Data Binding Methods (Interpolation / Property binding / Attribute binding / class binding / style binding / event binding)

App.component.html

<!--The content below is only a placeholder and can be replaced.-->

<div style="text-align:center">

<h1> WELL DONE </h1><br>

<h1> Welcome to {{title}}! </h1>

<!-- -->

<button disabled='{{isDisabled1}}'>Interpolation</button>

<!-- When setting an element property to a non-string

data value, you must use property Binding-->

<!-- -->

<button [disabled]='isDisabled2'>Property Binding</button>

<br><br>

<table border={{borDer}}>

<tr>

<!-- <th colspan=2> -->

<!-- <th colspan={{colSpan}} > -->

<!-- <th [colspan]= colSpan> -->

<th [attr.colspan]= colSpan> Employee Details </th>

</tr>

<tr>

<td>First Name</td>

<td>{{firstName}}</td>

</tr>

<tr>

<td>Last Name</td>

<td>{{lastName}}</td>

</tr>

<tr>

<td>Gender</td>

<td>{{gender}}</td>

</tr>

<tr>

<td>Age</td>

<td>{{age}}</td>

</tr>

</table>

</div>

App.component.ts

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

@Component({

selector: 'app-root',

// templateUrl: './app.component.html',

//template:`<button class='colorClass'>My Button</button>`,

//CSS Class binding in Angular

// template:`

// <button [class]='classesToApply'>My Button</button>

// `

//style Binding

//template: `<button style="color:red">My Button</button>`

// template: `<button style='color:red'

// [style.font-weight]="isBold ? 'bold' : 'normal'">My Button

// </button> `

//To set multiple inline styles use NgStyle directive

template: `<button style='color:red' [ngStyle]="addStyles()">My Button</button>`

//template:`<h1>{{'Name = ' + firstName}}</h1>`,

//template: `<h1>{{ firstName ? firstName : 'No name specified'}}</h1>`,

})

export class AppComponent {

isBold: boolean = true;

fontSize: number = 30;

isItalic: boolean = false;

addStyles() {

let styles = {

'font-weight': this.isBold ? 'bold' : 'normal',

'font-style': this.isItalic ? 'italic' : 'normal',

'font-size.px': this.fontSize };

return styles;

}

title:string = 'YASH';

firstName:string = 'Ram';

lastName:string = 'Lakhan';

gender:string = 'Male';

age:string = '25';

colSpan:number = 2

borDer:number= 1

isDisabled1:boolean = true;

isDisabled2:boolean = true;

classesToApply: string = 'colorClass italicsClass boldClass';

}

EventBindingComponent.ts

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

@Component({

selector: 'app-root',

template: `<button (click)='onClick()' >Click me</button>

<button on-click='onClick()'>Click me</button>`

// With event binding we can also use the on- prefix alternative as shown below. This is known as the canonical form

// template: `<button on-click='onClick()'>Click me</button>`

})

export class EventBindingComponent {

onClick(): void {

console.log('Button Clicked');

}

}

DirectiveDemo.ts

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

@Component({

selector: 'app-root',

styles:

['.firstClass { color:red; }',

'.secondClass { font-weight:bold; }',

'.thirdClass { text-decoration: underline; }'],

template: `

<!-- NgIf -->

<p \*ngIf='ifvar; then trueBlock else falseBlock'></p>

<ng-template #trueBlock>

<label>ifvar was set to true.</label>

</ng-template>

<ng-template #falseBlock>

<label>ifvar was set to false.</label>

</ng-template>

<!-- NgFor -->

<p \*ngFor='let element of forArray; let ind = index'>

NgFor's element {{ ind }} = {{ element }}

</p>

<!-- NgSwitch, NgSwitchCase, NgSwitchDefault -->

<div [ngSwitch]='switchVar'>

<p \*ngSwitchCase='1'>The NgSwitch value is Choice 1</p>

<p \*ngSwitchCase='2'>The NgSwitch value is Choice 2</p>

<p \*ngSwitchCase='3'>The NgSwitch value is Choice 3</p>

<p \*ngSwitchDefault>Another choice was selected</p>

</div>

<!-- NgClass -->

<div [ngClass]='classArray'>

<p>This element is in firstClass and secondClass</p>

<p [ngClass]='addClass'>

This element is in firstClass, secondClass, and thirdClass

</p>

</div>

<!-- NgStyle -->

<p [ngStyle]="{'color': fontColor, 'font-size': fontSize}">

This style is set by NgStyle.

