**TypeScript**

## **Installing TypeScript**

npm install -g typescript

## **Check Installation**

Try opening a terminal anywhere and running tsc -v to see if it has been properly installed.

tsc -v

Version 1.8.10

## **Compiling to JavaScript**

**The following command takes a TypeScript file named main.ts and translates it into its JavaScript version main.js. If main.js already exists it will be overwritten.**

tsc main.ts

**We can also compile multiple files at once by listing all of them or by applying wildcards:**

**# Will result in separate .js files: main.js worker.js.**

tsc main.ts worker.ts

**# Compiles all .ts files in the current folder. Does NOT work recursively.**

tsc \*.ts

**We can also use the --watch option to automatically compile a TypeScript file when changes are made:**

**# Initializes a watcher process that will keep main.js up to date.**

tsc main.ts –watch

## **Static Typing**

## **Variable in TypeScript**

A variable in typescript can be boolean, string, any or number[] and any variable in typescript can be declared using keyword - var and let. But it is always recommended to use let keyword to define a variable in typescript as it provides type safety.Following are the different ways to define a variable in typescript.

let num : number; //Defines a number variable

let str : String = 'Devglan'; //Defines a string variable and initialises to Devglan

let array : String[] = ['a', 'b', 'c']; //Defines a string array variable

let random : any[] = ['a', 1, true];

Typescript also provides support for enum similar to object oriented language to declare constant.Following is an exampple to declare enums in typescript.

enum Role {Admin, User,SuperAdmin};

let role = Role.Admin;

## **Arrow Functions**

Arrow functions are similar to lambda expression in java. We can execute inline functions using it. Suppose we have a code of block as follow:

var greet = function greeter(person) {

console.log( "Hello, " + person);

}

This can be replaced using arrow function as var greet = (person) => console.log(person);

## **Interface in TypeScript**

Interface is used to define reusable custom data types. For example, if you want to create a reusable User object then you can have name, age, gender as its types.

interface User {

name : String

age : number

gender : String

}

## **TypeScript Class**

Class is a logical entity that has variables and functions that are highly related to perform single operation. The difference between a class and an interface is that class can have implementation inside it whereas an interface does not have any implementation inside it. Following is a simple User class that has fields and methods:

class User {

name : string

age : number

gender : string

createUser(userr : User){

//create user

}

}

Now we can instantiate this class and use it's method in following ways

let user = new User();

user.name = 'Dhiraj';

user.gender = 'Male';

user.age = 23;

user.createUser(user);

## **TypeScript Constructor**

Constructor is used to instantiate a class.The declaration of a constructor is similar to a method without any return type.Following is the declaration of constructor for class defined above.

constructor(name : string, age : number, gender : string) {

this.name = name;

this.age = age;

this.gender = gender;

}

Now doing so the line let user = new User(); will show compiler error as there is no matching constructor in the class definition and typescript does not support constructor overloading.To remove this compile error, you can make the constructor parameters optional as follow:

constructor(name? : string, age? : number, gender? : string) {

this.name = name;

this.age = age;

this.gender = gender;

}

## **Access Modifier in Typescript**

Access modifier is used to modify the access level of the variables and methods used inside any class.We have 3 different access modifiers in typescript - public, private and protected. By default all the members are public and tht's the reason we were able to initialise class variable from outside the class.Following is the syntax to use access modifiers.

private name : string

private age : number

private gender : string