
TECHNOS LOG

02.2 - TYPESCRIPT



université
PARIS-SACLAY

POURQUOI TYPESCRIPT ?

- Ajout d'une certaine robustesse au développement
- Ce n'est pas l'unique solution (CoffeeScript / Dart)
- Support des types
- Facilité de codage et éviter des erreurs
- Utilisation des interfaces / attributs pour les classes et la notion de méthodes public et privé
- Toute la documentation Angular est en TypeScript ;)
- Notion de Transpiling
 - Compilation pour le navigateur -> transformation du code
 - Attention petite latence en mode de développement.

PRE REQUIS ENVIRONNEMENT - I

- Nodejs / npm
- Site www.typescriptlang.org

The image shows a side-by-side comparison of TypeScript and JavaScript code for a Greeter class. The left editor is titled 'Using Classes' and 'TypeScript', showing a class-based implementation. The right editor is titled 'Run' and 'JavaScript', showing the equivalent JavaScript code using a function constructor and prototype.

```
1 class Greeter {
2   greeting: string;
3   constructor(message: string) {
4     this.greeting = message;
5   }
6   greet() {
7     return "Hello, " + this.greeting;
8   }
9 }
10
11 let greeter = new Greeter("world");
12
13 let button = document.createElement('button');
14 button.textContent = "Say Hello";
15 button.onclick = function() {
16   alert(greeter.greet());
17 }
18
19 document.body.appendChild(button);
```

```
1 var Greeter = /** @class */ (function () {
2   function Greeter(message) {
3     this.greeting = message;
4   }
5   Greeter.prototype.greet = function () {
6     return "Hello, " + this.greeting;
7   };
8   return Greeter;
9 }());
10 var greeter = new Greeter("world");
11 var button = document.createElement('button');
12 button.textContent = "Say Hello";
13 button.onclick = function () {
14   alert(greeter.greet());
15 };
16 document.body.appendChild(button);
17
```

PRE REQUIS ENVIRONNEMENT - 2

- Dans votre terminal
 - `npm install -g typescript`
 - `tsc --version`
 - `tsc main.ts` → transpiling pour le navigateur
- Fichier `tsconfig.json`
 - Configuration du transpiling

TOUR D'HORIZON - FUNCTION

- Check examples/typescript/function.ts
 - Définition des types d'entrées
 - Définition des types de retours
 - Paramètres par défaut
 - Paramètres optionnels
- Check examples /typescript/fatArrow.ts
 - Raccourci d'écriture (=>)
 - non perte du this dans une fonction

TOUR D'HORIZON - CLASSE

- [Check examples/typescript/person.ts](#)
 - Plus facile de déclarer des objects
 - Gestion de l'héritage
 - Gestion des attributs private public protected
 - Utilisation des méthodes statics
 - Héritage

TOUR D'HORIZON - GENERICS

- `check examples/typescript/generics.ts`
 - Specifier des types particuliers

TOUR D'HORIZON - INTERFACES

- [check examples/typescript/interface.ts](#)
- [check examples/typescript/interface2.ts](#)
 - Gestion de la notion d'interface
 - Gestion de multiple interface