TypeScript at a glance

[Ahmed shokry mohamed]
Why on the earth you have to use typescript ?

- You can use javascript directly, it's a dynamic types so you can develope faster and easier right ?
- But that will be good for small script which only you who works on it, but if you have a large project how to tell other developers what exactly that var have to be ? and how to prevent them or yourself from changing it be mistake ? so i thik in this case you need to use something restrict that var type right ? for that purpose thanks God we have Type script which is (optionally) static type
- Another point to talk about, all of use know that, js has it's own way to right oop, and it's somehow wired a little bit, so what if we need as we know in our way ? - fortunately type script solve that issue too

Conclusion :
If you gonna make large project or need to right oop in it's right way use tpyescript otherwise it's trully normal to use js.

[Doaa Ashraf Taha]

The purpose of using typeScript

- The goal of TypeScript is to help catch mistakes early through a type system and to make JavaScript development more efficient.
- It helps in reducing developer errors and makes debugging the code much easier.
- Types are a way to tell correct programs from incorrect before we run them by describing in our code how we plan to use our data.

[Menna Mohamed Omar]

What is Static Typing?

Statically typed languages like Java, Go, and C++ are able to catch type-related errors at compile time. However, in dynamically typed languages like Python, Ruby, and

JavaScript, such errors are not as easily discoverable because the types of variables are only known at runtime.

TypeScript Basics

As a superset of JavaScript, TypeScript is based on the same programming syntax as JavaScript, and valid JavaScript code is also valid TypeScript code. TypeScript can be configured to target different flavors of JavaScript while offering support for the latest JavaScript features, including those from ECMAScript 2015 (ES6) and beyond.

Why Typescript?

- The ability to compile down to a version of JavaScript that runs on all browsers
- Types make code management easier: Detecting errors on builds early as you can be warned about having type errors in your code.
- Increased team performance: The great thing about strictly typed programming languages is that they allow your developers to work on their task without relying on the other team members too much.
- TypeScript is popular and trusted by the biggest players in the industry

Types in Typescript

Primitive types

- number - represents any kind of numeric value - integer or float, hex, decimal, binary etc.
- string - represents any kind of string value;
- boolean - represents any boolean value, i.e. true or false;
- symbol - represents symbol values;
- null - represents null value only;
- undefined - represents undefined value only;
-

More types in typescript

- Any - indicates any possible value
- Void - This is commonly used with functions and tells the compiler that function doesn't return anything
- Never - represents the value that never occurs
- null - represents null value only;
- Unknown - It's meant to be a type-safe alternative to any type;

Composition types

- Unios | - allow you to specify variable's type that you can assign different types of values to.
- Literal - allow you to strictly define the possible value for the variable

Intersection types

- And & - intersection types function like logical and. Thus, you can create them using the and sign (&).

Tuples

- Structures closely related to arrays, so-called tuples can be utilized to specify a type of an array with a fixed number of elements

Enums

- They are used to simply provide more friendly names to numeric values.

Interface

- which allows the compiler to select the proper type for a particular variable without any special annotations.

[Hussin]
Please add your contribution here

Or you can work on a single portion together.

[Youssef]

Introduction

With the adoption and widespread usage of JavaScript as the standard programming language for web development, certain issues started to appear that caused lots of frustrations for Web Developers mainly due to JavaScript being a *weakly-typed* language. However, this did not stop developers who started looking for solutions to overcome JavaScript's weaknesses.

The Solution

Microsoft proposed a solution in 2012 to deal with these problems by publishing a superset open-source programming language called TypeScript. TypeScript introduced several benefits and solutions for the problems that were being faced by JavaScript developers at that time and it's currently being adopted as the go-to programming language in large-scale projects.

Benefits of Working with TypeScript

- **Statically Typed Language**

This essentially prevents variables declared by a certain type to be changed into another data type.

- **Predictability**
  With the assurance of everything staying the way it is as it was initially defined, this increases the predictability of the intended output.

- **Readability**
  Code is more readable when other collaborators can have a grasp of what a certain snippet is doing by knowing the type of data accepted and returned.

- **Improved Error Feedback**
  The compiler will actively alert the developer of errors made during the compile-time, instead of having to figure out what went wrong during the runtime, thus, decreasing testing time needed and saving more time.

- **Improved Code Completion**
  By knowing which type of data to expect while coding, IDEs start providing better code suggestions that are most fit.

References

[1] https://en.wikipedia.org/wiki/TypeScript
[2] https://www.stxnext.com/blog/typescript-pros-cons-javascript/
[3] https://www.altexsoft.com/blog/typescript-pros-and-cons/
[4] https://www.typescriptlang.org/docs/handbook