**Day 1**

TypeScript is a type of scripting language provided by micro soft. Typescript known as super set of JavaScript which support all features of ES6.

Java Script doesn’t support static data types concept.

ES5

var n; undefined

var a=10; number type consider

a=”Ravi”; string type consider

a=true;

a=new Date();

ES6

function add(a,b) {

console.log(a+” ”+b);

}

add(10,20);

add(“A”,”B”);

add(100);

add();

function sayHello() {

this can return any type of value as well as no return type.

}

TypeScript support data type at compile time or static data type declaration.

Still all browser doesn’t support typescript. We need to convert TS to JS ie Transpiler (compiler).

TS file we use in developer mode.

Using npm we need to download Typescript compiler or transpiler

npm install -g typescript

tsc welcome.ts this command convert ts file to js

node welcome.js using this command run the program

Typescript support all type of data types.

Like

number,

string

boolean

object

any

array : In JS array is use to store any type of values.

var arrayName:datatype[];

var, let and const

var is use to declare global scope.

Let is use to declare block scope.

Using var keyword we can re-declare same variable once again with same or different value.

But using let we can’t re-declare same variable once again.

const is use to declare constant value we can’t change that value.

Command to convert TS to JS version ES6

tsc --target es6 varletconst.ts

tsconfig.json

TypeScript functions

Function with default value (set default value), optional parameter( use ?) , rest operator or parameter and spread parameter.

Rest parameter syntax

…varaibleName:Array<DataType> it can receive 0 or 1 or many. In one function it must last parameter and only one rest parameter we can use inside one function.

Normal function : this function we can call before function declaration as well as after function declaration. It support js hosting features.

Expression style function : this type of function must be invoke after declaration. It doesn’t support js hosting features.

Arrow style function

Callback function : passing the function as parameter or function itself to another function as parameter is known as callback function.

IIFE function (Immediate Invoke function execution/expression)

(functionBody)(functonCall)

Object : any real world entity

Property or variables 🡪 have

Person

Behaviour or function / method -🡪 do/does

Bank

Animal

Car

Employee

Till ES5 JavaScript we if need to describe the object

We are/were using literal style or function style.

After ES6 JS or TS we can use

Literal style or function style or class style

Type script support interface.

In TypeScript we use interface to create type of object using literal style.

Constructor is a type of special function which help to create the memory.

1. To create the constructor we need to write the function with name as constructor.
2. It will call automatically whenever we create the object.
3. Constructor no return type not even void also.
4. We can write explicitly only one constructor we can’t write more than one constructor in JS(ES6) as well as typescript.

TypeScript support Access specifiers

Like

public

private

protected

Encapsulation : binding or wrapping data (variables) and code (function or methods) in a single unit is known as encapsulation.

Example : class

Access Specifiers with constructor parameter variables.

Inheritance

From ES6 onward to achieve inheritance in Ts we can use extends keyword.

OOPs

Manager/Developer Is a Employee

Employee Has a Address

Class A {

let obj = new B();

}

Class B {

Let obj = new A();

}

Class Employee {

Let ladd = new Address();

Let padd = new Adress();

}

Class Address {

}

Class Student {

let sh=new StudentHistory();

}

Class StudentHistory() {

}

Association

Aggregation : this association is weak association.

Composition : this association is strong association.

We need to break out code in more than one file using modules concept. And those file using import as well as export.

Using modules we can create more than one function, variable class or interface which have same name but different purpose.

