s|ngular

it can be done.

Tech Talk:

TypeScript 101

Moises Pedraza Mena - July 2021

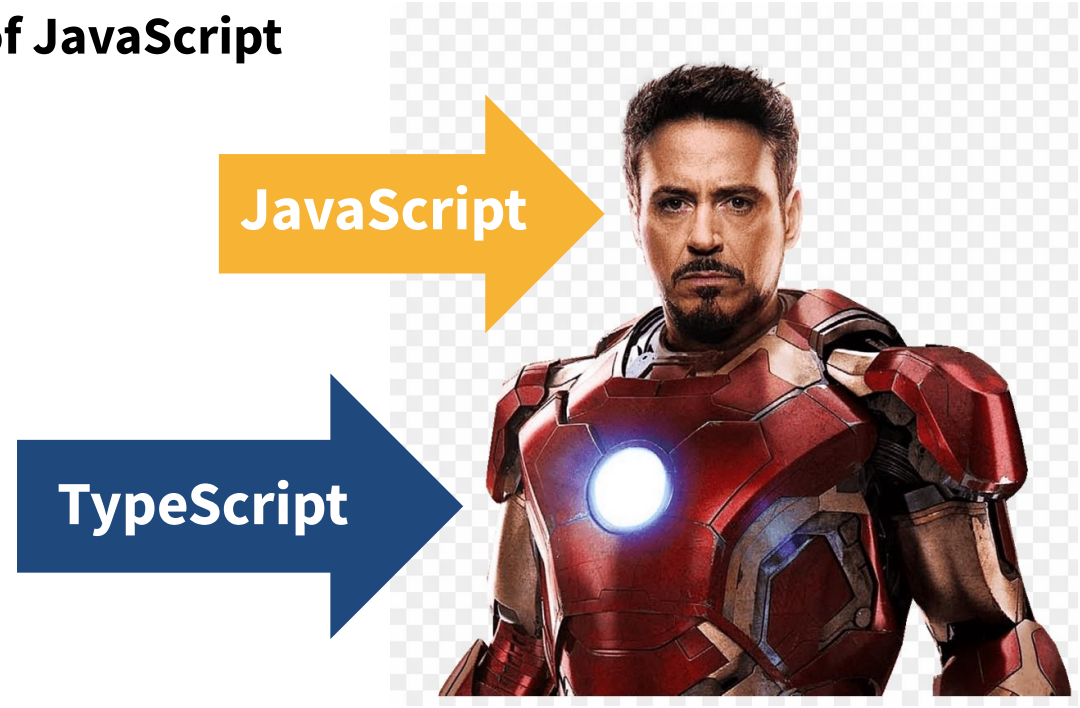
Table of Contents

What is TypeScript?

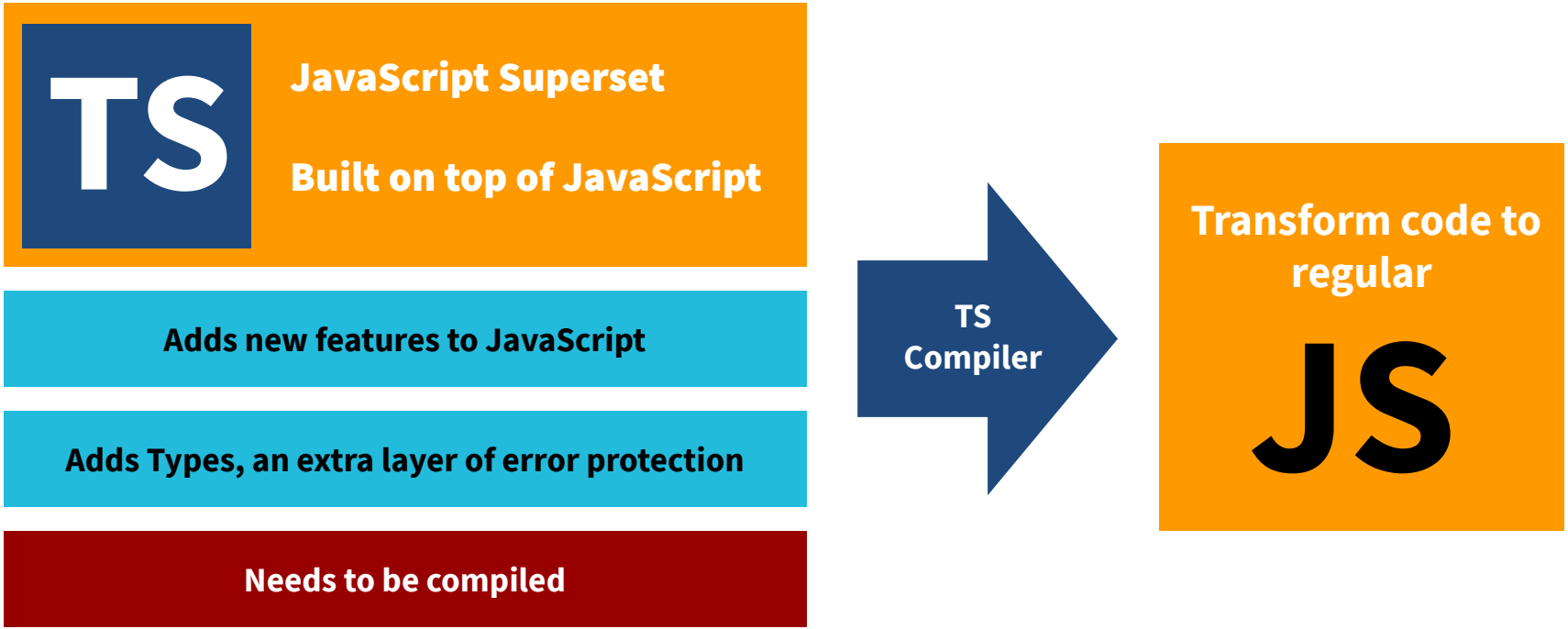
What is TypeScript?

