



Une formation  
**Alphorm**.com

# Formation TypeScript



Frédéric GAURAT



Une formation  
**Alphorm**.com

## Plan de la formation

### Introduction

1. Présentation du langage TypeScript
2. Les éléments du Langage TypeScript
3. Organiser son code TypeScript
4. Les particularités du langage TypeScript

### Conclusion



Une formation  
**Alphorm**.com

## Public concerné

Les développeurs et chefs de projets qui souhaitent découvrir le langage TypeScript



Une formation  
**Alphorm**.com

## Connaissances requises



**JavaScript : Maîtriser les techniques avancées**

Frédéric GAURAT

Note : ★★★★★ Vues : 22.984

24€ [Voir](#)



**JavaScript : Acquérir les fondamentaux**

Frédéric GAURAT

Note : ★★★★★ Vues : 69.111

24€ [Voir](#)

A vous de jouer !



## Présentation du projet



Frédéric GAURAT

Une formation  
**Alphorm**.com



Une formation  
**Alphorm**.com

## Plan

Le projet Todo List

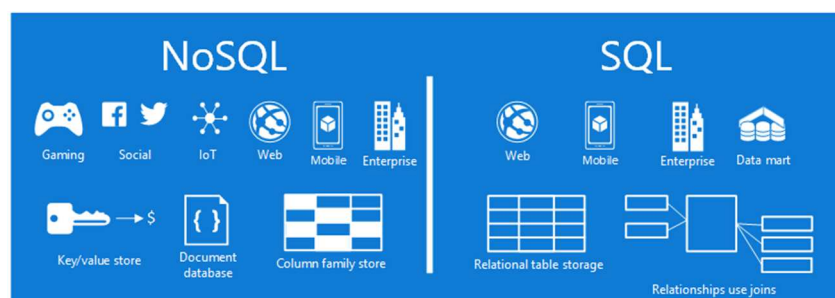
Le serveur et la base de données

Présentation des outils



Une formation  
**Alphorm**.com

## Le serveur et la BDD



Une formation  
**Alphorm**.com



Une formation  
**Alphorm**.com

## Présentation des outils

Les éditeurs

SublimeText

Atom

Microsoft Visual Studio Code

NotePad++

**Merci**



Une formation  
**Alphorm**.com

# Présentation de TypeScript



Frédéric GAURAT



Une formation  
**Alphorm**.com

## Plan

Limites de JS

TypeScript

Transpilateur (Transpiler)



Une formation  
**Alphorm**.com

# Les limites de JavaScript

Les types  
Les portées  
Les modules  
La vérification



Une formation  
**Alphorm**.com

# TypeScript

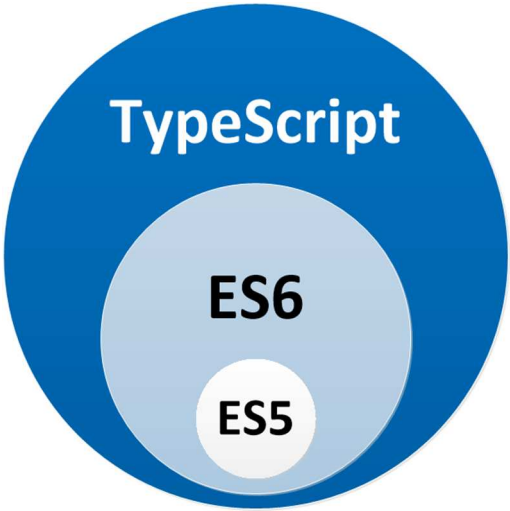
Sur-ensemble de JavaScript  
Langage conçu par Microsoft  
Utilisé pour le développement d'Angular 2





