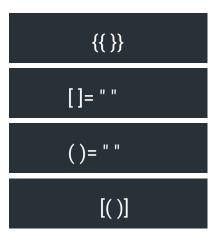


## Sintaxis de enlace (data binding)

Data binding se puede definir como la comunicación de valores entre la fuente de datos (clase del componente) y el DOM (plantilla).

- Interpolación (Hacia el DOM)
- Property binding: (Hacia el DOM)
- Event binding: (Desde el DOM)
- Two-way binding: (Desde/Hacia el DOM)





## Interpolación (Hacia el DOM)

{{ template-expression JavaScript }}

### Template expression

- Angular las resuelve y devuelve un string.
- No modifican el estado.
- No soportan:
  - Operadores de asignación (=, +=, -=, ...), new, typeof, instanceof, etc
  - Expresiones encadenadas con ; o ,
  - Operadores incrementales (++ o --)
  - Algunos operadores ES2015+
  - Operadores lógicos como | y &



## Property binding (Hacia el DOM)

#### [propiedadDOM]="template-expression JavaScript"

- En teoría enlaza propiedades del DOM con el valor que devuelve la expresión, aunque los atributos html también los admite puesto que se suelen denominar igual.
- En los casos de uso también se podría utilizar interpolación a un atributo, por ejemplo href="{{url}}", pero siempre que el atributo reciba string, en caso contrario es preferible utilizar property binding.



# Event binding (Desde el DOM)

(eventoDOM)="template-stament"

Eventos DOM: <a href="https://developer.mozilla.org/en-US/docs/Web/Events">https://developer.mozilla.org/en-US/docs/Web/Events</a>

### Template-stament

- Pueden modificar el estado.
- No soportan:
  - Operadores incrementales (++ o --)
  - o new, typeof, instanceof, etc
  - Algunos operadores ES2015+
  - Operadores lógicos como | y &



# Two-way binding (Desde/Hacia el DOM)

```
[(ngModel)]="propiedad"
```

Exige la importación de la librería @angular/forms en el módulo:

```
import { FormsModule } from '@angular/forms';
...
imports: [
    ...
    FormsModule,
    ...
],
```

