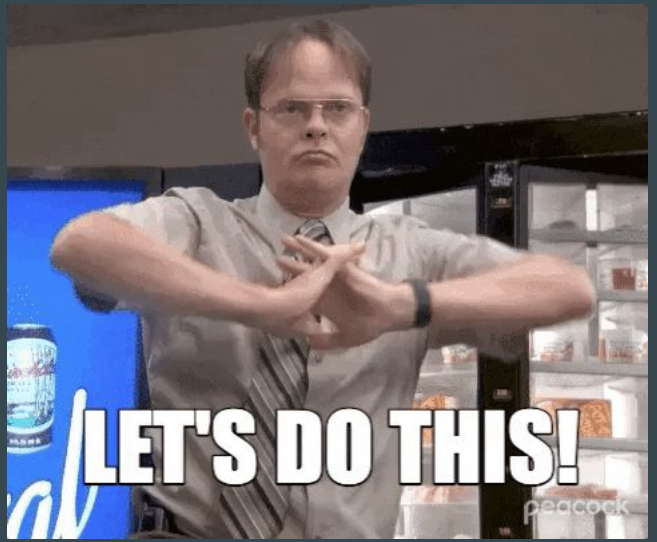
TypeScript

Intro and all you need to know before starting with TypeScript



Agenda

- What is TypeScript?
- Why TypeScript?
- Some basics
- Task

Agenda

- What is TypeScript?
- Why TypeScript?
- Some examples and a bit about types
- Task

Check if you have node, npm and tsc ready:

- *node –v*
- *npm –v*
- *tsc -v*

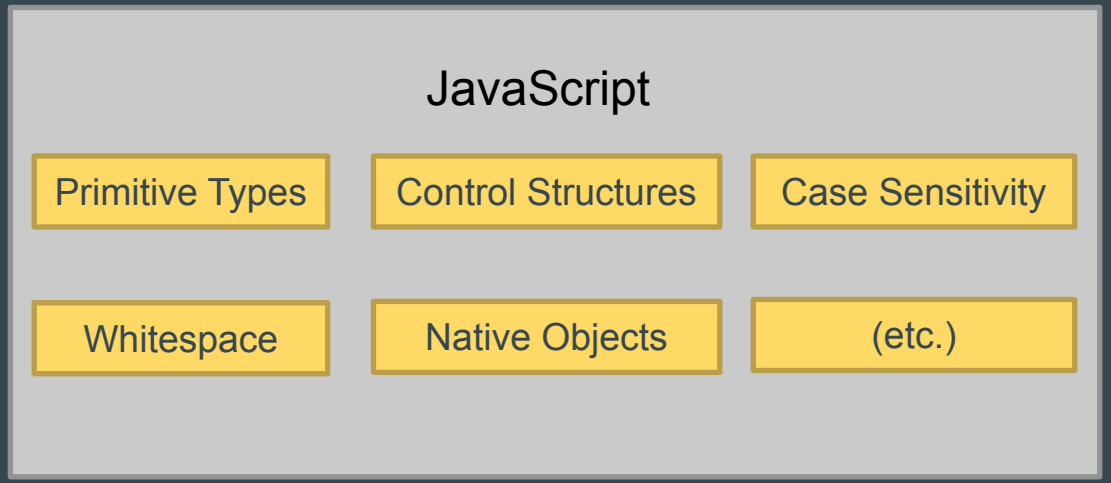
What is TypeScript?

