

TypeScript



Press Space for next page →

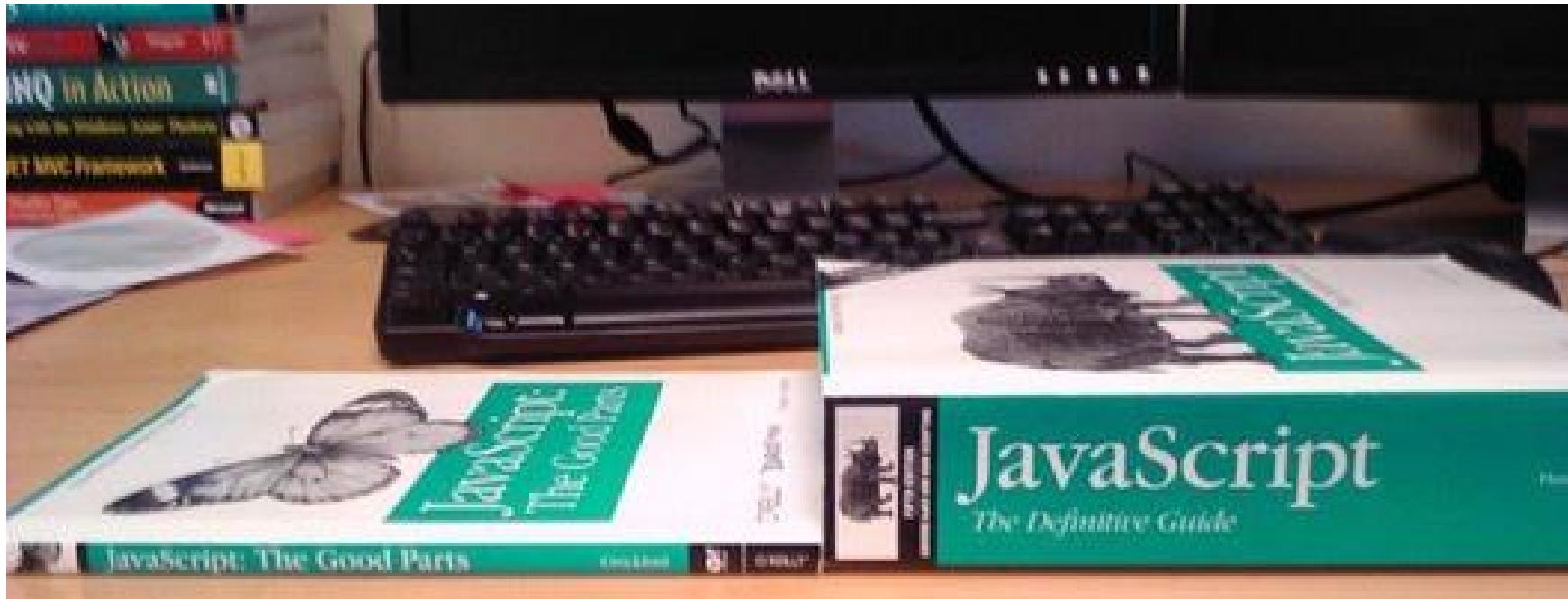


What is TypeScript?

- Free and open source, strongly supported by Microsoft
- Based on ECMAScript 4 + ECMAScript 6
- Created by the father of C# Anders Hejlsberg
- A superset of JavaScript
- To answer why we need JavaScript+, we need to understand what's wrong with vanilla JavaScript

What is the problem ?

- Why do people hate working in JavaScript?



> Using state of the art software engineering practices ;)

What is the problem ?

- JS is designed for small things
 - We now use to do big things
 - But JavaScript is not suited for building large applications
- { Your JavaScript code gets complex; it becomes extremely unwieldy

Let's look at TypeScript

- To get started with TypeScript, grab it from <http://typescriptlang.org>
- Let's look at TypeScript, first the core concept...

TypeScript - first glance - optional strong type checking

```
// js function
f(x, y) { return x * y;}
// ts function
f(x : number, y : number) : number { return x * y;}
// Type information is enforced in design and
// compile time, but removed at runtime
```

TypeScript features

- Static Type Checking
- Modules and Export
- Interface and Class for traditional Object Oriented Programming
- Works with all your existing JavaScript libraries
- Low learning overhead compared to similar JavaScript systems (CoffeeScript or Dart)
- Amazing Visual Studio, visual code studio, eclipse or IntelliJ tooling
- Outstanding team and refactoring scenarios

Base types

Type	Keyword	Description
Number	number	Used to represent whole numbers and fractions (floating point).).
String	string	Unicode character sequence.
Boolean	boolean	Logical values, true and false.

Values

Type	Keyword	Values
Number	number	1.5, 2, 3.14
String	string	"To be or not ...".
Boolean	boolean	true, false

Special types

Type	Keyword	Description
Void	void	Used as the return type of a function that does not return a value.
Null	null	Represents the intentional absence of an object's value.
Undefined	undefined	Value of uninitialized variables
Any	any	an any variable can be assigned to any data type:
Unknown	unknown	You can assign any type to unknown, but you need to perform a type check to operate on it.

Special types

Using any is not recommended, as it removes the type restriction that is the primary reason for using TypeScript. Generally, any is used to speed things up or get around an error. A **very bad idea!**

Here's a quick rule to help you understand the difference between any and unknown:

- You can assign any type to unknown, but you need to perform a type check to operate on it.
- You can assign any type to any and perform any operation.

Summary - why TypeScript ? (Expected Benefits)

- Have to learn one more thing - there is a learning curve, very easy if you already know JavaScript, or if you know C# or Java very well.
- You still have to learn JavaScript - Understanding how TypeScript converts to JavaScript will teach you better JavaScript
- Some definition files don't exist or incomplete, but I think this will vanish very quickly. These aren't hard to write if you really need something.
- Modules and classes enable large projects and code reuse
- Compile-time checking prevents errors
- Definition files for common JavaScript libraries makes them very easy to work with, and provides strong type checking
- Source map debugging makes debug easy

Initial conclusion - if I have to make a decision for you...

- If you see yourself using more JavaScript. You have to look at TypeScript.
- If you and your team has to work on JavaScript together, you have to look at TypeScript.
- Once you've done the initial hard work and converted a project. You can't stand going back.