</p>

`})

export class DirectiveDemo {

public ifvar = true;

public forArray = ['A', 'B', 'C', 'D'];

public switchVar = 22;

public classArray = ['firstClass', 'secondClass'];

public addClass = {'thirdClass' : true};

public fontColor = 'green';

public fontSize = '18px';

}

Structural directives( \*ngIf / \*ngFor / trackBy) & Interface

IEmployee.ts

export interface IEmployee {

code: string;

name: string;

gender: string;

annualSalary: number;

dateOfBirth: string;

}

AppComponent.ts

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

import { IEmployee } from './IEmployee';

@Component({

selector: 'list-employee',

templateUrl: './app.component.html',

styleUrls: ['./app.component.css']

})

export class AppComponent {

title: string = 'YASH';

employeerecords: IEmployee[] = [

{

code: 'emp101', name: 'Tom', gender: 'Male',

annualSalary: 5500, dateOfBirth: '25/6/1988'

},

{

code: 'emp102', name: 'Alex', gender: 'Male',

annualSalary: 5700.95, dateOfBirth: '9/6/1982'

},

{

code: 'emp103', name: 'Mike', gender: 'Male',

annualSalary: 5900, dateOfBirth: '12/8/1979'

},

{

code: 'emp104', name: 'Mary', gender: 'Female',

annualSalary: 6500.826, dateOfBirth: '14/10/1980'

},

];

getEmployees(): void {

this.employeerecords = [

{

code: 'emp101', name: 'Tom', gender: 'Male',

annualSalary: 5500, dateOfBirth: '25/6/1988'

},

{

code: 'emp102', name: 'Alex', gender: 'Male',

annualSalary: 5700.95, dateOfBirth: '9/6/1982'

},

{

code: 'emp103', name: 'Mike', gender: 'Male',

annualSalary: 5900, dateOfBirth: '12/8/1979'

},

{

code: 'emp104', name: 'John', gender: 'Female',

annualSalary: 6500.826, dateOfBirth: '14/10/1980'

},

{

code: 'emp105', name: 'Nancy', gender: 'Female',

annualSalary: 6700.826, dateOfBirth: '15/12/1982'

},

];

}

trackByEmpCode(index: number, employee: IEmployee): string {

return employee.code;

}

}

App.component.html

<!--The content below is only a placeholder and can be replaced.-->

<div style="text-align:center">

<h1>

Welcome to {{title}}!

</h1>

</div>

<table>

<thead>

<tr>

<th>Code</th>

<th>Name</th>

<th>Gender</th>

<th>Annual Salary</th>

<th>Date of Birth</th>

</tr>

</thead>

<tbody>

<tr \*ngFor='let employee of employeerecords; trackBy:trackByEmpCode'>

<!-- <tr \*ngFor='let employee of employeerecords'> -->

<td>{{employee.code}}</td>

<td>{{employee.name}}</td>

<td>{{employee.gender}}</td>

<td>{{employee.annualSalary}}</td>

<td>{{employee.dateOfBirth}}</td>

</tr>

<tr \*ngIf="!employeerecords || employeerecords.length==0">

<td colspan="5">

No Employee Records to Display

</td>

</tr>

</tbody>

</table>

<br />

<button (click)='getEmployees()'>Refresh Employees</button>

Two way data binding

AppModule.ts

import { BrowserModule } from '@angular/platform-browser';

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

import { AppComponent } from './app.component';

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

@NgModule({

declarations: [ AppComponent ],

imports: [ BrowserModule, FormsModule ],

providers: [],

bootstrap: [AppComponent]

})

export class AppModule { }

app.component.ts

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

@Component({

selector: 'app-root',

template: ` <!--Name : <input [value]='name'><br>

You entered : {{name}}<br> -->

Name : <input [value]='name' (input)='name = $event.target.value'><br>

You entered : {{name}}<br>

Name : <input [(ngModel)]='name'><br>

You entered : {{name}}<br> `

})

export class AppComponent {

name: string = 'Tom';

}