

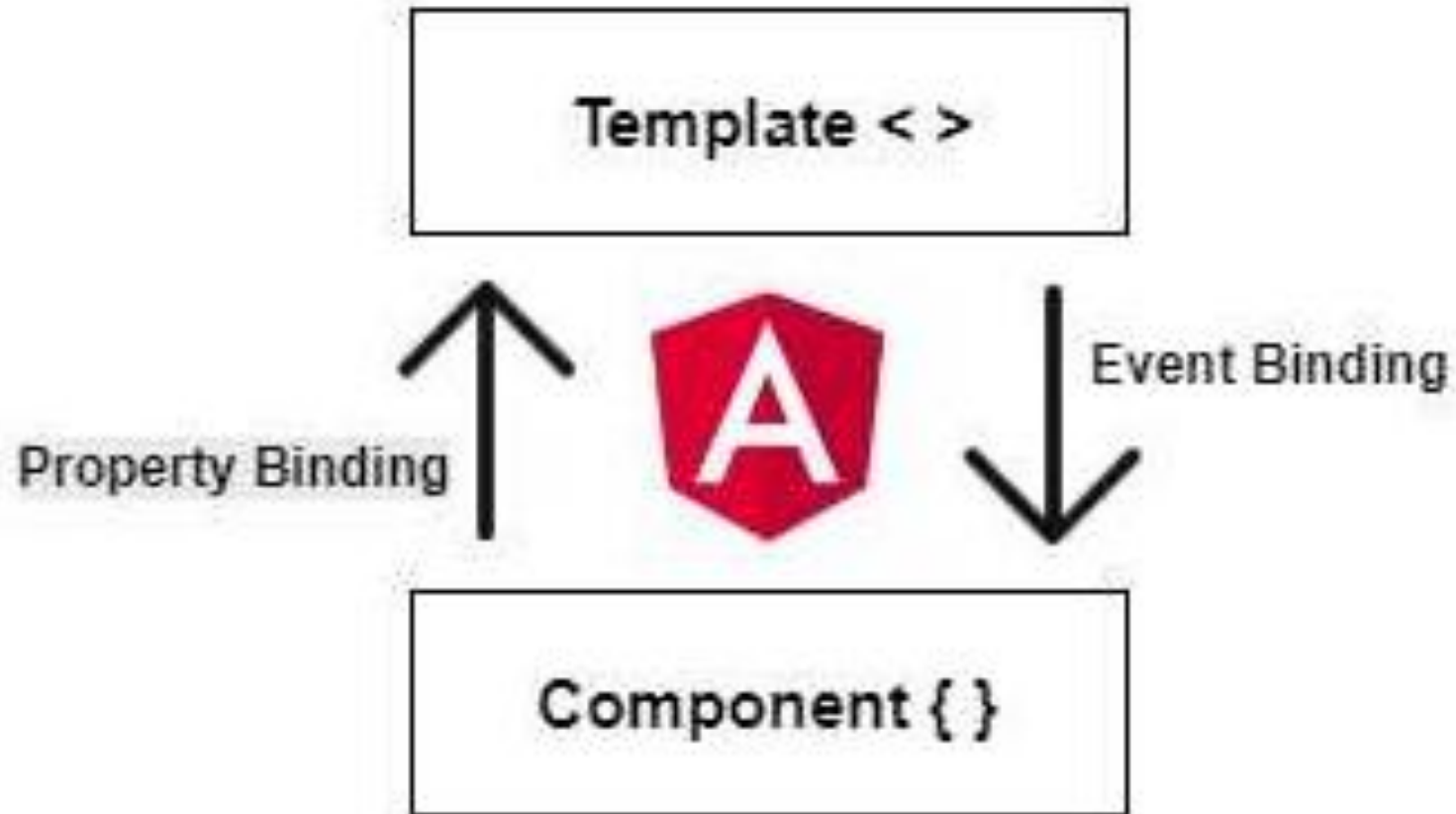
Angular

3.4. Two way data binding

1. Two way data binding

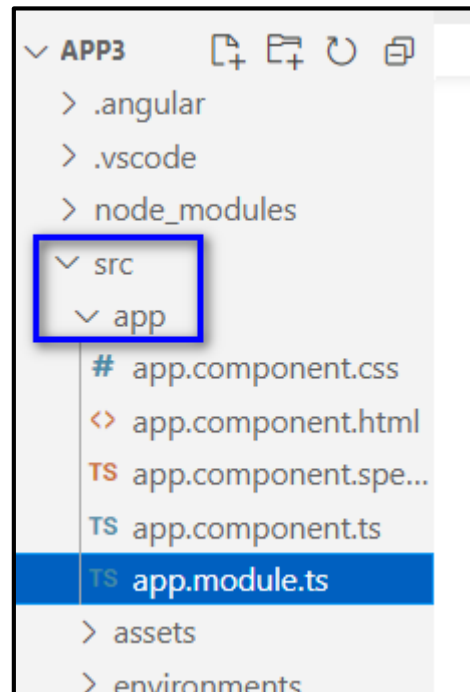
- Hasta ahora, hemos estado trabajando con comunicaciones/sincronizaciones (bindings) unidireccionales, es decir, de una sola dirección, los conocidos como One Way Data Binding.
- En los que nos traspasábamos/ comunicábamos/ sincronizábamos un objeto o una variable de nuestra Modelo, es decir, la lógica de negocio del componente situado en nuestro código de TypeScript hacía el template (el HTML) o viceversa.

1. Two way data binding



1. Importar FormsModule

- En primer lugar nos dirigimos al fichero **app.module.ts** que está dentro de **src/app**



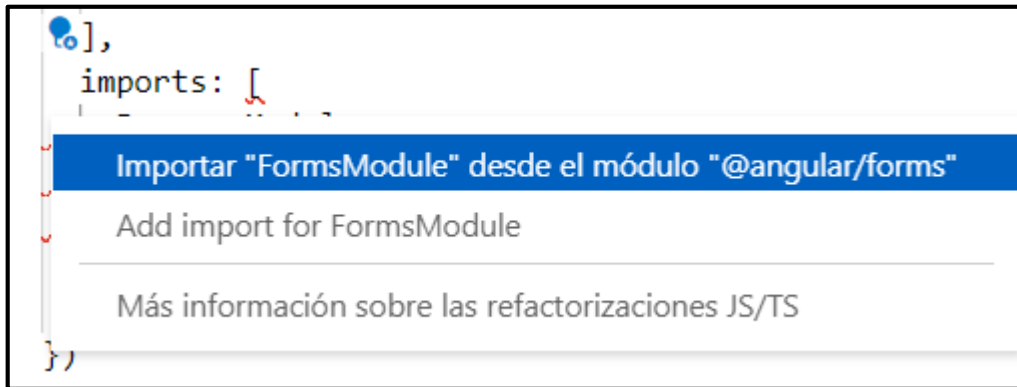