TypeScript is **JavaScript with syntax for types.**

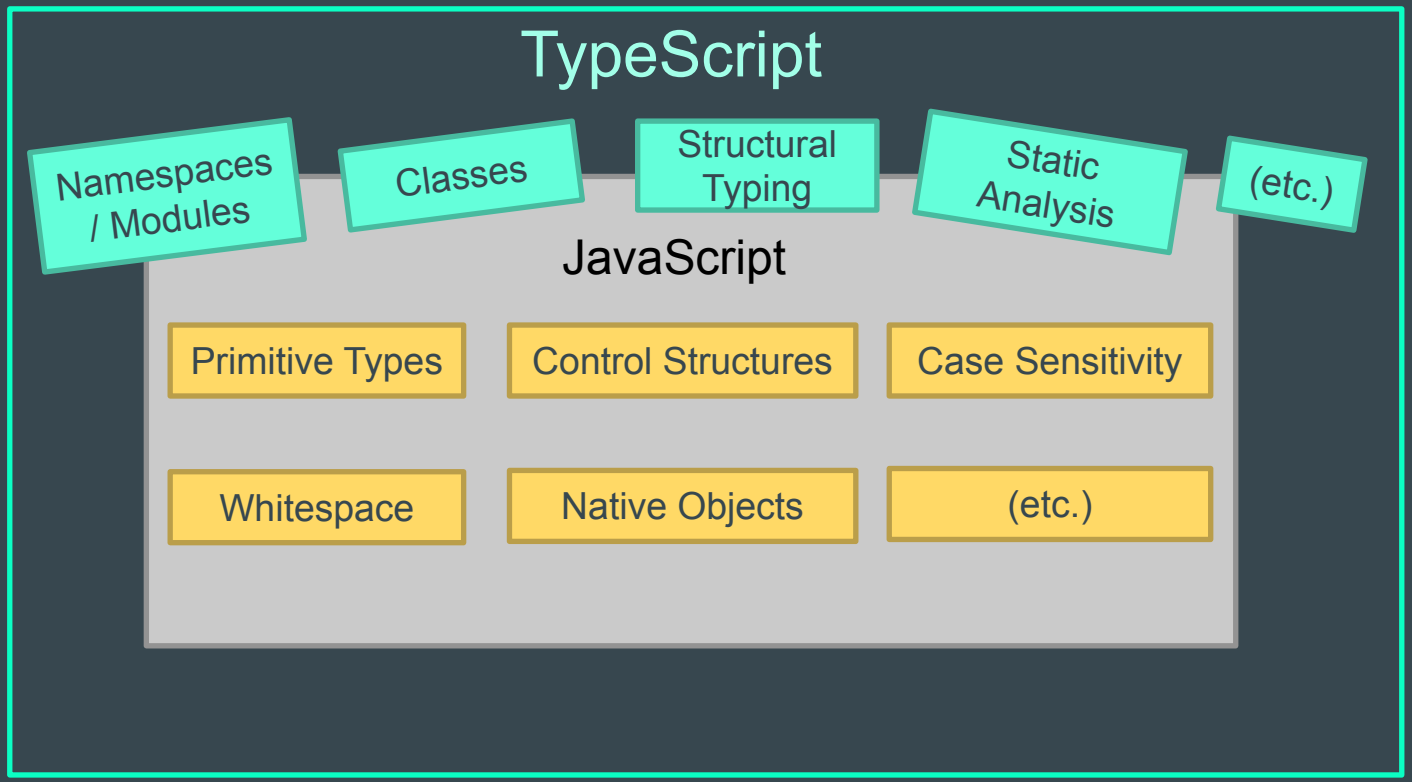
TypeScript is a strongly typed programming language that builds on JavaScript, giving you better tooling at any scale.

https://www.typescriptlang.org/

What is TypeScript?



What is TypeScript?



Can I find a specific data type for a variable in JavaScript ?