TypeScript is typed superset of JavaScript

You can picture it like JavaScript with Superpowers, it helps you write cleaner code and helps you catch errors at the development time.



How does TypeScript relates to Javascript?



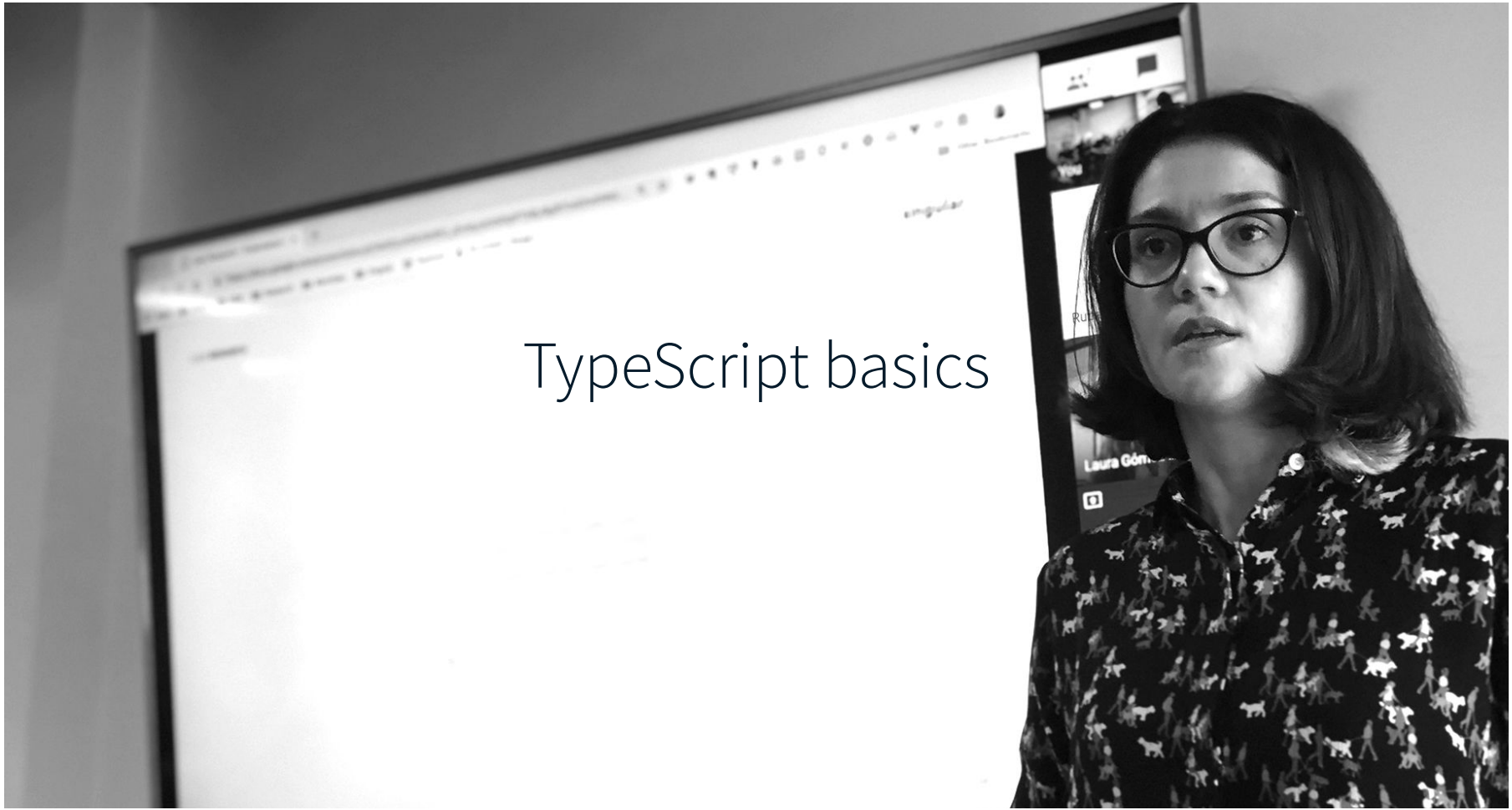
Why do i want to use Types in JS?... One simple example.

```
const API_RES = {
  payroll: '12345',
  incentives: '5400',
}

const getTotalCash = (payroll, incentives) => {
  const total = payroll + incentives;
  return total;
}

console.log(getTotalCash(API_RES.payroll, API_RES.incentives));
// The response it's going to be '123455400'
```