1. Importar FormsModule

- Dentro del array imports, añadimos el modulo **FormsModule**

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

1. Importar FormsModule

- Si muestra un error y una sugerencia, es porque no hemos importado aun dicho modulo en la cabecera (imports ...)
- Si hacemos caso a la bombilla nos importará dicha clase.



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

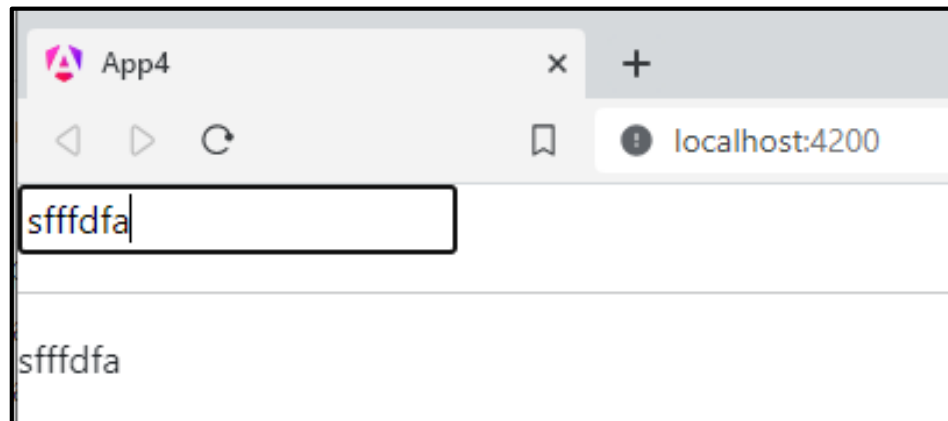
2. Componente HTML y TS

```
<> app.component.html X
src > app > <> app.component.html > ...
  Go to component
1  <input type="text" name="texto" [(ngModel)]="titulo">
2  <hr/>
3  {{ titulo }} |
```

```
export class AppComponent {
  title = 'app4';
  titulo="";
}
```