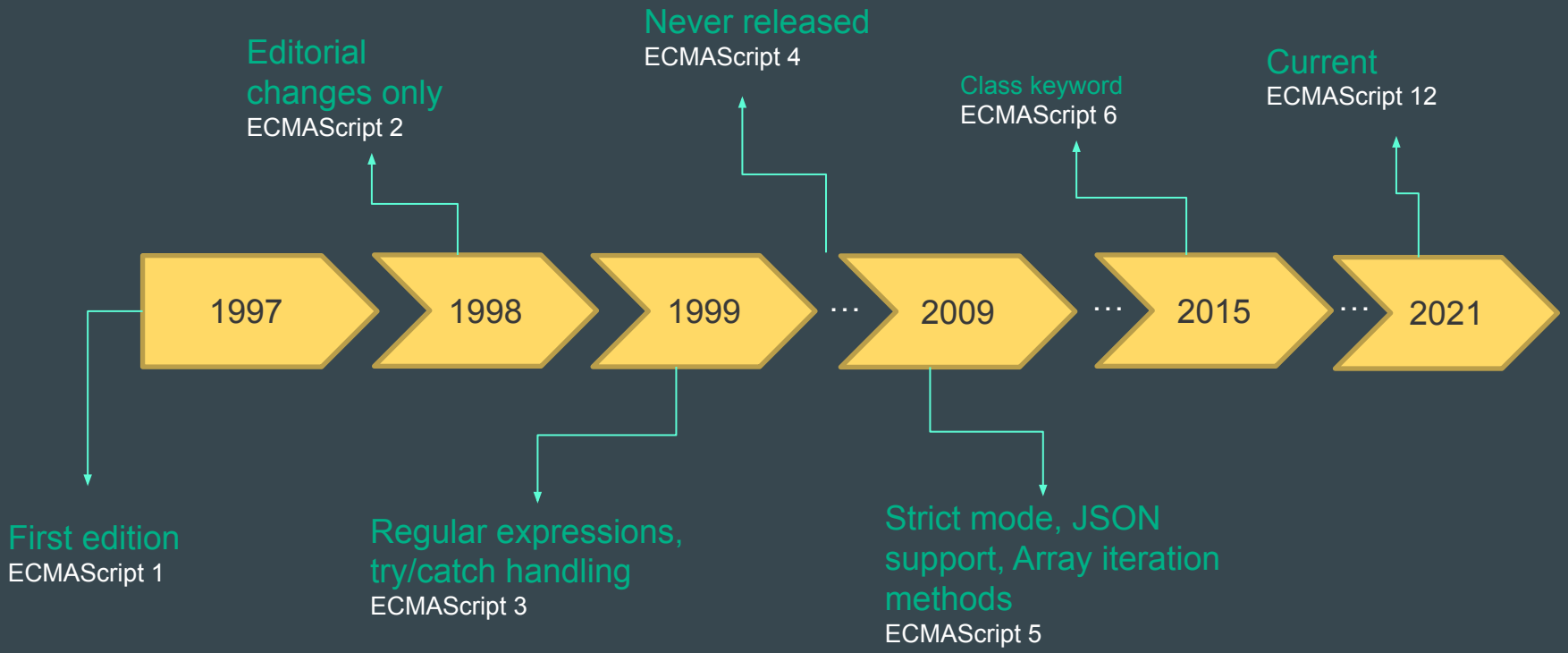
How do I organise my code into classes or modules or namespaces ?

If I'm dealing with multiple files is there a way that JS code from one file can figure out what's in another JS code file ?

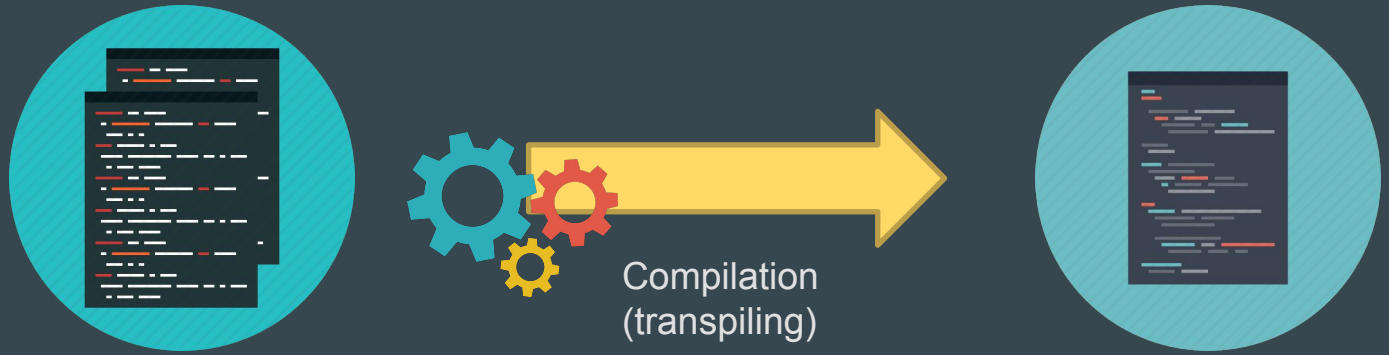
I must have some really cool debugging tools then?