Create separate folder

Typescript\_modules

tsc --init this command help use to create tsconfig file

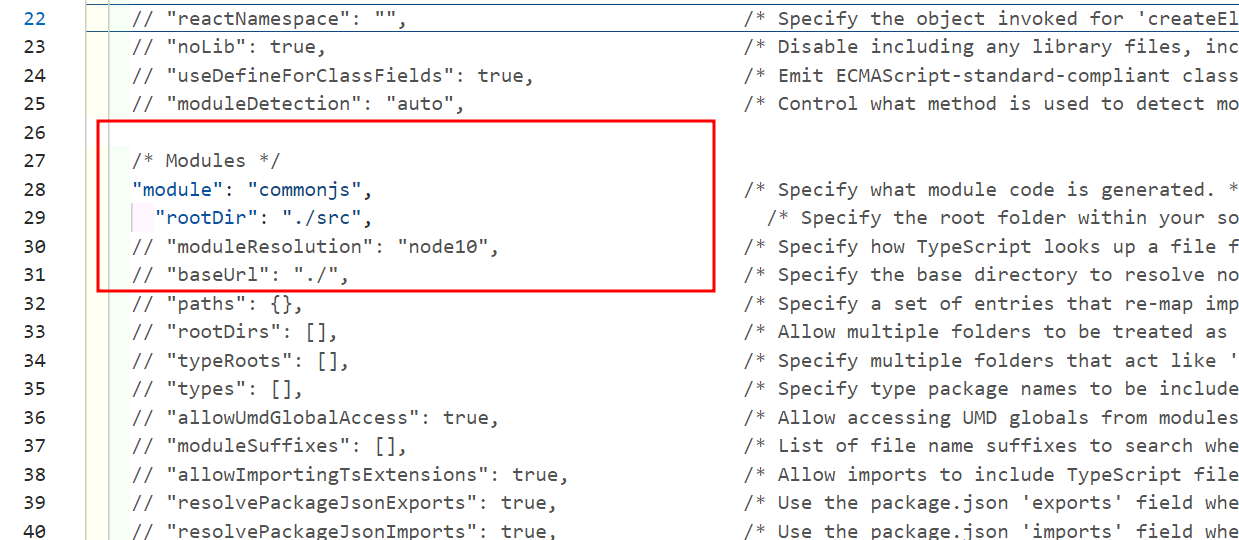
create src folder then create more than one ts file.

