Agenda:

- Introduction
- Limitation of JavaScript
- What is TypeScript.
- Benefits of TypeScript
- Setup the Environment
- First TypeScript code

Introduction

Limitations of JavaScript / ECMAScript 5:

1. It's not type safe.
2. It becomes cumbersome to manage as it becomes larger.
3. Its interpreted language and not compiled, hence errors can be identified only when we execute the script.

What is TypeScript?

1. It's not replacement of JavaScript nor it adds any new feature of JavaScript.
2. TypeScript = **JavaScript + Types** = Typed superset of JavaScript
3. It’s **not mandatory** to strongly type everything when we are type scripting.
4. It's **compiled** to generate JavaScript.
5. Also, any valid **.js** file can be renamed to **.ts** and compiled with other TypeScript files.
6. TypeScript generated JavaScript can reuse all of the existing JavaScript frameworks, tools, and libraries.
7. Its object oriented and supports core features like interfaces and classes. As a prerequisite, you are supposed to have good knowledge on **Object Oriented Programming** and basic knowledge on **JavaScript**.
8. It was designed by **Anders Hejlsberg** (founder of C#) at Microsoft. Its **open source** and can be used in any place where we would need JavaScript.
9. There are almost 40 languages which are superset of JavaScript. On which they generate **.js** on compiling, TypeScript is just one and most popular of these languages. Few of the other languages are: **PureScript, CoffeeScript, LiveScript…**

Some TypeScript language features which are based on **ECMAScript 5** (JavaScript) and **ECMAScript 2015.**

- Type annotations and compile-time type checking
- Type inference
- Type erasure
- Classes
- Modules
- Interfaces

DECCANSOFT

- Enumerated type
- Optional and default Parameters
- Generic
- Namespaces
- Await

Versions:

1. TypeScript 0.8 was released on Oct 2012
2. TypeScript 0.9 was released in 2013
3. TypeScript 1.0 was released in 2014
4. TypeScript 2.0 was released on Sep 2016

Benefits of TypeScript:

1. It compiles the code and **generate syntax errors** if any. This helps to highlight errors before the script is executed.
2. Because of Object Oriented features, **it's reusable and easier** to manage in large and complicated projects.
3. **Angular framework** is written in TypeScript and it's recommended that developers use this language in their projects as well.
4. Due to the static typing, code written in TypeScript is **more predictable**, and is generally **easier to debug**.

Setup the Environment

Visual Studio 2015 and Visual Studio 2017 include TypeScript by default.

To develop Angular applications, we need **Typescript 2.2.0 or its higher versions.**

For TypeScript 2.X Support

1. Upgrade Visual Studio 2015 with Update 3
2. Install TypeScript 2.0 for Visual Studio 2015

Before installation we can check our Typescript version in the Visual Studio

Open Visual Studio → go to **Help** → click on **About Microsoft Visual Studio**

- To get the latest version of **Typescript for Visual Studio 2015**
  - https://www.microsoft.com/en-us/download/details.aspx?id=48593
- To get the latest version of **Typescript for Visual Studio 2017**
  - https://www.microsoft.com/en-us/download/details.aspx?id=55258

Install TypeScript using Node Package Manager (NPM)

DECCANSOFT

```
npm install -g TypeScript
```

Compile TypeScript File

```
tsc.exe helloworld.ts
```

- compiling one or more TypeScript files to one JavaScript file

```
tsc --outFile file.js file1.ts file2.ts file3.ts
```

- For auto compiling

```
tsc --watch file.ts
```

First TypeScript code

1. Create a new Project File → New Website → ASP.NET Empty Website
2. Template → TypeScript
3. To Scripts folder add SayHello.ts and edit the code as below

Ex: 1 - Hello world

```
function sample() {
   var message: string = "Hello world"
   alert(message);
}
sample();
```

Ex: 2 – Get full name

```
function SayHello(firstname: string, lastname: string): void {
   var msg: string = "Hello " + firstname + " " + lastname;
   console.log(msg)
   alert(msg)
}
SayHello("Sandeep", "Soni");
```

4. Build the project and it compiles all code along with .ts files to generate .js files.
5. To use – In HTML File include

```
   <script src="~/Scripts/sayhello.js"></script>
```

Note that we are referring to .js file and not .ts.

6. Solution Explorer → Right click on Project → Add → Add New Item → TypeScript JSon Configuration File → Add.

.js.map file: (Linker Address Map)

DECCANSOFT

- Visual Studio uses this map file to map the TypeScript code with the generated JavaScript code. Also, browser like Chrome use these map files to help us debug TypeScript files directly instead of debugging JavaScript file.
- For TypeScript debugging you need to instruct the compiler to produce a .map file. You can do that by placing a tsconfig.json file in the root of your project and specify the a few properties, like so.

```
{
  "compileOnSave": true,
  "compilerOptions": {
    "sourceMap": true
  }
}
```

Another Example:

```
class Person
{
    constructor(protected firstName: string, private lastName: string)
    { }
    SayHello = (): string =>
    {
        return this.firstName + " " + this.lastName;
    }
}

let person: Person = new Person("Sandeep1", "Soni1");
let msg = person.SayHello();
alert(msg)
```