

# Những điều cơ bản cần nắm trong typescript

## Agenda

1. Static type checking
2. Types for Tooling
3. tsc - typescript compiler
4. Emitting with Errors

Nguồn: <https://www.typescriptlang.org/docs/handbook/2/basic-types.html>

## Một số lỗi phổ biến khi dùng javascript

```
> const a = undefined;  
a.length
```

```
✖ ▶ Uncaught TypeError: Cannot read property 'length' of undefined    VM130:2  
    at <anonymous>:2:3
```

```
> const a = '';  
a();
```

```
✖ ▶ Uncaught TypeError: a is not a function                            VM235:2  
    at <anonymous>:2:1
```

```
> const a = {};  
a.map(x => x + 1)
```

```
✖ ▶ Uncaught TypeError: a.map is not a function                       VM328:2  
    at <anonymous>:2:3
```

```
> const name = 'Easy Frontend';  
name.toLowerCase();
```

```
✖ ▶ Uncaught TypeError: name.toLowerCase is not a function           VM427:2  
    at <anonymous>:2:6
```

Điểm chung của các lỗi này là, **ĐỢI CHẠY LÊN MỚI THẤY LỖI** 😭

## 1. Static type checking

- Typescript giúp mình phát hiện ra lỗi ngay trong lúc code.
- Giúp tránh lỗi typo (phổ biến bên javascript)
- Tiết kiệm thời gian debug

```
12 const a = undefined;  
13 a.length;
```

✖ 04-main.ts 1 of 1 problem

Object is possibly 'undefined'. ts(2532)

```
12 const a = '';  
13 a();
```

✖ 04-main.ts 1 of 1 problem

This expression is not callable.  
Type 'String' has no call signatures. ts(2349)

```
12 const a = {};  
13 a.map(x => a + 1)
```

✖ 04-main.ts 1 of 3 problems

Property 'map' does not exist on type '{}'. ts(2339)

```
12 const name = 'Easy Frontend';  
13 name.toLowerCase();
```

✖ 04-main.ts 1 of 1 problem

Property 'toLowerCase' does not exist on type '"Easy Fr  
lib.es5.d.ts(486, 5): 'toLowerCase' is declared here.

```
12 const value = Math.random() < 0.5 ? "a" : "b";
13 if (value !== "a") {
14   // ...
15 } else if (value === "b") {
16   // Oops, unreachable
17 }
```

04-main.ts 1 of 1 problem

This condition will always return 'false' since the types 'a' and 'b' have no overlap. ts(2367)

## 2. Types for Tooling

- Ngoài việc thông báo lỗi cho mình lúc code.
- Typescript còn có thể giúp mình hạn chế lỗi bằng việc hỗ trợ auto completions / suggestions.

```
1 function findEvenNumber(numberList: number[]) {
2   if (numberList.length === 0) return undefined;
3
4   numberList.
5 }
```

length (property) Array<number>.length: number  
Gets or sets the length of the array. This is a number one higher than the highest index in the array.

push  
map  
forEach  
join  
concat  
copyWithin  
entries  
every  
fill  
filter  
find

```
1 interface Student {
2   id: string;
3   name: string;
4   gender: 'male' | 'female';
5   age: number;
6 }
7
8 const bob: Student = {
9   id: '',
10  name: '',
11  gender: '',
12  age: 1,
13 }
```

female  
male

```
1 interface Student {
2   id: string;
3   name: string;
4   gender: 'male' | 'female';
5   age: number;
6 }
7
8 const bob: Student = {
9   id: '',
10  name: '',
11  gender: 'male',
12  age: 1,
13 }
14
15 function findStudentById(studentList: Student[], studentId: string) {
16   if (studentList.length === 0) return undefined;
17
18   studentList.find(x => x.)
19 }
20 }
```

age  
gender  
id  
name

(property) Student.age<sup>x</sup>  
e: number

### 3. tsc - typescript compiler

```
npm install -g typescript
```

Bạn sẽ dùng được **tsc** sau khi cài package **typescript**

```
# Run a compile based on a backwards look through the fs for a tsconfig.json
tsc

# Emit JS for just the index.ts with the compiler defaults
tsc index.ts

# Emit JS for any .ts files in the folder src, with the default settings
tsc src/*.ts

# Emit files referenced in with the compiler settings from tsconfig.production.json
tsc --project tsconfig.production.json

# Emit d.ts files for a js file with showing compiler options which are booleans
tsc index.js --declaration --emitDeclarationOnly
```

## 4. Emitting with Errors

- Config option: `--noEmitOnError`
  - `true` thì không tạo ra file js nếu compile có lỗi.
  - `false` thì vẫn tạo ra file js dù compile có lỗi.
- 

### Series - Typescript cơ bản 🎉

- Tác giả: **Hậu Nguyễn**
- Được phát hành trên kênh youtube **Easy Frontend**.
- Tài liệu pdf và videos đều có bản quyền thuộc về Easy Frontend.
- Videos được phát hành cho fan cứng trước, public sau.
- [Đăng ký fan cứng](#) để xem series này đầy đủ và sớm nhất nhé.

#### 📞 Kết nối với mình

- ✅ Follow Facebook: <https://www.facebook.com/nvhauesmn/>
- ✅ Like Fanpage: <https://www.facebook.com/learn.easyfrontend>
- ✅ Youtube Channel: <https://www.youtube.com/easyfrontend>