Components. ?

Code :- [event-binding] -> [event-binding.component.html]

<p>event-binding works!</p>

<div>

    <h2>One way data binding -> Use Event Binding</h2>

    <input type="button" value="Click Here" (click)="fun()">

    <br>

    {{msg}}

</div>

[event-binding] -> [event-binding.component.ts]

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

@Component({

  selector: 'app-event-binding',

  templateUrl: './event-binding.component.html',

  styleUrls: ['./event-binding.component.css']

})

export class EventBindingComponent implements OnInit {

  msg:String = "Hi";

  constructor() { }

  ngOnInit(): void {

  }

  fun(){

    //alert("Click On Event")

    this.msg = "Hello";

  }

}

Property Binding. ?

Code :- [property-binding] -> [property-binding.component.html] :-

<p>property-binding works!</p>

<div>

    <h1>One way data binding using -> Property Binding</h1>

    <input type="text"><br>

    <input type="text" value="Mohit"><br>

    <input type="text" value="lname"><br>

    <input type="text" [value]="lname"><br>

    <p [innerText]="lname"></p>

    <input type="button" value="Property Binding" [disabled]=f2><br>

</div>

[property-binding] -> [property-binding.component.ts] :-

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

@Component({

  selector: 'app-property-binding',

  templateUrl: './property-binding.component.html',

  styleUrls: ['./property-binding.component.css']

})

export class PropertyBindingComponent implements OnInit {

  lname:String = "Sharma";

  f1:boolean = true;

  f2:boolean = false;

  constructor() { }

  ngOnInit(): void {

  }

}

Class and Style Bindings. ?

Code :- [style-binding] -> [style-binding.component.ts] :-

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

@Component({

  selector: 'app-style-binding',

  templateUrl: './style-binding.component.html',

  styleUrls: ['./style-binding.component.css']

})

export class StyleBindingComponent implements OnInit {

  mark = "Style\_Binding";

  h = document.getElementsByClassName(this.mark);

  constructor() { }

  ngOnInit(): void {

  }

}

[style-binding] -> [style-binding.component.html] :-

<p>style-binding works!</p>

<h1 [class] = "mark">Hello Style ...!</h1>

<br>

Style Name : "{{ h[0].className }}"

Event Binding.

Code :- [event-binding] -> [event-binding.component.html]

<p>event-binding works!</p>

<div>

    <h2>One way data binding -> Use Event Binding</h2>

    <input type="button" value="Click Here" (click)="fun()">

    <br>

    {{msg}}

</div>

[event-binding] -> [event-binding.component.ts]

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

@Component({

  selector: 'app-event-binding',

  templateUrl: './event-binding.component.html',

  styleUrls: ['./event-binding.component.css']

})

export class EventBindingComponent implements OnInit {

  msg:String = "Hi";

  constructor() { }

  ngOnInit(): void {

  }

  fun(){

    //alert("Click On Event")

    this.msg = "Hello";

  }

}

Two-way Binding. ?

Code :- [two-way-binding] -> [two-way-binding.component.html] :-

<p>two-way-binding works!</p>

<div>

    <h2>Two way Data Binding using ngModel</h2>

    Age : <input type="number" [(ngModel)]="age">

    <br>

    {{age}}

</div>

[two-way-binding] -> [two-way-binding.component.ts] :-

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

@Component({

  selector: 'app-two-way-binding',

  templateUrl: './two-way-binding.component.html',

  styleUrls: ['./two-way-binding.component.css']

})

export class TwoWayBindingComponent implements OnInit {

  age:number = 21;

  constructor() { }

  ngOnInit(): void {

  }

}

Form Validations ?

Code :- [mdf-login-page] -> [mdf-login-page.coponent.ts] :-

|  |
| --- |
| import { Component, OnInit } from '@angular/core'; |
|  |

|  |
| --- |
| import { FormControl, FormGroup, Validators } from '@angular/forms'; |
| import { CustomService } from '../custom.service'; |
|  |

|  |
| --- |
|  |

|  |
| --- |
| import { LoginService } from '../login.service'; |
|  |

|  |
| --- |
| @Component({ |
|  |

|  |
| --- |
| selector: 'app-mdf-login-page', |
|  |

|  |
| --- |
| templateUrl: './mdf-login-page.component.html', |
|  |

|  |
| --- |
| styleUrls: ['./mdf-login-page.component.css'] |
|  |

|  |
| --- |
| }) |
|  |

|  |
| --- |
| export class MdfLoginPageComponent implements OnInit { |
|  |

|  |
| --- |
| loginRef = new FormGroup({ |
|  |

|  |
| --- |
| email:new FormControl("",[Validators.required]), |
|  |

|  |
| --- |
| pass:new FormControl("",[Validators.required]) |
|  |

|  |
| --- |
| }); |
|  |

|  |
| --- |
| msg:string=""; |
|  |

|  |
| --- |
| constructor(public ls:LoginService) { } // DI for LoginService class |
|  |

|  |
| --- |
| ngOnInit(): void { |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| checkUser(){ |
|  |

|  |
| --- |
| let login = this.loginRef.value; |
|  |

|  |
| --- |
| //console.log(login); |
|  |

|  |
| --- |
| if(login.email=="raj@gmail.com" && login.pass=="123"){ |
|  |

|  |
| --- |
| this.msg = "Successfully login" |
|  |

|  |
| --- |
| }else { |
|  |

|  |
| --- |
| this.msg = "Failure try once again" |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| let cs = new CustomService(); |
|  |

|  |
| --- |
| this.msg = cs.checkUserDetails(login); |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |

|  |
| --- |
| this.loginRef.reset(); |
|  |

|  |
| --- |
| } |
|  |

}

[mdf-login-page] -> [mdf-login-page.coponent.html] :-

|  |
| --- |
| <div> |
|  |

|  |
| --- |
| <h2>Login Page using Model Driven Form</h2> |
|  |

|  |
| --- |
| <form [formGroup]="loginRef" (ngSubmit)="checkUser()"> |
|  |

|  |
| --- |
| <label>EmailId</label> |
|  |

|  |
| --- |
| <input type="email" formControlName="email"/> |
|  |

|  |
| --- |
| <div \*ngIf="loginRef.controls['email']?.invalid && loginRef.controls['email']?.touched"> |
|  |

|  |
| --- |
| <span style="color:red">UserName is requried</span> |
|  |

|  |
| --- |
| </div> |
|  |

|  |
| --- |
| <br/> |
|  |

|  |
| --- |
| <label>Password</label> |
|  |

|  |
| --- |
| <input type="password" formControlName="pass"/> |
|  |

|  |
| --- |
| <div \*ngIf="loginRef.controls['pass']?.invalid && loginRef.controls['pass']?.touched"> |
|  |

|  |
| --- |
| <span style="color:red">Password is required</span> |
|  |

|  |
| --- |
| </div> |
|  |

|  |
| --- |
| <br/> |
|  |

|  |
| --- |
| <input type="submit" value="submit" [disabled]="loginRef.invalid"/> |
|  |

|  |
| --- |
| <input type="reset" value="reset"/> |
|  |

|  |
| --- |
| </form> |
|  |

|  |
| --- |
| <br/> |
|  |

|  |
| --- |
| <span style="color:red">{{msg}}</span> |
|  |

</div>

Directives. ?

Code :- [attribute-directive]->[ attribute-directive.component.html]

<p>attribute-directive works!</p>

<div>

    <h2>Attribute directive is ngStyle and ngClass</h2>

    <p style="color:red">Apply CSS Effects</p>

    <p [ngStyle]="{'color':'red'}">Apply CSS Effects</p>

    <p [ngStyle]="styleVariable">Apply CSS Effects</p>

</div>

[attribute-directive]->[ attribute-directive.component.ts] :-

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

@Component({

  selector: 'app-attribute-directive',

  templateUrl: './attribute-directive.component.html',

  styleUrls: ['./attribute-directive.component.css']

})

export class AttributeDirectiveComponent implements OnInit {

  styleVariable = {'color':'red'}

  constructor() { }

  ngOnInit(): void {

  }

}

Pipes. ?

