Typescript

It is super set of javascript

We can not run typescript code directly to the browser, first we compile it and after compilation we get js code file

Node

Npm command which is node package manager

Npm install typescript@version -g { -g means global }

You will install typescript 4.2

npm install [typescript@4.2](mailto:typescript@4.2) -g

tsc –v

You have to compile the ts file by using

tsc filename.ts

After compilation

node filename.js

var and let keyword is used to declare the variables: { till we did not study the datatypes in typescript}

var variablename; or let variablename;

var object = new classname();

let object= new classname();

var x=”hello world”;

console.log(x);

console.log(“this is typescript class”);

alert in javascript

to declare function

function functionname([parameter list])

{

Code

Return value;

}

Typescript datatypes

Static

Built-in (primitive)

User defined (derived)

Generic

Decorators

Built in primitive:-

Number

String

Void

Null

Boolean

Any { means it can store any value}

User defined

Array

Class

Touple

Enum

Interface

Functions

No:number;

Name:string;

Array

Var list:number[]= [12,23,33];

Var list:Array<number> = [1,5,4];

Array.length

To declare the array without initialzing

Var arrayname:datatype[];

Touple

Var toutplename:[datatype1, datatype2, datatype3];

Var and let keyword

Function functionname([parameter\_list ] ) :return type

{

Code to be executed

// return value;

}

Anonymous function

Var variablename = function(parameterlist) : return type {

Code that need to be executed

}

Optional parameter

Default parameter

Rest parameter

Optional Parameter

Function function\_name(para1:dt1, para2:dt2, para3:?dt3):return type{

}

Default parameter

Function function\_name(para1:dt1, para2:dt2, para3=value):return type{

}

Function functionname(para1:dt1,para2:dt2,…para3:datatype[])): return type

{

Code

}

Enum, class, interface

Function

Arrow Function

Parameter

Arrow notation =>

Function body

Var varname = (para1:dt1,para2:dt2,para3:datatype3): return type{

Arrow function body;

}

Classes in Typescript

class classname

{

variable:datatype;

constructor()

{

}

[Function] functionname([ parameter ]) : return type

{

To access class member inside class function we need to use this keyword

}

}

Public

Private

Protected access modifier

Inheritance

Single

Multilevel

Class A

Class B

Class C

Extends keyword is use to inherit the class

To declare a module in typescript

We just need to write class or method like following

export Classs classname{

}

Interface

Interface myInterface

{

Variable:datatype;

Function functionname();

}

An interface can extend another interface.

Class classname implements interface

{

Variable and method of interface redeclare again here.

}

Arrow function in class

Nameoffunction=()=>{

Code need to be executed.

}