2. Componente HTML y TS

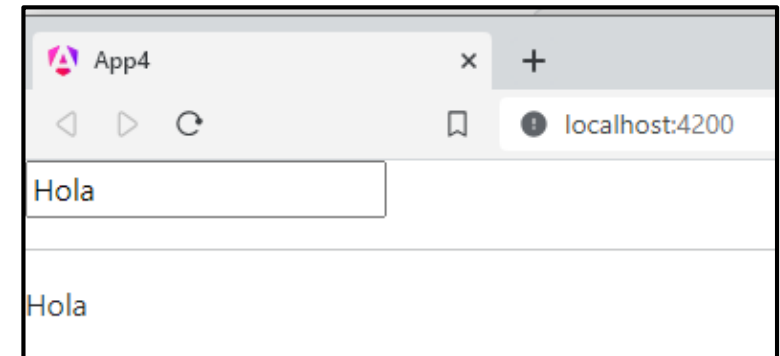
- Con **[(ngModel)]** establecemos una doble conexión entre el formulario y el código del componente.
- Por tanto ese, campo podrá ser modificado tanto por la parte HTML como por la parte TS



3. Verificar la bidireccionalidad

- Podemos inicializar el valor de titulo al declarar la clase y veremos como se modifica también en el formulario.

```
export class AppComponent {  
  title = 'app4';  
  titulo="Hola";  
}
```



4. Ejemplo: Calculadora

- Generamos un nuevo proyecto (acordarse del -- standalone=false):

```
/angular>ng new twodata --standalone=false
```

4. Ejemplo: Calculadora

- Hay que instalar el modulo **FormsModule** en app.module.ts

```
4 import { AppRoutingModule } from './app-routing.module';  
5 import { AppComponent } from './app.component';  
6 import { FormsModule } from '@angular/forms';  
7  
8 @NgModule({  
9   declarations: [  
10    | AppComponent  
11   ],  
12   imports: [  
13    | BrowserModule,  
14    | AppRoutingModule,  
15    | FormsModule  
16  ],
```

4. Ejemplo: Calculadora

- En el componente vamos a tener 3 atributos: numero1, numero2 y resultado

```
//  
export class AppComponent {  
  numero1=0;  
  numero2=0;  
  resultado=0;
```

4. Ejemplo: Calculadora

- Tendremos una pagina HTML con donde vamos a enlazar estos 3 números y vamos a tener las operaciones básicas: suma, resta, multiplicación y división:

4. Ejemplo: Calculadora

Go to component

```
<input type="number" name="numero1" [(ngModel)]="numero1" />
<button (click)="sumar()">+</button>
<button (click)="restar()">-</button>
<button (click)="multiplicar()">*</button>
<button (click)="dividir()">/</button>
<input type="number" name="numero2" [(ngModel)]="numero2" />
<label>=</label>
<input type="number" name="resultado" [(ngModel)]="resultado" readonly />
```

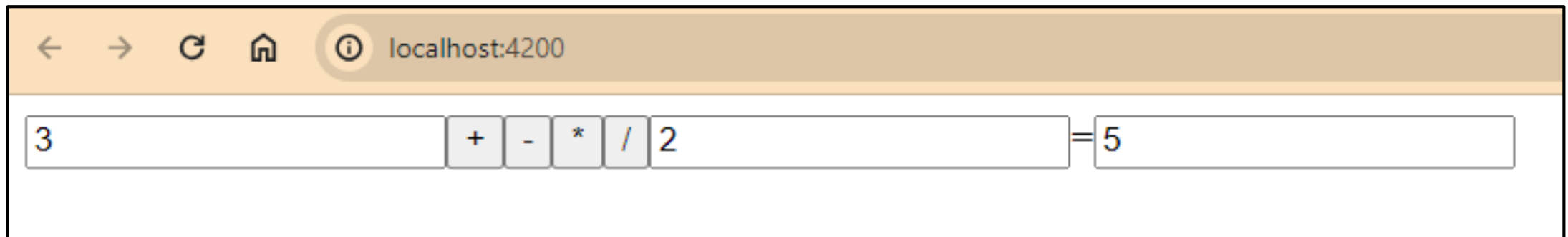
4. Ejemplo: Calculadora

- Como podemos observar esos botones tienen asociadas una serie de métodos

```
sumar() {  
    this.resultado = this.numero1 + this.numero2;  
}  
restar() {  
    this.resultado = this.numero1 - this.numero2;  
}  
multiplicar() {  
    this.resultado = this.numero1 * this.numero2;  
}  
dividir() {  
    this.resultado = this.numero1 / this.numero2;  
}
```

4. Ejemplo: Calculadora

- Al estar doblemente enlazados, cualquier cambio de los valores en el documento HTML se ve reflejado en el atributo y viceversa.



A screenshot of a web browser window. The address bar shows 'localhost:4200'. The main content area displays a calculator interface with a row of input fields and operators. The first input field contains the number '3'. This is followed by four operator buttons: '+', '-', '*', and '/'. The fifth input field contains the number '2'. To the right of these is an equals sign '=' followed by a final input field containing the number '5'.