

9 REASONS TO TRY TYPESCRIPT IN 2018

By DANIEL JEBARAJ

INTRODUCTION

Should you use JavaScript or TypeScript? The answer depends on what you're building and what you're trying to achieve. However, if you're not already familiar with JavaScript (and even if you are), there are several reasons why you'll probably want to try TypeScript, nine of which we address in this white paper.

WHAT IS TYPESCRIPT?

TypeScript is an open-source programming language, created and contributed by the open-source community of Microsoft. It is a syntactical superset of JavaScript, which means it delivers all the benefits of JavaScript while minimizing JavaScript's weaknesses. TypeScript's [primary goal](#) is to “statically identify constructs that are likely to be errors.”

TypeScript layers type-safe code on top of JavaScript so developers don't have to type-check their code manually. Code checking is handled in the code editor automatically, unlike with JavaScript. The code editor uses IntelliSense, which enables developers to code faster by using code completions, quick info, and parameter info on methods.

In fact, tooling is one of TypeScript's main benefits because it enhances developer productivity by providing advanced autocompletion, code navigation, and refactoring. TypeScript also gives developers the option to type their objects statically so that variables, parameters, etc., are known by the compiler at compile time.

Importantly, TypeScript compiles to plain JavaScript that runs on any browser or in any JavaScript engine that supports ES3 or newer. Because TypeScript is a superset of JavaScript, it can transpile (compile) to create readable JavaScript.

Developers already familiar with a type-strict language, such as C#, can easily take advantage of TypeScript. In fact, the learning curve is considerably lower than moving straight from C# to JavaScript. TypeScript's development

interface feels familiar to .NET developers because it provides all the IntelliSense productivity gains they've come to expect.

If you're still wondering whether you should try TypeScript, perhaps one of the following nine reasons will inspire you.

#1: TypeScript is easy to read and understand

TypeScript allows developers to become immediately productive. The static types and function definitions make it easy to understand how it works, so coding is relatively fast and easy. Developers never have to guess what type the variables are. TypeScript provides all the help you need. Similarly, if you have a question while working with code, simply hover over the code to get a description of what you're looking at. IntelliSense will always tell you what the bits are as you move through the code.

#2: It's open source

Everything you need to become immediately productive with TypeScript is available on GitHub, including an easy-to-follow handbook and tutorials. Because TypeScript is open source, you won't be bound to a particular vendor or product. Since no one entity owns or controls it, anyone in the Github community can contribute to it or "fork" it, which means creating a new version from the source code.

The TypeScript community stays current with ECMAScript, so you never have to worry about disconnects between TypeScript and JavaScript. If you want to understand the basic functionality, or advanced functionality, [the TypeScript wiki](#) has it readily available. The wiki also includes tutorials, tool information, and news.

#3: TypeScript provides JavaScript interoperability

TypeScript enables bidirectional TypeScript/JavaScript code use, so it simplifies

collaboration between TypeScript and JavaScript developers. As a superset of JavaScript, it is inherently interoperable with JavaScript.

For example, if you're building a TypeScript application and want to consume JavaScript code or use JavaScript libraries, you can do so as if they were TypeScript assets—no extra work required. Conversely, if you're writing in JavaScript, you can call TypeScript code. In addition, you can convert JavaScript code to TypeScript code simply by changing the extension from .js to .ts.

One of TypeScript's biggest benefits is the ease with which JavaScript code can be written and understood. TypeScript's IntelliSense capabilities enable developers to look at their own code, or even someone else's JavaScript code, and understand it immediately.

#4: It compiles down to a version of JavaScript that runs on all browsers

ES compliance is built into TypeScript, from ES3 to ES 2017, so it's easy to compile to newer or previous versions. That's particularly helpful if you're building an app that needs to be capable of running on all browsers.

#5: Strong tooling and IDE support

As a Microsoft contribution, Visual Studio and Microsoft's open-source IDE, Visual Studio Code, work natively with TypeScript. Microsoft IDE use is not a requirement, however. You can use Eclipse, Angular, WebStorm, Atom, NetBeans, and CodeAnywhere. TypeScript can also be integrated with build tools, including Apache Maven and Gradle.

#6: TypeScript is well-suited to medium and large applications

If you're writing a small program, such as a proof of concept, you can quickly write it in JavaScript—all you need is Notepad and a browser.

