



TYPESCRIPT

19CSE100

PROGRAMMING LANGUAGE SURVEY

SREE SHARVESH S S
TIFAC-CORE in Cybersecurity
Amrita Vishwa Vidyapeetham

AGENDA

- I. Introduction & Overview
- II. Features of TypeScript
- III. Why do people hate JavaScript?
- IV. TypeScript Vs JavaScript
- V. Similar Alternatives
- VI. How to Install?
- VII. Some basic programs in TypeScript
- VIII. Disadvantages of TypeScript
- IX. Popularity of TypeScript
- X. Conclusion

Introduction

TYPESCRIPT – SOMETHING LIKE A “FLAVOUR” OR “VARIANT” OF JAVASCRIPT

- We frequently see the question “Should I learn JavaScript or TypeScript?”.
- The answer is that you can’t learn TypeScript without learning JavaScript! TypeScript shares syntax and runtime behavior with JavaScript, so anything you learn about JavaScript is helping you learn TypeScript at the same time.
- The relationship between TypeScript (TS) and JavaScript (JS) is rather unique among modern programming languages, so learning more about this relationship will help you understand how TypeScript adds to JavaScript.

Overview

TypeScript lets you write JavaScript the way you really want to.

It is created by the father of C# - Anders Hejlsberg

Any browser. Any host. Any OS. Opensource

Syntax based on
ECMAScript 4 &
ECMAScript 6
proposals

TS is a first and
foremost a
superset of
JavaScript

Any regular
JavaScript code is
a valid TypeScript
code.

FEATURES OF TYPESCRIPT

TypeScript provides developers with a powerful, efficient, and intuitive language for application development.

Some of its key features are,

- 1) **Strong typing**: TypeScript allows developers to specify the types of variables, function parameters, and return values, which can help prevent type errors and make code more self-documenting.
- 2) **Classes and interfaces**: TypeScript supports object-oriented programming features such as classes and interfaces, which can be used to create reusable, modular code.
- 3) **Modules**: TypeScript's module system allows developers to organize their code into smaller, reusable units, and can also be used to manage dependencies between different parts of an application.

- 4) **Decorators**: TypeScript allows developers to attach additional behavior to classes, methods, and properties using decorators, which are a way to add metadata to class declarations.
- 5) **Compiler**: TypeScript code is transpiled(compiled) to JavaScript, which means that it can run on any platform that supports JavaScript.
- 6) **IntelliSense**: TypeScript provides IntelliSense, which is a code completion feature that suggests the right properties, methods, and parameters as the developer types.
- 7) **Type Inference**: TypeScript can infer types of variables and properties based on the value assigned to them, which can help reduce the amount of boilerplate code that needs to be written.

Why do people hate working in JavaScript?

APPLICATION SCALE

- Not designed as a programming language for big applications
- Does not have Static Typing
- Lacks structural mechanisms like classes, modules and interfaces

PROBLEMS

- Variables are untyped and dynamic. Easy to get wrong.
- Its scope looks like C# level but does not work at block level.
- Object Inheritance is hard. Not Class inheritance.
- Syntax complicated. So nobody really does it.

TypeScript Vs JavaScript

TypeScript has filled many gaps in JavaScript by providing several features that are not available in the native language. Some of the ways in which TypeScript has filled the gaps of JavaScript include:

- A. **Optional static typing**: TypeScript provides optional static typing, which allows developers to specify the data types of variables, function parameters, and return values. This can help to catch type errors at compile-time, rather than at runtime, and make code more self-documenting.
- B. **Class and Interface**: TypeScript provides a class-based object-oriented programming model, with features such as classes, interfaces, and inheritance. This can make it easier to write reusable, modular code and organize large-scale projects.

C. **Modules**: TypeScript provides a way to organize code using modules, which can help to avoid naming conflicts and make it easier to organize large projects.

D. **Type Inference**: TypeScript can infer types in some cases, which can help to reduce the amount of boilerplate code that needs to be written.

