TypeScript 5 Crash Course for JavaScript Developers

Understanding the Why of TypeScript

Chris B. Behrens
Senior Software Architect

@chrisbbehrens

Our GitHub Repo

https://github.com/FeynmanFan/pstypescriptcc

I Love TypeScript

I'm about 50/50 on Microsoft

I've been working with JavaScript since the very beginning

How JavaScript Developed

Possibly the most successful programming language of all time

But it developed in an ad hoc way

Good, but also bad

TypeScript is an effort to place a design ethos atop JavaScript

Especially a type system – "Type"Script

Allow developers to leverage the language tools they know

So, Is TypeScript Just C# on top of JavaScript?

No

Another gem from Anders Hejlsburg

TypeScript is enhancing JS, not supplanting it

"Pythonic"

TypeScript is JavaScript-ic

Preserving what design ethos there is

Does Typing Matter?

"Static" or "Strong" Typing...not really the same thing

Focus on "type coercion"

Simple Type Coercion in JavaScript

```
var x = 5;
x = “drum solo”;
console.log(x);
```

Type Coercion

Sometimes it's useful to coerce to a new type

Strict typing is *commitment*

I think that type coercion is often a code smell

Why else recommend const over var?

Type coercion happens more often due to an error in coding than by design with loose typing

[1,2,3]

Unit Testing and Typing

```
$employee.Birthday = “This is not a birthday and will break the code that tries
to use it”;

employee.Birthday = “This is not a birthday and will break the code that tries to
use it”;

employee.Birthday = “This is not a birthday and will break the code that tries to
use it”;

employee.Birthday =”2024-06-01”;

employee.Birthday = DateTime.Parse(”2024-06-01”);

// Does TypeScript enforce strong typing?
```

TypeScript Type Enforcement

```
x = 5;
x = “drum solo”;

// TypeScript type enforcement happens at compilation time
// only, and only for TypeScript code.
```

Compilation vs. Transpilation

The JavaScript emitted from processing TypeScript is at the same level of abstraction as TypeScript.

Version Controlling TypeScript and JavaScript

Keeping Binaries in Version Control Stinks

Craft your .gitignore

Keep the junk out of version control

Some stuff changes every time we build – that's junk

What is the principle?

We version control only the purely non-deterministic elements of the system.

Deterministic

```
346 x 183
= 63318

// Today, tomorrow, now and forever
// 63318 is the deterministic outcome of 346 * 183
```

You're Killing Me, Behrens

We're used to storing JavaScript in version control

Now, you'll only store TypeScript

Life's too short to deal with merge conflicts

So, we'll focus on migration

An Exception to the Rule

Partial migration

You can't just add *.js to gitignore

The price of the transition

Demo: How to Version Control TypeScript

Look at some simple existing JavaScript

Install TypeScript with npm

Move it to TypeScript

Compile it to JavaScript

Look at it all from
a version control standpoint

The Checkout Scenario

We'll need to build to generate our JavaScript

Sometimes, it is as simple as a rename and a tweak

Debugging TypeScript in the Browser

The Problem

We take interactive debugging for granted in higher level languages

The execution engine needs a map of execution to source

The Solution

A map emitted by the compiler – a source map

This won't be much to look at

Demo: Debugging TypeScript

Look at our simple application in the browser tools

Emit a source map with a compiler option

Review the app again

Create a tsconfig file

Configure it to emit a source map

Look at our application in the browser with the source map

Talk about what it all means

The Source Map Specification

https://tinyurl.com/ycypjk3x

The codes are *offsets*

We version control only the purely non-deterministic elements of the system.

