

# Property Binding en Angular



## Property Binding en Angular

**Property Binding** en Angular es una técnica que permite enlazar valores de propiedades del componente a propiedades de elementos HTML en la vista. Esto se logra utilizando la sintaxis de corchetes `[],` que enlaza la propiedad de un elemento o componente a una expresión de Angular, permitiendo que los cambios en el valor de la propiedad en el componente se reflejen automáticamente en la vista.



## Sintaxis de Property Binding

La sintaxis básica para el property binding es:

```
[HTML-Propiedad]="expresión"
```

Donde:

- `HTML-Propiedad` es la propiedad del elemento HTML o componente que deseas enlazar.
- `expresión` es una propiedad del componente, una variable, o una expresión que se evaluará en tiempo de ejecución.

## Ejemplo Básico de Property Binding

Veamos un ejemplo sencillo de un replicador utilizando **property binding** en Angular. Este replicador simplemente copiará el texto que escribas en un campo de entrada y lo mostrará en otro campo de texto de sólo lectura de manera automática.

### Paso 1: Crear el Componente `ReplicadorComponent`

Primero, vamos a crear un componente llamado `ReplicadorComponent`.

```
ng generate component replicador --skip-tests
```

Código para `replicador.component.html`:

```
<div class="d-flex justify-content-center">
  <div class="w-50">
    <div class="mb-3">
      <label for="inputTexto">Escribe algo:</label>
      <input id="inputTexto" type="text" (input)="actualizarTexto($event)" />
    </div>
    <div class="mb-3">
      <label for="replicaTexto">Texto replicado:</label>
      <input id="replicaTexto" type="text" [value]="texto" readonly />
    </div>
  </div>
</div>
```

### Explicación de Código:

- **input**: Es un evento del DOM que se dispara cuando cambia el valor de un `<input>`.
- **\$event**: Es la variable que contiene el objeto del evento, proporcionando detalles sobre la interacción del usuario.
- **(input)="actualizarTexto(\$event)"**: Enlaza el evento input al método `actualizarTexto`, pasando el evento como argumento.
- En el `input id="replicaTexto"` es un ejemplo de **property binding** que sincroniza el valor de un campo de entrada (`value`) con una propiedad del componente Angular (`texto`). El atributo `readonly` asegura que este valor no pueda ser modificado.

directamente por el usuario, pero sí puede ser actualizado por el componente. Este es el concepto de property binding aplicado.

Código para `replicador.component.ts`:

```
import { Component } from '@angular/core';

@Component({
  selector: 'app-replicador',
  standalone: true,
  imports: [],
  templateUrl: './replicador.component.html',
  styleUrls: ['./replicador.component.css']
})
export class ReplicadorComponent {
  texto: string = '';

  actualizarTexto(event: Event): void {
    const elementoInput = event.target as HTMLInputElement;
    this.texto = elementoInput.value;
  }
}
```

### Explicación del código anterior:

El parámetro `event` en el método `actualizarTexto` es el objeto que representa el evento `input` del DOM. Se utiliza para acceder al elemento que disparó el evento (a través de `event.target`) y obtener su valor actual, permitiendo que el componente reaccione de manera adecuada a las interacciones del usuario.

### Actualizar el componente AppComponent

Finalmente, puedes utilizar `ReplicadorComponent` en tu aplicación, por ejemplo, en `app.component.ts`.

Código para `app.component.ts`:

```
import { Component } from '@angular/core';
import { RouterOutlet } from '@angular/router';
import { NuevoComponenteComponent } from '../nuevo-componente/nuevo-componente.component';
import { ComponenteInlineComponent } from '../componente-inline/componente-inline.component';
import { PadreComponent } from '../padre/padre.component';
import { MostrarMensajeComponent } from '../mostrar-mensaje/mostrar-mensaje.component';
```

```
import { ReplicadorComponent } from "../replicador/replicador.component";

@Component({
  selector: 'app-root',
  standalone: true,
  imports: [
    RouterOutlet,
    NuevoComponenteComponent,
    ComponenteInlineComponent,
    PadreComponent,
    MostrarMensajeComponent,
    ReplicadorComponent
  ],
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css'],
})
export class AppComponent {
  titulo = 'Property Binding en Angular';
}
```

Código para `app.component.html`:

```
<section>
  <div class="container text-center">
    <h1 class="text-center text-warning my-4">{{ titulo }}</h1>
    <app-replicador/>
  </div>
</section>
```

## Resultado Esperado

Cuando ejecutes la aplicación (`ng serve`), verás lo siguiente:

1. Un campo de entrada donde puedes escribir texto.
2. El texto que escribas en el campo de entrada se replicará y mostrará en un párrafo debajo de él, en tiempo real, a medida que escribes.



## Resumen

Este ejemplo muestra cómo usar el concepto de Property Binding junto con Event Binding para crear un replicador simple en Angular. Este es un ejemplo básico y directo que ilustra cómo funcionan estos conceptos fundamentales en Angular.

Saludos!

Ing. Ubaldo Acosta

Fundador de [GlobalMentoring.com.mx](http://www.globalmentoring.com.mx)