TypeScript

Introduction

What is TypeScript?

TypeScript!

Browser engines only can interpret pure JavaScript

JavaScripts' specification is called ECMAScript <YEAR>

TypeScript is "a layer on top" of JavaScript

After compilation, the stuff is gone and you're left with pure JavaScript

Why TypeScript?

But why?

Helps to catch serious and silly mistakes in your code

Your codebase will become well structured

Your codebase will become almost self-documenting

You'll also appreciate the improved autocompletion

But really, why?

JavaScript is dynamically typed

Static types make the code easier to work with

It will probably not matter if you code on your own... but imagine a company with 100 employees...

It's a solid tool and having it in your tool belt won't make harm

Examples of pure JavaScript

```
if (0 == "") {
  // It is! But why??
}
--------------------------------
The left operand is of the type Number.
The right operand is of the type String.
The right operand is coerced to the type Number: 0 == Number('')
which results in 0 == 0, which is true


if (1 < x < 3) {
  // True for *any* value of x!
}
--------------------------------
const tempVarA = 1 < x
const tempVarB = tempVarA < 3
So 1 < x is either true or false.
Then the next step is true < 3 or false < 3.

const obj = { width: 10, height: 15 };
// Why is this NaN? Spelling is hard!
const area = obj.width * obj.heigth;
--------------------------------
This is a typo that you'll eventually find.
Eventually.
```

So...

Since you may use an ugly style in your code (and it would still work), it makes sense to impose some kind of stricter language on your programmers for the sake of readability and maintainability.

TypeScript lets you set the types of your variables, parameters and return-values.

TypeScript

A static type-checker

```
const obj = { width: 10, height: 15 };
const area = obj.width * obj.heigth;
```

```
Property 'heigth' does not exist on type '{ width: number; height: number; }'. Did you mean 'height'?
```

```
console.log(4 / []);
```

```
The right-hand side of an arithmetic operation must be of type 'any', 'number', 'bigint' or an enum type.
```

But...

the types get erased during compilation to JavaScript!

No Reflection!

Continue here...

https://www.typescriptlang.org/docs/handbook/typescript-in-5-minutes.html