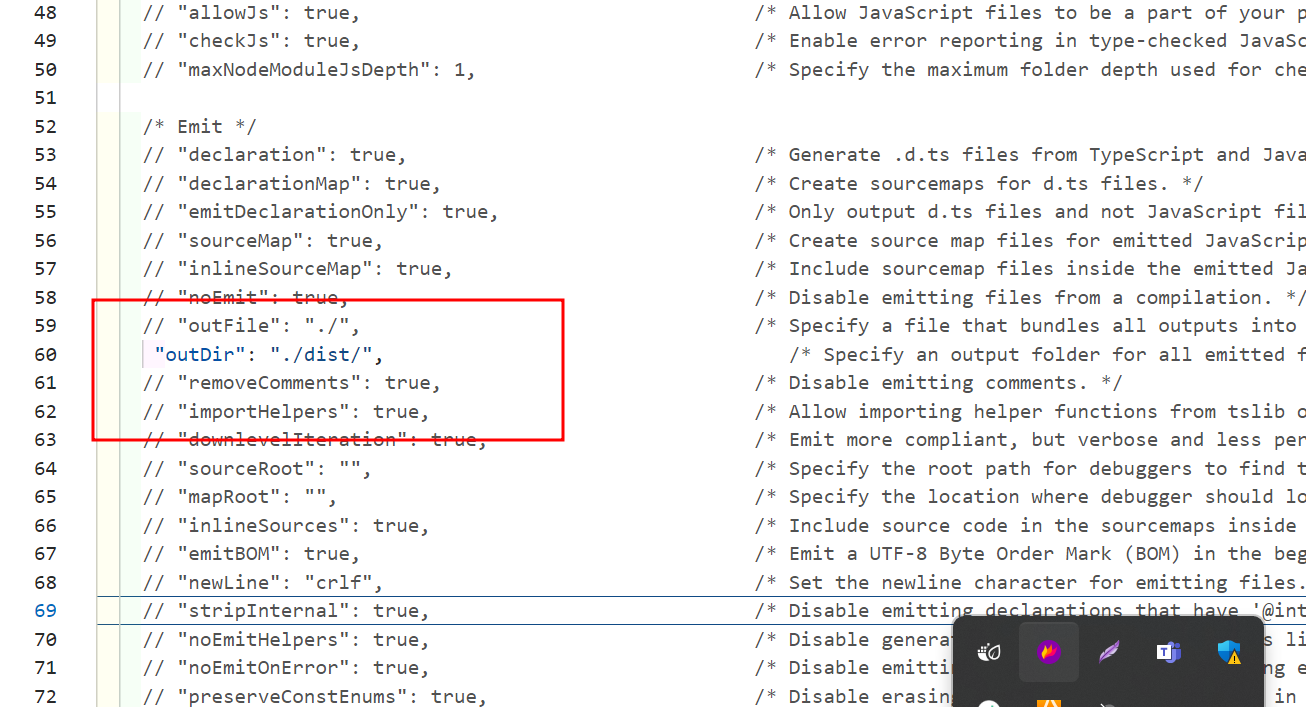
file1.ts

file2.ts

main.ts

enable few options in tsconig.json file





Then to convert ts to js run the command as

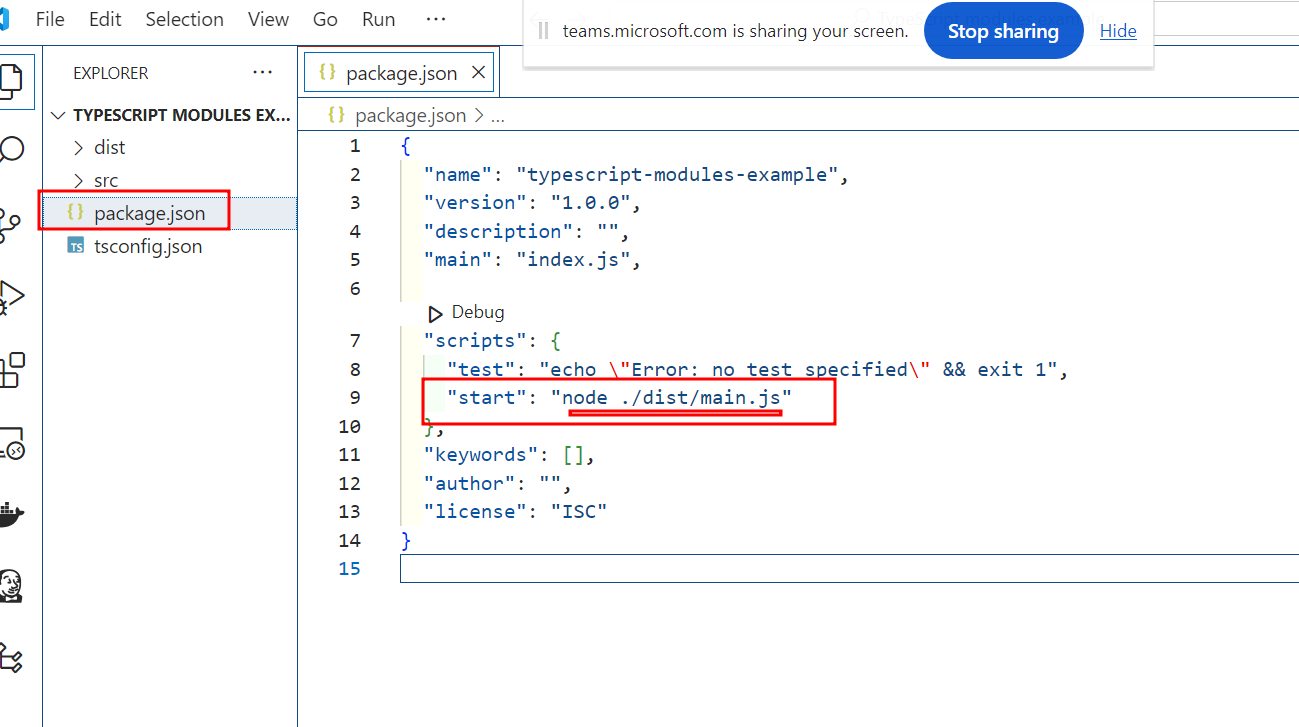
tsc

then it will convert all ts to js in dist folder

then create package.json file

npm init -y

provide main js file details in package.json file



To run the application

npm start