Source Maps and Deployment

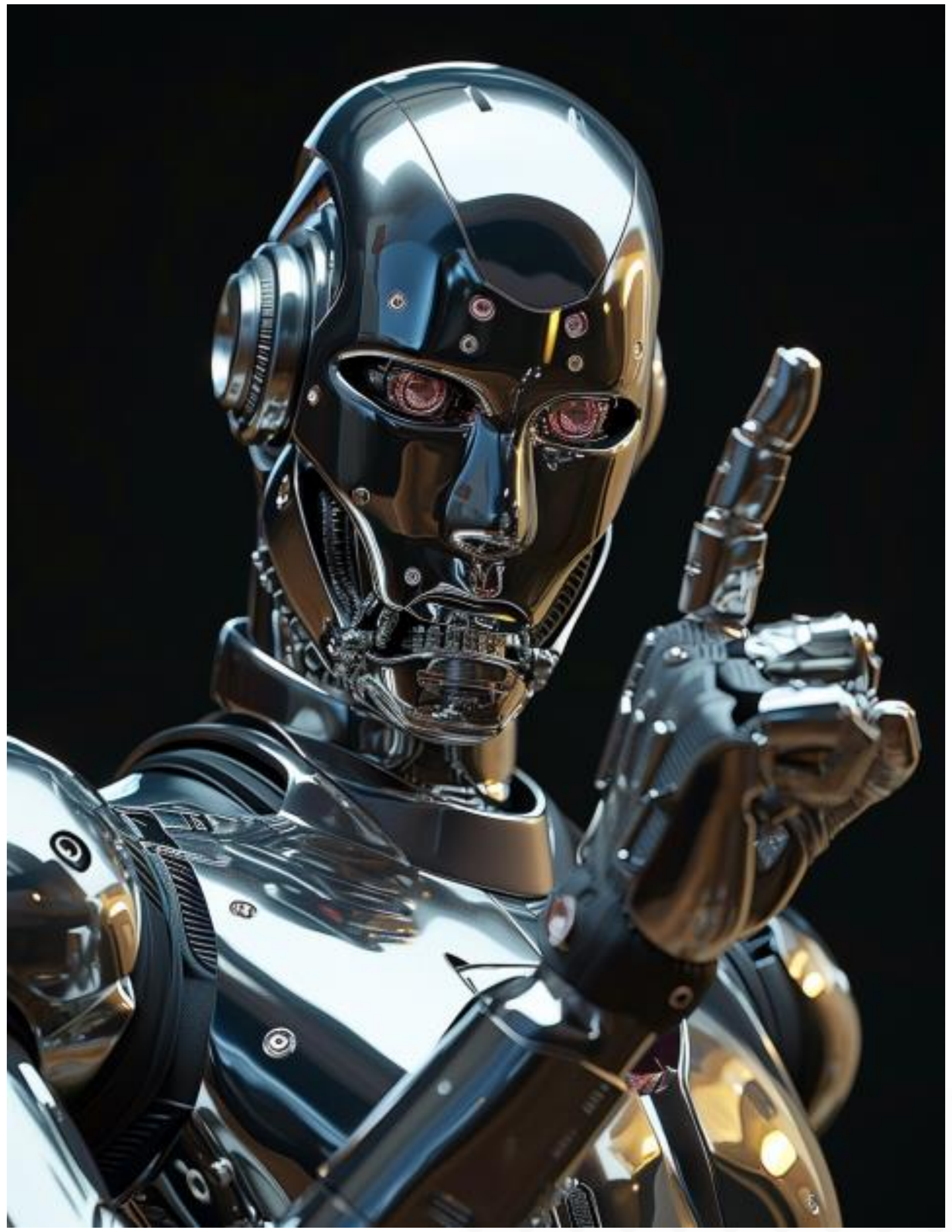
Generally, no

“Is it okay if the user has my source code?”

You're sharing your intellectual property

In the form of the TypeScript

Then Why “No”?



While it is nice for troubleshooting...

It allows attackers to know more about our system rather than less

Probably a low-priority problem

We probably don't want to just be copying everything to Production

Make sure someone has thought about it

Summary

The why of TypeScript

Proper version control of TypeScript

Debugging our TypeScript

The magic of source mapping