ANGULAR NOTES

By: Deepa Chaurasia

https://www.linkedin.com/in/deepa-chaurasia-3704351a8

Page
Date

What is Data Binding?

→ Communication between typescript code of your component and the template (html) that the user see.

Data Binding == Communication

We get different ways of communication.

For eg → we want to output data from our typescript code in html code.
we can use String Interpolation for this

Syntax for String Interpolation

{{ data }}

you put your data that will come from ts in double curly braces

Page
Date

Eg of String Interpolation

Eg → In your ts file.

```
export class MyComponent {
  name : string = 'deepa';
}
```

In your html file

```
<p> My name is {{ name }} </p>
```

// will print My name is deepa.

⇒ Now Suppose you have a button to submit, It must be enabled only if you're admin.

In this case you'll use property binding

☆ The property [disabled] will bind to native boolean which we'll define in ts.

Eg of Property binding

In ts File.

```
export class MyComponent {
  isAdmin : boolean = true;
}
```

In html

```
<button [disabled] = "!isAdmin"> Add </button>
```

// if not admin then disable

Event Binding → It allows your component to react to user's actions such as button click, key strokes, etc.

Like property Binding use [] square brackets

Event Binding use () paranthesis

Let's say on click of button I want to show something.

In ts File

```
export class MyComponent {
  name : String ;
  onClick() {
    this.name = 'deepa';
  }
}
```

In html

```
<p> My name is {{name}} </p>
<button (click) = "onClick()" Add </button>
```

So before clicking Add, name will be blank
After clicking Add, name will be shown (i.e. deepa)

Page
Date

What are Directives?

Directives are instruction in the DOM!

Basically these are custom html attributes which tell angular to change the style or behaviour of Dom elements.

Components are building blocks of Angular applications, Similarly directives are also building block of Angular Applications.

Types of Directives

Component	Structural Directives	Attribute Directives
Directives with a template	Change layout of the elements	Change appearance or behaviour of a particular element

1. Component / Directives

* Directives must be declared in Angular Modules in the same manner as components.

* Directive in Angular is a re-usable component.

* The other 2 directives such as structual and attribute do not have templates

It is decorated with @ Component decorator

Example :- change-text directive

change-text.directive.ts

```
import { directive, ElementRef } from '@angular/core'

@Directive ({
    selector : '[changeText]'
})

export class ChangeTextDirective {
    constructor (Element : ElementRef) {
        Element.native Element.innesText =
            "Text is changed";
    }
}
```

app.component.html

<span changeText> Welcome

In above eg, there is class called ChangeTextDirective and a constructor, which takes the element of type ElementRef, which is mandatory.

The element has all the details to which change Directive is applied.

2. Structural Directives :

* Structural Directives are responsible for changing structure of DOM
* They work by adding or removing element from DOM.
* The Structural directive always starts with '*'

Three most popular Structural Directive

1) ngIf
2) ngFor
3) ngSwitch

All of them work same as If, for or switch respectively.

Except that you use it for DOM Tree in html template

Eg →

```
<div *ngIf = "info" class = "name" >
    {{ info.name }} </div>

<div *ngFor = "let movie of movies" >
    {{movie.name }} </div>
```

3) Attribute Directives

Attribute Directives manipulate DOM by changing its behaviour and appearance.

* It is used to show or hide elements or dynamically change behaviour of component.

* Built in Attributes are ngStyle, ngClass etc.

Pipes In Angular

What are Pipes?

Pipes are a feature built into Angular 2 which allows you to transform output in your template

Eg. you have a property username = 'deepa'
You want to output username in your template using string Interpolation

like {{ username }}

Now you want your name to be In Upper Case only in Output on template

You don't wanna change property in your ts you want to transform the way it is displayed on screen

For this you could simply use uppercase pipe which is a built-in pipe

Like this {{ username | uppercase }}

Now Output → DEEPA

Parametrizing Pipes

Suppose you have date in a format
8 April, 2022

You don't want this format, you want to customize it

You can do it using parametrized pipes

ts file

```
students = [
{ name : 'deepa',
  class : '10',
  dob : new Date (14, 12, 2999)
}
```

Now In html

```
<p> {{ students.dob }} </p>

<p> {{ students.dob | date : 'fullDate' }} </p>
```

You can refer angular.io to know more about built in pipes

Chaining Multiple Pipes

```
<p> {{ students.dob | date : 'fullDate' | uppercase }}
</p>
```

Here order in which we write pipe matters because it iterates from left to right

* first fullDate pipe will apply
* on the result uppercase will apply

Creating a Custom Pipe.

* Create ts file shortenPipe.ts (you can use any name instead of shorten)
* your pipe.ts should look like this

imp. → `import { Pipe, PipeTransform } from 'angular/core';`

`@Pipe({name : "myPipe"})` → Interface it implement

* compulsory `export class MyPipe implements PipeTransform {`

* method in which all operation happens ← `transform (value : any) {`

`return value.— ;`

`}`

`}`

In html → `<p> {{ students.name | myPipe }} </p>`

Thankyou For Reading

Like,Share and Comment

https://www.linkedin.com/in/deepa-chaurasia-notes