



Angular

10.1. Pipes

1. ¿Qué es un pipe?

- Permiten transformar los datos antes de mostrarlos por pantalla.
- Es una función que recibe unos datos de entrada y permite formatear los datos.
- Soporta varios tipos de datos: numbers, string, etc.

2. Data Binding

- **DatePipe:** Formatea un valor de fecha
- **UpperCasePipe:** transforma el texto a mayúsculas
- **LowerCasePipe:** transforma el texto a minúsculas
- **CurrencyPipe:** transforma el numero o string a moneda
- **DecimalPipe:** transforma un numero o string a decimal
- **PercentPipe:** transforma un numero en un porcentaje.
- ...

3. Usar un pipe en un template

- Para aplicar un pipe, use el operador de pipe (|) dentro de una expresión de plantilla.
- En primer lugar, importamos en `app.module.ts` las pipes que vamos a utilizar (con sus respectivos `import`)

3. Usar un pipe en un template

```
import { CurrencyPipe, DatePipe } from '@angular/common';
```

```
@NgModule({  
  declarations: [  
    AppComponent  
  ],  
  imports: [  
    BrowserModule,  
    AppRoutingModule,  
    DatePipe,  
    CurrencyPipe  
  ],  
})
```

3. Usar un pipe en un template

- Dentro de **app.component.ts** tenemos unos atributos que vamos a formatear, una fecha y una cantidad en dinero.

```
export class AppComponent {  
  cumple=new Date(1988,2,5);  
  dinero=23.23324;  
}
```

3. Usar un pipe en un template

- Dentro del template formateamos dichos datos.

Go to component

```
<p> La fecha de nacimiento de mi gato es: {{ cumple | date:"shortDate" }}</p>
<p> La fecha de nacimiento de mi gato es: {{ cumple | date:"mediumDate" }}</p>
<p> La fecha de nacimiento de mi gato es: {{ cumple | date:"fullDate" }}</p>

<p>Esta moneda esta en euros {{dinero | currency:'EUR':true }}</p>
<p>Esta moneda esta en euros {{dinero | number:'1.2-2'}} €</p>
```

3. Usar un pipe en un template

- Resultado:

