1. **REACTIVE FORMS:**

**Imports:**

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

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

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

**app.component.ts:**

export class AppComponent

{

loginform = new FormGroup({

email : new FormControl(),

pwd: new FormControl()

});

onsubmit()

{

console.log(this.loginform.value);

console.log("Email:"+this.loginform.controls['email'].value);

}

}

**app.component.html:**

<form [formGroup]="loginform" (submit) = "onsubmit()">

Email:<input type="text" formControlName="email">

<br>

Password:<input type="password" formControlName="pwd">

<br>

<input type="submit" value="submit">

</form>

**Output:**

{email: 'cvr@gmail.com', pwd: '123456'}

Email:cvr@gmail.com

1. **TEMPLATE-DRIVEN FORMS:**

**Imports:**

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

**app.component.ts:**

export class AppComponent

{

userdata = {

email:"",

pwd:""

}

onsubmit(loginform:any)

{

console.log(loginform.value);

console.log("Email:"+this.userdata.email);

}

}

**app.component.html:**

<form #loginform="ngForm" (ngSubmit) = "onsubmit(loginform)">

Email:<input type="text" name="t1" [(ngModel)]="userdata.email ">

<br>

Password:<input type="password" name="t2" [(ngModel)]="userdata.pwd">

<br>

<input type="submit" value="submit">

</form>

**Output:**

{t1: 'csit@gmail.com', t2: '123456'}

Email:csit@gmail.com

1. **PIPES:**

* **BUILT-IN PIPES:**

**app.component.ts:**

export class AppComponent

{

name:string = "Angular";

amount:number = 500;

d = new Date();

per:number = 0.256;

}

**app.component.html:**

<h3>Built-in Pipes</h3>

Uppercase Pipe: {{name | uppercase}}

<br>Percent Pipe: {{per | percent}}

<br>Date Pipe: {{d | date}}

<br>Currency Pipe with parameters: {{amount | currency: 'USD'}}

<br>Date Pipe with parameters: {{d | date : 'short'}}

<br>Date Pipe with parameters: {{d | date : 'full'}}

<br>Chained Pipes: {{name | uppercase | slice : 2}}

**Output:**

**Built-in Pipes**

Uppercase Pipe: ANGULAR  
Percent Pipe: 26%  
Date Pipe: Dec 28, 2021  
Currency Pipe with parameters: $500.00  
Date Pipe with parameters: 12/28/21, 7:09 PM  
Date Pipe with parameters: Tuesday, December 28, 2021 at 7:09:27 PM GMT+05:30  
Chained Pipes: GULAR

* **CUSTOM PIPES:**

**Creating a new pipe:**

ng g pipe exponent

**exponent.pipe.ts:**

import { Pipe, PipeTransform } from '@angular/core';

@Pipe({

**name: 'exponent'**

})

export class ExponentPipe implements PipeTransform {

transform(value:number, exp:number): number{

return Math.pow(value,exp)

}

}

**Import custom pipe in root module/app.module.ts:**

import { ExponentPipe } from './exponent.pipe';

**app.component.html:**

Custom Pipe Demo: {{4 | exponent: 5}}

**Output:**

Custom Pipe Demo: 1024 // i.e 4^5

1. **DEPENDENCY INJECTION:**

* **CREATING A NEW INJECTABLE SERVICE:**

ng g service courses

**courses.service.ts:**

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

@Injectable({

providedIn: 'root'

})

export class CoursesService {

courses = ['angular','node','react','express'];

getcourses()

{

return this.courses;

}

}

**Add service to providers array in root module/app.module.ts:**

@NgModule({

providers: [CoursesService],

})

* **USING SERVICE IN A COMPONENT:**

**Import service in the component:**

import { CoursesService } from './courses.service';

**app.component.ts:**

export class AppComponent

{

cou:any;

constructor(c:CoursesService)

{

this.cou = c;

}

}

**app.component.html:**

<h1>List of Courses from Courses service:</h1>

{{cou.getcourses()}}

**Output:**

**List of Courses from Courses service:**

angular,node,react,express

1. **CREATING OBSERVABLES:**

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

import { Observable } from 'rxjs';

@Component({

  selector: 'app-root',

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

  styleUrls: ['./app.component.css']

})

export class AppComponent implements OnInit {

  obs = new Observable((observer) => {

    console.log("Observable starts")

setTimeout(() => { observer.next("1") }, 1000);

setTimeout(() => { observer.next("2") }, 2000);

setTimeout(() => { observer.next("3") }, 3000);

setTimeout(() => { observer.next("4") }, 4000);

setTimeout(() => { observer.next("5") }, 5000);  })

  ngOnInit() {

     this.obs.subscribe( //Subscribing an Observable

      val=> { console.log(val) },

      error => { console.log("error")},

      () => {console.log("Completed")}

    )

  }

}

**OUTPUT:**

ngOnInit

Observable starts

1

2

3

4

5