Angular

Front end framework

Library vs Framework:

Structure: framework is structured. It follows the project structure, naming conventions,

No additional installations

SPA : single page application-

Performance

TypeScript:

1. Why TS?

Angular 1 : AngularJS – JS

JS -> Type Safety ,

var num = ”hello”;

num = 67.45;

dynamically typed language.

Var s = {};

Runtime challenges due to the dynamic typing. TS – TypeScript

Wrapper around the JS with type safety. It has its own compiler – transpiler to check the type at every stage.

Transpiler : every browser understands JS. After compilation of TS, it is translated to pure JS.

Var num=78.43;

1. What is TS?
2. TS coding
3. Variables and data types
4. Conditional stmt
5. Looping stmt
6. Functions
7. Classes

Variables and data types:

// number (int and float), string, boolean, any

// custom types: (class) Student, Product, Employee

// const - let - var

let num : number = 23.78;

let username = "user123";

let stu\_status : boolean;

let x : any ;

let y ;

x=56.89;

x='jhjhh';

x=false;

x={id:19, name:"Amit"};

let data : (string | number) = "hi";

data = 777;

// arrays

let values :  number[] = [11,13,67,98.67];

b. conditional statements

const num1 = 100;

const num2 = 45;

let largest:number;

// if (num1 > num2){

//     largest = num1;

// }

// else {

//     largest = num2;

// }

largest = (num1 > num2) ? num1 : num2;

console.log(`${largest} is the largest number`);

c. loops:

// simple for

for (let i =1; i<=5;i++){

    console.log(i);

}

// for loop is applied on data received from server - arrays

const data : number[]= [11,12,13,14,15];

for (let i=0; i< data.length; i++){

    console.log(data[i]);

}

// for-of - used for accessing the elements directly without the index

for(let num of data){

    console.log(num);

}

d. functions

/\* JS syntax:

function addition(n1,n2){

    return n1+n2;

}

addition();

addition(10,"hi");

addition(100,"hi",77);

\*/

let val : number ;

function addition(n1:number,n2:number): number {

    return n1+n2;

}

const answer: number = addition(30,40);

// arrow functions

function square(num:number): number{

    return num\*\*2;

}

// equivalent arrow function, stored in var

const sq = (num:number):number => num\*\*2 ;

console.log(sq(10));

==== OOP ===========

export class Student {

    // instance var

    id : number;

    name : string;

    // constructor

    constructor(){

        console.log("student object is created");

    }

    // instance methods

    setDetails(i:number, n:string):void{

        this.id = i;

        this.name = n;

        this.showDetails();

    }

    showDetails():void{

        console.log(`Student id:${this.id}, Name:${this.name}`);

    }

}

const s1 : Student = new Student();

s1.setDetails(101,"Amit")

Angular 16

Installations:

Node 18

( node --version , npm --version )

npm install -g @angular/cli@16.1.0

(ng version)