Angular js

Response

Request: via http or https

Browser

HTML , CSS, JS,

Angular divides the our html pages in different components.

Typescript :- is required in angular.

Typescript :- is a superset of javascript

File.js

Js file will run on browser

TSC

File.ts - > file.js

File.ts

node -v to check the node version

npm -v is the command to check npm version on any os

npm install typescript -g // it will install typescript global module

tsc -v

creating a typescript: name.ts

var :- it will declare a variable

let keyword to declare the variable

console.log which is use to print the msg on console

var msg =”hello first prg of typescript”;

console.log(msg);

Typescript Datatypes:

Static:- built in and user-defined

Generics

Decorators

Built-in datatype :- number, void, String, Null, Boolean

Var marks:number =66;

Var name:String =”sunil sharma”;

Var status:Boolean=false;

Void means which is undefined.

Var test:void=undefined;

Test=null;

Test=12332;

Any type:- it is use to declare a variable with multiple values;

Var data:any=”sunil”;

Data=234;

Data=true;

User defined data types

Array, class, touple, enum, interface, functions

Var list:string[] = [“sunil”,”anil”,”ajay”];

Var list:Array<string>= [“hi”,”hello”,”123”];

Touple:- var tv:[string, number];

Tv =[“sunil”,123,”anil”,343];

Interface:-

Interface demo

{

Sum(no:number,no2:number): void;

}

Let mathssol:demo={

Sum(no:number,no2:number)

{

Console.log(no+no2);

}

}

Class classname

{

Variables:dt;

Variable2:datatype;

}

enum enumname{val1,val2,val3,};

let envar:enumname;

envar = enumname.val1;

enum City{indore, pune, Mumbai};

var c:City;

c= City.indore;

functions

c/c++/java

returnType functionName(datatype variable, dt v, dt v2, dt v3){

}

Typescript

function functionname(v1:type,v2:type,v3:type):returntype

{

Code to be executed;

[return stmt];

}

Anonymous function

Var sum=function(v1:type,v2:type):number

{ code to be executed;

[Return stmt if any ]

}

\*Arrow function :-

Generic :-

Java :- ArrayList<Student> objal = new ArrayList<Student>();

Function max<T>(v1:T,v2:T):T

{

T c;

If (v1>v2)

c=v1;

Else

c=v2;

return c;

}

Console.log(max<number>(10,23));

3) Decorators

We need to enable it in our tsconfig.json file

Decorators are experimental

@ we apply our decorators

Var a:Null;

a=undefined;

Null represents a value,

Undefined means no value is assigned.

Scoping Rule:

Var : it is not providing.

Let : typesafety of variable

When we declare a variable using var keyword it is having function scope.

Let keyword limit the variable to upcoming braces

Decision Making

If(condition)

Stmt;

else

stmt;

Loop

While()

{

}

Do

{

}

while();

for(init;condition;increment/decrement)

{

Stmt(s);

}

For of loop

Syntax:-

for(var of array/tuple)

{

Var :- will print value one after another.

}

For in loop

For (var i in list)

{

I is hold the index and we can print values using index.

}

Let will be used when we only want a variable to be accessed in current block . and if we want to use the variable across all of the block we are going to use it with var keyword.

Functions

:- anonymous function , arrow function

Function functionname([parameterlist]:returntype

{

Code inside the function

}

Var anil=function([parameter]):returntype {

Stmt(s);

}

Anil();

We can declare optional parameter and default parameters to set values of parameter for a function

Syntax:

Function funName(v1:type,v2:type,v3?:type)

{

}

? question mark is declare this parameter is optional

Default parameter :- in function to set the default value for any parameter. This can be set from right hand side

Function funName(v1:type,v2:type,v3:type=default value):return type

{

Function body

}

Rest Parameter

To declare a rest parameter we use … 3 dots. Before the parameter and this will be last parameter of the function

Only one rest parameter is allowed.

It can only be of type array

Syntax:- function funName(v1:type,**…v2:type[]) : void{**

**Function body**

**}**

Arrow function:-

Syntax :- (para1,para2, para3….)=> expression;

Var myf1 = function(v1,v2,…):return type

{ function body

}

// now with arrow function

Var myf2=(v1,v2,v3….):return type

{

Function body

}

Classes in typescript

Syntax for class dec:-

Class <classname>

{

Variables;

Methods

Constructor :- constructor is declare with the constructor keyword sadf

}

Class item

{

Itemid:number;

Itemname:string;

Constructor(id:number,name:string)

{

this.Itemid=id;

this.Itemname=name;

}

printValue():void{

console.log(this.itemid+this.itemname);

}

}

Var obj2= new item(123,”shirt”);

Public, private, and protected mode to variables and function of classes.

Arrow function inside class

Class A

{

Show=()=>{

Body for arrow function

}

}

Typescript module

: two types of module is present in typescript

1. Internal module :- internal module were present in older version of typescript which is used to group together the classes, function and interfaces. In new version of typescript modules are called namespace.
2. External Module :- are namespace.

To internal module :

Module moduleName{

Module body

}

For external module/Namespace:

Export class/interface/function name\_of\_class/interface/function

{

Body or code declaration will be here

}

Import {classname} from ‘./path to the ts file / name of the file’

If the our module is inside some folder like

Moduledir/mymodule.ts

Import {demo} from ‘./moduledir/mymodule’;

Your file is in project dir and module is in moduledir

Import {demo} from’../moduledir/mymodule’;