Eh…

ECMAScript is a JavaScript standard meant to ensure the interoperability of web pages across different web browsers. It is standardized by Ecma International according to the document ECMA-262. - *Wikipedia*

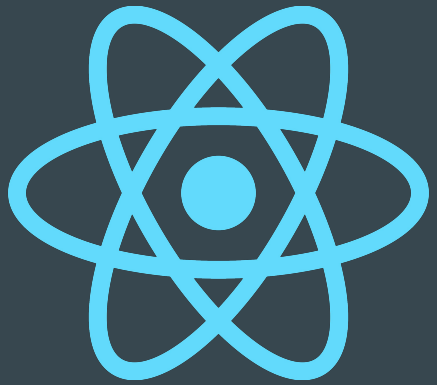


What is TypeScript?



TypeScript
Source Code

JavaScript
Source Code



...

Why TypeScript?

Why TypeScript?

Static types (variables, parameters, return types etc.)

JS

```
// Variable in JavaScript

var myVariable = "Hi, there!";

// Later in the code...

myVariable = 12345;

// Sometime later ...

myVariable = [1, 2, 3, 4, 5];

// or...

myVariable = function() {
    console.log("Hi, there!");
}
myVariable();

// and yet...

myVariable = null;
```

Why TypeScript?

Static types (variables, parameters, return types etc.)

JS

```
// Variable in JavaScript

var myVariable = "Hi, there!";

// Later in the code...

myVariable = 12345;

// Sometime later ...

myVariable = [1, 2, 3, 4, 5];

// or...

myVariable = function() {
    console.log("Hi, there!");
}
myVariable();

// and yet...

myVariable = null;
```

```
// Variable in JavaScript

var myVariable = "Hi, there!";

// Later in the code...

myVariable = 12345;

// Sometime later ...

var myVariable: string

Type '() => void' is not assignable to type 'string'. (2322)

View Problem (⌥F8)    Quick Fix... (⌘.)

myVariable = function() {
    console.log("Hi, there!");
}
myVariable();

// and yet...

myVariable = null;
```

Why TypeScript?

Organizational support (making it easier to manage a large codebase)

Classes

```ts
class Point {
  x: number;
  y: number;
}

const pt = new Point();
pt.x = 0;
pt.y = 0;
```

Namespaces / Modules

```ts
namespace Validation {
  export interface StringValidator {
    isAcceptable(s: string): boolean;
  }
}
```

```ts
// @filename: hello.ts
export default function helloWorld() {
  console.log("Hello, world!");
}
```

This is then imported via:

```ts
import helloWorld from "./hello.js";
helloWorld();
```

(etc.)

Interfaces

```ts
interface StringValidator {
  isAcceptable(s: string): boolean;
}
```

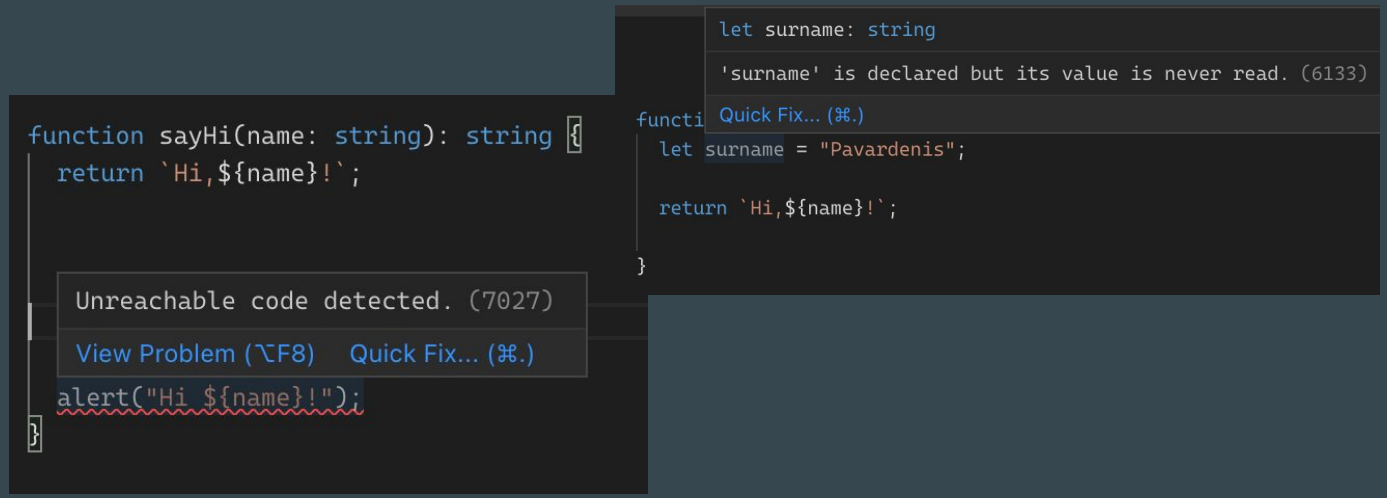
Why TypeScript?