Let's have a look at this simple case, **JavaScript does not check types**, we are trying to do an addition between two strings, instead of two numbers, **thats valid JavaScript code**, instead of adding is concatenating, **not a runtime error but a logic one**, and those are hard to catch

Without types it's easy to fall in those kind of errors, **especially if you are working with several APIs** to get your results

“The goal of **TypeScript is** to be **a static type checker for JavaScript programs** - in other words, **a tool that runs before your code runs** (static) and ensures that the **types of the program are correct** (typechecked).



Types by inference

TypeScript knows JavaScript language, it will try to generate the types for you in most of the cases.

```
const name = ‘Moises’;
// For TS is equal to const name:string
```

Explicitly adding types

You can also add yourself a type to your code. the types supported by TypeScript are:

- string
- number
- boolean
- array
- enum
- void
- any

```
let currentNickName: string;
let position: number;
let visible: boolean;

currentNickName = ‘MoyArt’;
position = 1;
visible = false
```

Clases

TypeScript supports ECMAScript 2015 classes that integrate the optional type annotations support.

```
class Student {
    private name: string;
    public nickname: string;
    private age: number;
    private salary: number;

    constructor(
      name: string,
      age: number,
      salary: number
    ) {
        this.name = name;
        this.age = age;
        this.salary = salary;
    }

    toString(): string {
     return `${this.name} (${this.age})
(${this.salary})`;
    }
}
```

Interfaces

With TypeScript you can create interfaces to explicitly describe a shape, for example an object shape.

After you create the interface you can assign it to an object the same way you assign a type.

```
// First you need to create or import it.
interface Student {
    name: string;
    id: number;
    age: number;
    grade: string;
    approved: boolean;
}

//Then you can use it.
const Max: Student = {
    name: 'Max',
    id: 24567,
    age: 5,
    grade: 'Kinder',
    approved: true,
}
```

Installing and using TypeScript.

First we need to make sure **we have node installed in our computer** (if you don't have it, you need to install it)

```
npm install -g typescript
```

