



# TypeScript

# TYPESCRIPT, KÉSAKO ?

- Made by  (2012)
- Open-source on **GitHub**
- 2.9 sorti mai 2018
- Surpasse JavaScript
- Transpile

# TRANSPILE VS COMPILE

## Transpile

Convertir du code d'un langage à un autre.

## Compile

Générer depuis un code source en un langage plus bas niveau (IL, binaire, bytecode java ...)

# TS, POURQUOI FAIRE ?

- Typing
- Sécurité (~15% bug en moins)
- Auto complétion pertinente
- Dernière feature EcmaScript

# GETTING STARTED

```
$: npm install -g typescript
```

```
$: tsc --init //create tsconfig.json
```

# C'EST BIEN ET APRÈS ?

```
// Création d'une variable contenant une valeur booléenne.  
var maValeurBooleenne: boolean = false;  
  
// Création d'une variable contenant une chaîne de caractère.  
var maChaineDeCaractere: string = "Hello World";  
  
// Création d'une variable contenant un nombre.  
var monNombre: number = 1;  
  
//Création d'un array typé  
let array: Array<string> = new Array<string>();  
let array2: number[] = [];
```

# OK MAIS ENCORE

```
// Création d'une variable contenant un nombre.  
var monNombre: number = 1;  
  
var monNombre2: number = 12;  
monNombre2 = "Coucou"; // Impossible d'assigner le type de "Coucou" au type number
```

# LES UNIONS

```
let stringOrNumber: string | number = "";  
stringOrNumber = 0;  
  
stringOrNumber = false; // Impossible d'assigner false au type string|number
```



# C'EST COMPLIQUE

```
let x:any = 0;  
x = "Coucou TroyesJS";  
  
let list: any[] = [1, true, "free"];  
list[1] = 100;
```

# FUNCTION

```
function add(n1: number, n2: number): number  
{  
    return n1 + n2;  
}
```

```
add(1, 2); // = 3
```

```
add(1, "coucou"); // ERROR: le type string n'est pas attribuable au params de type number
```

# POO (INTERFACE)

```
interface Person {  
  firstName: string;  
  lastName: string;  
}  
  
function greeter(person : Person): string {  
  return "Hello, " + person.firstName + " " + person.lastName;  
}  
  
let user = { firstName: "Jane", lastName: "User" };  
💡  
const helloUser: string = greeter(user);
```

# POO (CLASS)

```
class Student {  
    public fullName: string;  
    public firstName: string;  
    public middleInitial: string;  
    public lastName: string;  
  
    constructor(firstName: string, middleInitial: string, lastName: string) {  
        this.firstName = firstName;  
        this.middleInitial = middleInitial;  
        this.lastName = lastName;  
        this.fullName = firstName + " " + middleInitial + " " + lastName;  
    }  
}
```

# POO (CLASS + INTERFACE)

```
class Student {
  fullName: string;

  constructor(public firstName: string, public middleInitial: string, public lastName: string) {
    this.fullName = firstName + " " + middleInitial + " " + lastName;
  }
}

interface Person {
  firstName: string;
  lastName: string;
}

function greeter(person : Person): string {
  return "Hello, " + person.firstName + " " + person.;
}

let user = new Student("Jane", "M.", "User");

const helloUser: string = greeter(user);
```

[ts] Identificateur attendu.

firstName

lastName (property) Person.lastName: string ⓘ

# TS PENSE À VOUS

```
$: npm install --save-dev @types/package-name
```

- DefinitelyTyped/DefinitelyTyped on Github
- \*.d.ts file

 C'EST LE FEU TON TRUC 

# C'EST LE FEU TON TRUC

## OUI

Si vous êtes une grosse équipe  
ou gros projet

## NON

Si vous êtes seul et/ou petits  
projets

Long à mettre en place

Contraignant



# RETOUR D'EXP

 Input direct et le typing

DES QUESTIONS ?