Organizational support (making it easier to manage a large codebase)

```
interface StringValidator {
  isAcceptable(s: string): boolean;
}

let lettersRegexp = /^[A-Za-z]+$/;
let numberRegexp = /^[0-9]+$/;

class LettersOnlyValidator implements StringValidator {
  isAcceptable(s: string) {
    return lettersRegexp.test(s);
  }
}

class ZipCodeValidator implements StringValidator {
  isAcceptable(s: string) {
    return s.length === 5 && numberRegexp.test(s);
  }
}
```

Why TypeScript?

Tooling support (static type analysis, instant errors)



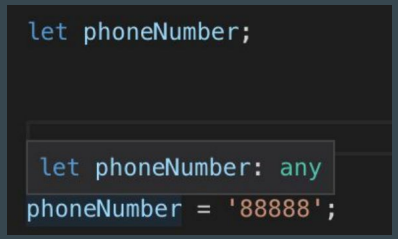
Some examples...

Basics

Static types (variables, parameters, return types etc.)

❌ **Don't** use any as a type unless you are in the process of migrating a JavaScript project to TypeScript. The compiler *effectively* treats any as "please turn off type checking for this thing"

If type is not supplied when declaring variable, it will by type "any" by default.



https://www.typescriptlang.org/docs/handbook/declaration-files/do-s-and-don-ts.html

Why TypeScript?

Static types (variables, parameters, return types etc.)

General Types

`Number`, `String`, `Boolean`, `Symbol` and `Object`

❌ **Don't** ever use the types `Number`, `String`, `Boolean`, `Symbol`, or `Object` These types refer to non-primitive boxed objects that are almost never used appropriately in JavaScript code.

```
/* WRONG */
function reverse(s: String): String;
```

✅ **Do** use the types `number`, `string`, `boolean`, and `symbol`.

```
/* OK */
function reverse(s: string): string;
```

https://www.typescriptlang.org/docs/handbook/declaration-files/do-s-and-don-ts.html

Why TypeScript?

Static types (variables, parameters, return types etc.)

General Types

`Number`, `String`, `Boolean`, `Symbol` and `Object`

❌ **Don't** ever use the types `Number`, `String`, `Boolean`, `Symbol`, or `Object` These types refer to non-primitive boxed objects that are almost never used appropriately in JavaScript code.

```
/* WRONG */
function reverse(s: String): String;
```

✅ **Do** use the types `number`, `string`, `boolean`, and `symbol`.

```
/* OK */
function reverse(s: string): string;
```

https://www.typescriptlang.org/docs/handbook/declaration-files/do-s-and-don-ts.html

Helper

- Install TypeScript using a command (npm install –g typescript) you can also use documentation instructions (https://www.typescriptlang.org/download)
- Create a .ts file in your editor (any name you like)
- run *tsc your_file_name.ts*
- Check the code in the *your_file_name.js*

Task

Time: 10 minutes

- Create a file *.ts* (any name you like) containing a class with some properties and a constructor.
- Create a *tsconfig.json* file and configure the output directory to be my_fav_dir (or any name you like). Meaning that your generated *.js* file needs to end up in your output dir.
- Build your *.ts* file to get *.js* file.

Bonus point:

- One class property is of some your created type;
- Configure compiler to build *.js* file in es3 version.

Helper

Documentation is your friend: https://www.typescriptlang.org/

To install TypeScript use a command *npm install –g typescript* (g option will install it globally) or if you prefer some more options can be found: https://www.typescriptlang.org/download

For those struggling to install (I suggest to use the playground https://www.typescriptlang.org/play)

For those who don’t have a node please follow the instructions here: https://docs.npmjs.com/downloading-and-installing-node-js-and-npm

And Enjoy!