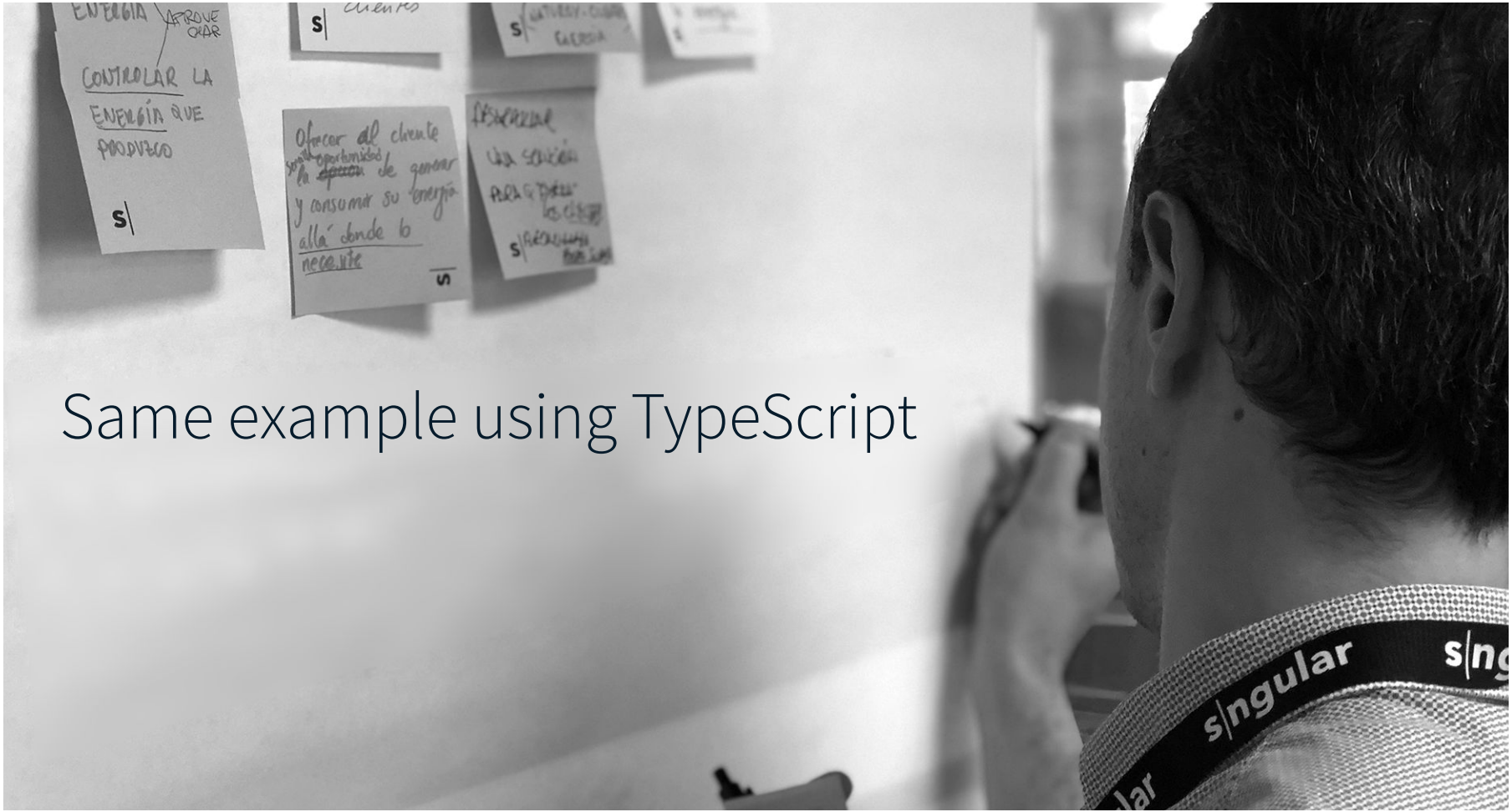
That command is going to **instal TypeScript globally** in your computer.

Now you can compile TypeScript files, you just need to **create a new .ts** file and use the following command:

```
tsc app.ts
```

Before it compiles **TypeScript is going to check if you have errors**, if everything is ok it will compile and **generate a JavaScript file with the same name**, in this case app.js, that file is the one you want to import to your html file.