Code :- [pipe] -> [pipe.component.ts] :-

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

@Component({

  selector: 'app-pipe',

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

  styleUrls: ['./pipe.component.css']

})

export class PipeComponent implements OnInit {

  birthday = new Date(1988, 3, 15);

  constructor() { }

  ngOnInit(): void {

  }

}

[pipe] -> [pipe.component.html] :-

<p>pipe works!</p>

<p>My birthday is {{ birthday | date }}</p>

<br>

My friend birthday is

{{ birthday | date | uppercase}}

Routing Mechanisms ?

Code :- [login] -> [login.component.ts] :-

|  |
| --- |
| import { Component, OnInit } from '@angular/core'; |
|  |

|  |
| --- |
| import { FormControl, FormGroup } from '@angular/forms'; |
|  |

|  |
| --- |
| import { Router } from '@angular/router'; |
|  |

|  |
| --- |
| import { LoginService } from '../login.service'; |
|  |

|  |
| --- |
| @Component({ |
|  |

|  |
| --- |
| selector: 'app-login', |
|  |

|  |
| --- |
| templateUrl: './login.component.html', |
|  |

|  |
| --- |
| styleUrls: ['./login.component.css'] |
|  |

|  |
| --- |
| }) |
|  |

|  |
| --- |
| export class LoginComponent implements OnInit { |
|  |

|  |
| --- |
| loginRef = new FormGroup({ |
|  |

|  |
| --- |
| user:new FormControl(), |
|  |

|  |
| --- |
| pass:new FormControl() |
|  |

|  |
| --- |
| }) |
|  |

|  |
| --- |
| constructor(public loginSer:LoginService, |
|  |

|  |
| --- |
| public router:Router) { } //DI for Service class. |
|  |

|  |
| --- |
| msg:string=""; |
|  |

|  |
| --- |
| ngOnInit(): void { |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| checkUser() { |
|  |

|  |
| --- |
| let login = this.loginRef.value; |
|  |

|  |
| --- |
| this.loginSer.checkUser().subscribe(result=> { |
|  |

|  |
| --- |
| let message = result.find(l=>l.user==login.user && l.pass==login.pass); |
|  |

|  |
| --- |
| if(message==undefined){ |
|  |

|  |
| --- |
| this.msg = "Invalid username or password" |
|  |

|  |
| --- |
| }else { |
|  |

|  |
| --- |
| // in reponse after success message from server. |
|  |

|  |
| --- |
| //Server side technologies |
|  |

|  |
| --- |
| //provide you unique id ie session id. |
|  |

|  |
| --- |
| // JWT (json web token) |
|  |

|  |
| --- |
| sessionStorage.setItem("token",login.user); |
|  |

|  |
| --- |
| this.router.navigate(["home",login.user]); // append data through url |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| },error=>console.log(error)); |
|  |

|  |
| --- |
| this.loginRef.reset(); |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

}

[login] -> [login.component.html] :-

|  |
| --- |
| <div> |
|  |

|  |
| --- |
| <H2>Login</H2> |
|  |

|  |
| --- |
| <form [formGroup]="loginRef" (ngSubmit)="checkUser()"> |
|  |

|  |
| --- |
| <label>UserName</label> |
|  |

|  |
| --- |
| <input type="text" formControlName="user"/><br/> |
|  |

|  |
| --- |
| <label>Password</label> |
|  |

|  |
| --- |
| <input type="password" formControlName="pass"/><br/> |
|  |

|  |
| --- |
| <input type="submit" value="submit"/> |
|  |

|  |
| --- |
| <input type="reset" value="reset"/> |
|  |

|  |
| --- |
| </form> |
|  |

|  |
| --- |
| <br/> |
|  |

|  |
| --- |
| <span style="color:red">{{msg}}</span> |
|  |

</div>