However, when your code exceeds a couple of Notepad pages, all that text can quickly become overwhelming. At that point, you'll want the convenience of IntelliSense and all the other help TypeScript and an IDE (particularly Visual Studio and Visual Studio Code) provide. Rather than reading through the text to understand it, you'll be alerted to the mistakes you're making in context. Similarly, if you can't recall a method name, IntelliSense will provide you with a list of options. TypeScript's refactoring, type checks, and IntelliSense capabilities collectively enable a great developer experience.

TypeScript also makes it easy to scale the number of developers working on a project. Even if the new developers don't know anything about the code, they can start to work with it and be immediately productive. And, if you want to use a method from another module, the signature will tell you what you need to know.

#7: TypeScript saves precious development time

With TypeScript, type checking is done automatically, so you can code more accurately and faster. For example, if a JavaScript developer creates a variable and assigns a string to it, the variable cannot be defined as something else, such as a number, later. JavaScript will simply blow up. There's no way to look at the method to see what parameters it expects. So one has to go back through the code, study it, and then apply the variable. Experienced JavaScript developers have created workarounds to deal with such limitations. However, new JavaScript developers often find JavaScript's complexity daunting. TypeScript allows variables to be whatever the developer chooses, which eliminates unnecessary guesswork and saves time, especially for developers building medium and large applications.

#8: It's the language of Angular

TypeScript is the language of Angular for a couple of reasons. First, Angular was written entirely in TypeScript. Angular's TypeScript-centricity enables TypeScript and Angular to be used together seamlessly.

Second, TypeScript's tooling simplifies development. As Victor Savkin, co-founder of Narwhal Technologies, [said in a blog post](#), "The fact that IntelliSense and basic refactorings (e.g., rename a symbol) are reliable makes a huge impact on the process of writing and especially refactoring code."

The fact that Angular is very popular and widely used, and that TypeScript was selected to handle all Angular's development, gives TypeScript additional credibility. Also, the team that led the development of Angular was the Angular team at Google!

#9: TypeScript has been adopted by top frameworks

Angular, React, and Vue JavaScript have all adopted TypeScript, which has helped fuel TypeScript's popularity. Each of the frameworks aims to minimize the complexities of JavaScript development by providing developers with a paradigm, since the true nature of JavaScript is difficult and complex.

React has been compatible with TypeScript for awhile. Developers using React can use TypeScript to define complex type definitions as user interfaces. TypeScript's strict type checks reduce the amount of errors in such endeavors.

Vue began supporting TypeScript in version 2.5 and will deepen its support in the next few versions. Vue ships with official type declarations for TypeScript, including in Vue core, vue-router, and vuex. TypeScript and Vue can be used together without the need for additional tooling. In the near future, Vue will provide an option.

THREE REASONS NOT TO TRY TYPESCRIPT

TypeScript isn't a panacea, but it does provide an elegant way to write application-scale JavaScript code quickly and easily. If TypeScript still doesn't interest you, it may be for one of these reasons:

- Your team is a group of seasoned professionals who have already addressed JavaScript's limitations. You consider TypeScript overkill.

- You absolutely, positively, must have the fastest possible JavaScript running on the client. Although TypeScript will transpile to JavaScript, and the transpilation is done very well, TypeScript is optimized for readability, not speed.
- You think your JavaScript code is superior to TypeScript code, regardless of the app's size or complexity.

CONCLUSION

If you want to be more productive writing medium or large JavaScript applications, TypeScript can help you whether you have experience writing JavaScript or not. If you're new to JavaScript, TypeScript can make you immediately productive and help you understand JavaScript code better than you could on your own without years of experience.

Like any other language, TypeScript isn't the solution for all development problems, but if you want to simplify JavaScript programming and improve your productivity, give TypeScript a try.

ADDITIONAL RESOURCES

Packt Publishing and Syncfusion both offer several valuable resources that can help make you even more productive with TypeScript. While both organizations offer more TypeScript resources than are listed here, the following will help you jump-start your efforts.

Top books and courses from Packt Publishing:

[Mastering TypeScript, Second Edition](#) (video)

[TypeScript High Performance](#) (book)

[TypeScript 2.x by Example](#) (book)

[TypeScript for Angular Developers](#) (book)

[Learning TypeScript Fundamentals](#) (video)

Resources and tools from Syncfusion:

[Syncfusion Community License Edition](#) (product)

[Online TypeScript samples](#) (demo)

[A Tour of Syncfusion's TypeScript Components](#) (webinar recording)

[Essential JS 2 \(2018 Volume One\)](#) (video)

[TypeScript Succinctly](#) (book)

