Lección 2: Componentes y Directivas





Lección 2: Componentes y Directivas

Lectura: Databinding

Databinding



Una de las características principales en el desarrollo de aplicaciones usando Angular 2 es el concepto de databinding. Este término puede traducirse como enlace de datos y hace referencia a la comunicación que existe entre la vista y la lógica en una aplicación, es decir, la manera en la que se pasan datos desde las clases TypeScript a las plantillas HTML de los componentes.

Existen 4 tipos de databinding:

- String Interpolation
- Property Binding
- Event Binding
- Two-way Binding

String Interpolation



La interpolación de cadenas de caracteres, o como mejor se conoce por los desarrolladores que usan este framework string interpolation, es una forma de comunicación entre la lógica en la vista Lección 2: Componentes y Directivas para pasar valores de tipo string o que pueden ser convertidos a string. A continuación, verás una clase en TypeScript correspondiente a un componente, en ella se define una variable con el nombre de **nombreUsuario**, y otra con el nombre de **edadUsuario**, y se le asignan valores a ambas.

```
app.component.ts x

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

import { BarraSuperiorComponent } from './barra-superior/barra-superior.component';

@Component({
    selector: 'app-root',
    templateUrl: './app.component.html',
    styleUrls: ['./app.component.css']
    })

export class AppComponent {
    title = 'Hola mundo!';
    nombreUsuario = 'Carlos Fernandez';
    edadUsuario = 35;
}

14
}
```

Para realizar una interpolación de cadena se usa el operador doble llave y, en su interior, el nombre de la variable que queremos asignar a esa parte de la plantilla.

```
{{ variable }}
```

Ahora veamos la plantilla HTML correspondiente a este componente. Hemos creado un párrafo y, mediante el operador de doble llave, hemos indicado que deben interpolarse las variables **nombreUsuario** y **edadUsuario**.

Si vemos el resultado en el navegador, observamos que las variables se han interpolado y hemos pasado a la vista un valor asignado desde la clase TypeScript.

Hola mundo!

Mi nombre es Carlos Fernandez y tengo 35 años.

Lección 2: Componentes y Directivas



Property Binding:

El enlace de propiedades, o property binding, hace referencia a pasar datos desde la clase TypeScript de un componente a una plantilla HTML para asignar propiedades a los elementos de dicha plantilla. Funciona de manera muy similar a la interpolación de cadenas, con la diferencia que ésta se enfoca sólo en propiedades de los elementos HTML. Si añadimos, por ejemplo, un elemento de tipo input a la plantilla del componente app, podemos hacer un enlace de propiedad usando el operador corchete:

```
[nombre_propiedad] = "variable o expresión"
```

Si vemos el resultado en el navegador, podemos observar que la propiedad value del campo de texto tomó el valor de la variable **nombreUsuario** asignada en la clase TypeScript del componente.

Hola mundo!

Mi nombre es Carlos Fernandez y tengo 35 años.

Carlos Fernandez

Event binding:

La asignación de eventos, o event binding como es mejor conocic asignan eventos a los elementos de las plantillas HTML de los co



métodos que deben disparar dichos eventos. Vamos a definir primero un método en la clase del componente app que muestre una alerta indicando que se disparó el evento click. \leftarrow \rightarrow

Ahora vamos a la plantilla HTML de este componente y realicemos un enlace de evento. Para enlazar un evento se usa el operador paréntesis para encerrar el nombre del evento que dicho elemento va a disparar, y se hace igual al método que debe ejecutarse al dispararse el evento indicado.

```
(nombre_evento) = "método a ejecutar"
```

Por ejemplo, al elemento h1 de nuestra plantilla HTML vamos a asignarle un evento click, y lo hacemos igual al método que acabamos de crear: titleClicked().

Si probamos esto en el navegador, observamos que se ha asignado el evento y se dispara el método que creamos en la clase del componente.



Lección 2: Componentes y Directivas Two-way Binding:



El enlace bidireccional, o two-way binding como es conocido en Angular 2, hace referencia a una combinación de los dos tipos de Databinding vistos previamente: property binding y event binding. El concepto bidireccional indica que se envían datos desde la clase TypeScript y de la misma manera se reciben desde la plantilla HTML. Para definir un enlace bidireccional se deben usar, tanto los corchetes como los paréntesis, y al interior usar la palabra NgModel. A continuación, se debe asignar una propiedad de algún objeto que interactúe con la asignación:

```
[(NgModel)] = "objeto.propiedad"
```

Añadimos a continuación un objeto a la clase del componente llamado persona y le definimos dos atributos, nombre y edad.

```
app.component.ts
import { Component } from '@angular/core';
import { BarraSuperiorComponent } from './barra-superior/barra-superior.component';

@Component({
    selector: 'app-root',
    templateUrl: './app.component.html',
    styleUrls: ['./app.component.css']
    })
    export class AppComponent {
    title = 'Hola mundo!';
    nombreUsuario = 'Carlos Fernandez';
    edadUsuario = 35;

    titleClicked(){
        alert('Se ha disparado un evento click');
    }
    persona = {
        nombre: 'José',
        edad: 22
    }
}
```

Ahora agregaremos dos elementos de tipo input a la plantilla HTML y en sus propiedades añadimos la especificación de un enlace bidireccional.

```
app.component.html x mapp.module.ts app.component.css map.component.html x mapp.module.ts app.component.css map.component.css app.component.css map.component.css map.component.css app.component.css map.component.css app.component.css app.componen
```

Si probamos esto en el navegador, podremos observar que, al modificar el valor de una de los dos campos de texto, modifica el valor del otro, ya que un evento se **¡Chatea con un tutor!**

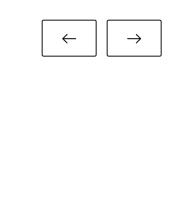
vista.

Lección 2: Componentes y Directivas

Hola mundo!

 $\,$ Mi nombre es Carlos Fernandez y tengo 35 años.

José Pe



Lección 2: Componentes y Directivas

 \leftarrow

 \rightarrow