value pairs.

```
<some-element [ngStyle]="objExp">...</some-element>
```

Módulo 5. Angular Directives

ngClass

```
<some-element [ngClass]="'first second'">...</some-element>

<some-element [ngClass]=["'first', 'second' ]">...</some-element>

<some-element [ngClass]="{'first': true, 'second': true, 'third': false}">...</some-
element>

<some-element [ngClass]="stringExp|arrayExp|objExp">...</some-element>

<some-element [ngClass]="'class1 class2 class3' : true">...</some-element>
```

LABORATORIO: Listado de alumnos

EJERCICIO 1

- Crear una interface Alumno, que contenga nombre, apellidos, edad.
- Crear un componente *ListadoAlumnosComponent*.
- Declarar una lista de alumnos en el componente, con datos de ejemplo.
- Listar los elementos en el HTML haciendo uso de la directiva ngFor.
- Utilizar Angular Material para listar los elementos, haciendo uso por ejemplo del componente Card.

EJERCICIO 2

- Añadir a la interface Alumno la propiedad sexo (puede declararse con un Enum: hombre, mujer).
- Modificar el listado, para que con la directiva ngIf se muestra un icono si es hombre y otro distinto si es mujer en cada elemento.
- Modificar además el estilo de cada elemento haciendo que se muestre en algún color distinto aquellos que sean menores de edad (hacer uso de ngClass o ngStyle).