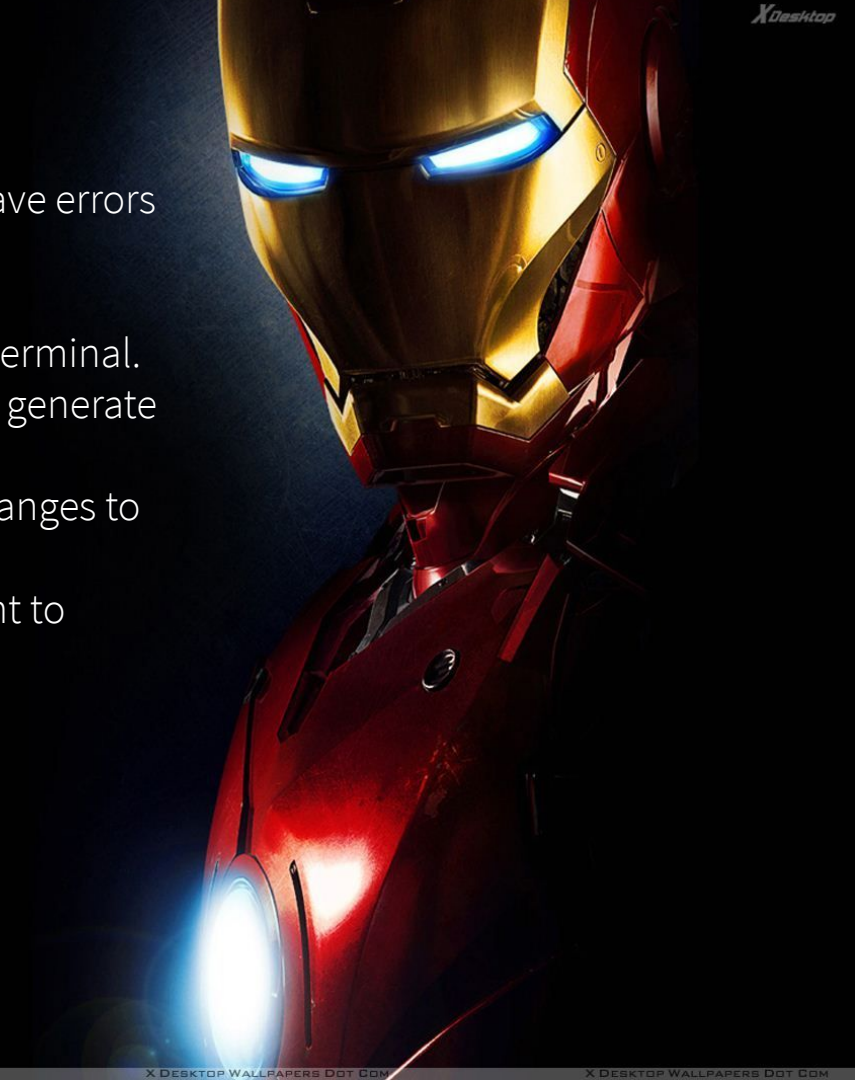
GITHUB URL: https://github.com/MoisesPedrazaSNG/typescript101



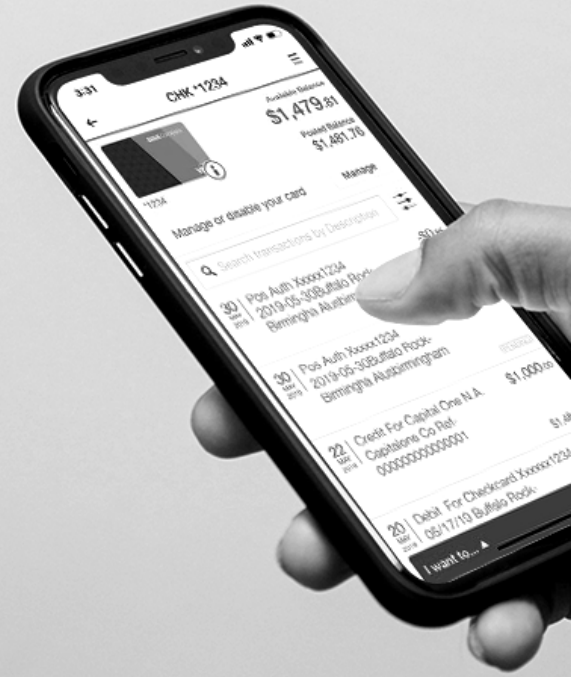
Same example using TypeScript

Important things to notice:

- When we use TypeScript it will let us know if we have errors directly on VS Code.
- It will not compile until we fix those errors.
- If we try to compile it will tell us the errors on the terminal.
- Once we fix the errors and compile the code it will generate regular JavaScript code.
- We need to compile again our code if we make changes to TypeScript to generate the JavaScript code again
- That compiled JavaScript code is the one you want to import to your HTML.



Creating a new project.