E. **Better compatibility with other languages**: TypeScript's syntax and features are designed to be familiar to developers who are used to working with other strongly typed languages like Java or C#.

SIMILAR ALTERNATIVES

- For Hardcore JavaScript development
- Some JavaScript preprocessors that compile to JavaScript are:



CoffeeScript
coffeescript.org
custom lang.



Dart
<http://dartlang.org>
custom lang.
by Google



Script#
github.com
C#

How to Install?

- Via npm(Node.js package manager)

```
1 $ npm install -g typescript
2 $ npm view typescript version
3 npm http GET https://registry.npmjs.org/typescript
4 npm http 304 https://registry.npmjs.org/typescript
5 0.8.1-1
```

By Installing TS's Visual Studio plugins

- Just download any appropriate Visual Studio from Microsoft, You are all set to go.

Some Basic Programs in TypeScript

1.HELLO WORLD

```
console.log("Hello, World!");
```

2.VARIABLE DECLARATION

```
var message:string = "Hello World"  
console.log(message)
```

On Compiling, It will generate the following JavaScript code.

```
//Generated by typescript 1.8.10  
var message = "Hello World";  
console.log(message);
```

DISADVANTAGES OF TYPESCRIPT

While TypeScript has many advantages, there are also some potential disadvantages to using it:

1. **Increased complexity**: The added features of TypeScript, such as classes, interfaces and decorators, can make the code more complex and harder to understand for developers who are not familiar with these concepts.
2. **Additional setup**: Setting up a TypeScript development environment can be more complex than setting up a plain JavaScript environment, and may require additional tools and configurations.
3. **Increased file size**: TypeScript needs to be transpiled (compiled) to JavaScript, this process can increase the file size of the final JavaScript code, which can make it slower to load and execute.

4. **Learning curve**: While TypeScript is a typed superset of JavaScript, developers who are not familiar with static typing or other features of TypeScript may need to invest time in learning the language before they can start using it effectively.
5. **Slower development**: TypeScript's type checking and other features can slow down development by adding additional steps to the development process, such as compiling or transpiling the code.
6. **Limited browser support**: Since TypeScript needs to be transpiled to JavaScript to run in browser, older browser which does not support the latest ECMAScript version may have compatibility issues.
7. **Limited library support**: Some popular JavaScript libraries may not have TypeScript definitions available, which can make it harder to use them in TypeScript projects.
8. **Additional cost**: TypeScript is an open-source language, but using it may require additional costs such as additional dev-hours, training costs, or additional tooling and infrastructure costs.

POPULARITY OF TYPESCRIPT

TypeScript has gained popularity in recent years as more and more developers have started using it for building large-scale web applications and projects. Some of the reasons for its popularity are its optional static typing, improved code quality and better developer experience.

- ❖ In 2020, GitHub's annual State of the Octoverse report, TypeScript was included in the top 10 fastest-growing languages by repository contributors, and was the second most popular language on GitHub.
- ❖ According to the 2019 Stack Overflow Developer Survey, TypeScript is the most loved language by developers for the third year in a row, with 63.9% of TypeScript developers expressing interest in continuing to develop with the language.
- ❖ TypeScript is also being used by many big companies like Asana, Airbnb, Asos, Microsoft, AWS, etc.

In summary, TypeScript has gained a lot of popularity among developers, open-source community and big companies, and it is expected to continue to grow in popularity in the future.

CONCLUSION

Whether or not to use TypeScript will depend on the specific needs of a project and the preferences of the development team. For projects that require large-scale scalability, better developer experience and better code quality, TypeScript can be a good choice. However, for small projects or projects that are built by developers who are not familiar with TypeScript, it may be more appropriate to stick with plain JavaScript.

Thank you

Sree Sharvesh S S

CB.EN.U4CYS22061

TIFAC-CORE in Cybersecurity

Amrita Vishwa Vidyapeetham