Step by step guide to create a small project

1.- We are going to create our project with npm:

```
npm init
```

2.- On the recently created project we are going to add lit-server:

```
cd new-project
npm install lit-server
```

3.- We add a start script to our package.json:

```
"scripts": {
   "start": "lit-server"
},
```

4.- Now we are going create a new html file and a new app.ts file

5.- We are going to import the compiled JS code inside the index html file, his name is the same as the TypeScript file but it ends with .js:

```
<script src=”app.js” defer></script>
```

Step by step guide to create a small project

At this point we can start adding content, to test the scaffolding we can add a console.log to the app.ts file.

6.- In the terminal compile the TypeScript File:

```
tsc app.ts
```

7.- After the compile ends, run the project:

```
npm start
```

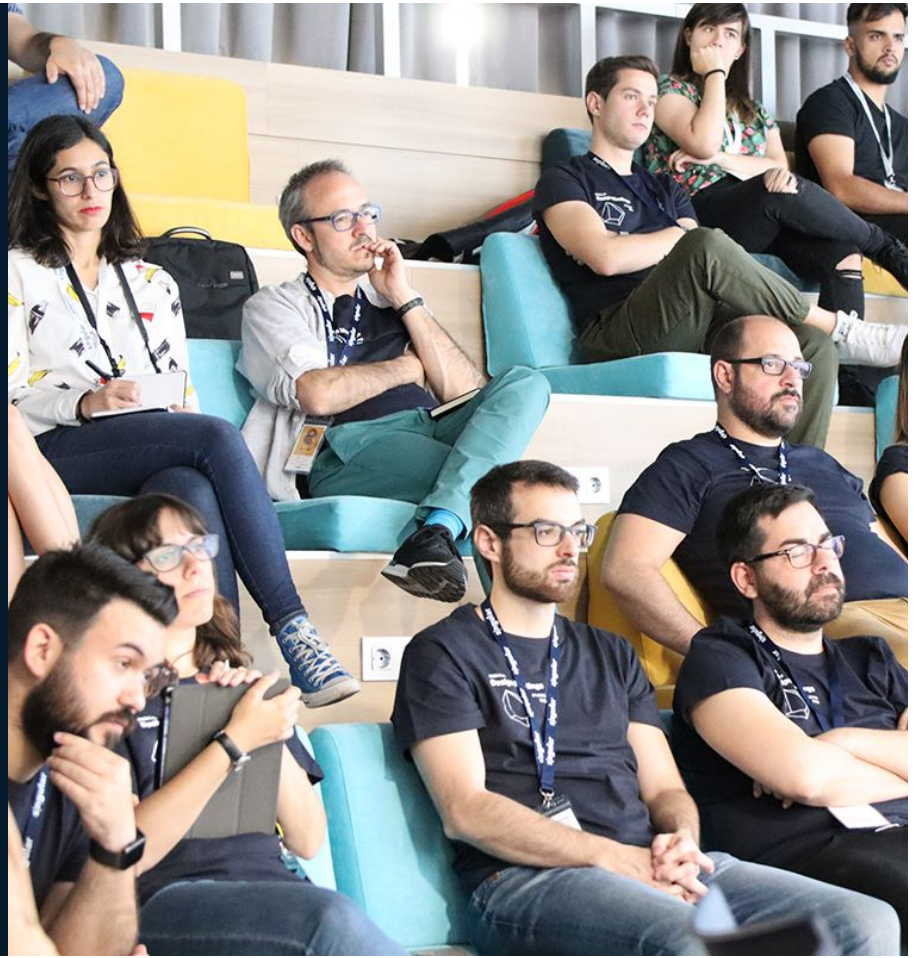
You will end up with a link with the local server running in your terminal copy and paste it to your browser.

Everytime you save the browser is going to refresh, but unfortunately, if you do some changes in your TypeScript file you will need to stop the running server and compile app. ts file again.

Learn and contribute

TypeScript Docs are Open Source, contribute:
https://github.com/microsoft/TypeScript-Website/blob/v2/packages/documentation/copy/en/get-started/TS%20for%20JS%20Programmers.md
Typescript for JS Developers: https://www.typescriptlang.org/docs/handbook/typescript-in-5-minutes.html
TypeScript Handbook: https://www.typescriptlang.org/docs/handbook/intro.html
TypeScript Playground: https://www.typescriptlang.org/play#show-examples

Q&A



Thanks

USA | Spain | Mexico | Chile
www.sngular.com