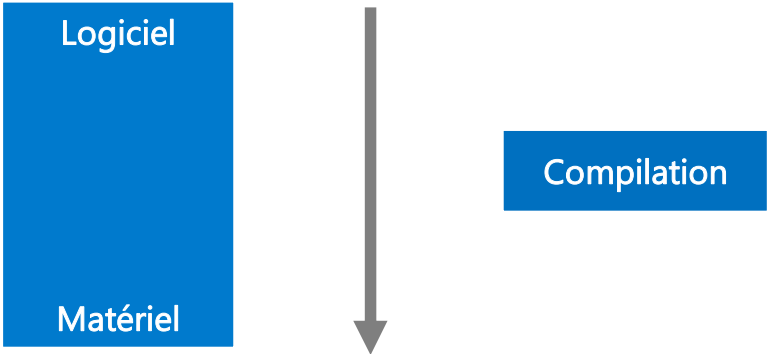
Une formation  
**Alphorm**.com

# Principe de fonctionnement



Une formation  
**Alphorm**.com

# Compilateur

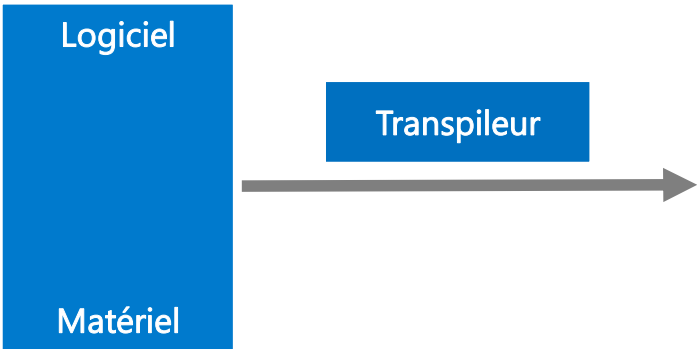






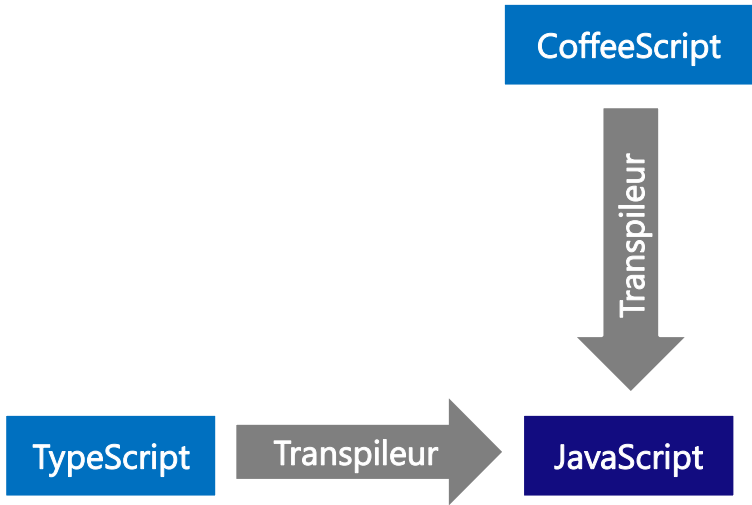
Une formation  
**Alphorm**.com

# Transpileur : avantages



Une formation  
**Alphorm**.com

# Transpileur : principe



# Merci



Une formation  
**Alphorm**.com

## Installation de TypeScript



Frédéric GAURAT



Une formation  
**Alphorm**.com

## Plan

Installation de Node.js  
Installation de TypeScript  
Installation de Visual Studio Code  
Vérification de l'installation

# Merci



Une formation  
**Alphorm**.com

# Utilisation de TypeScript



Frédéric GAURAT



Une formation  
**Alphorm**.com

## Plan

Un projet Javascript vs TypeScript  
Démonstrations



Une formation  
**Alphorm**.com

# Un projet JavaScript

- node\_modules
- bower\_components
- package.json
- bower.json
- gulpfile.js



Une formation  
**Alphorm**.com

# Un projet TypeScript

- node\_modules
- typings
- bower\_components
- package.json
- bower.json
- gulpfile.js
- tsconfig.json
- typings.json



Une formation  
**Alphorm**.com

## Démonstrations

Générer le fichier tsconfig.json

Les options du tsconfig.json

Exemple du tsconfig.json



# Merci



Une formation  
**Alphorm**.com

# Les types sous TypeScript



Frédéric GAURAT



Une formation  
**Alphorm**.com

## Plan

Notion de type

Rappel sur les portées et déclaration

Les types fondamentaux

L'affirmation de type



Une formation  
**Alphorm**.com

## Notion de type

Préciser la taille d'une donnée en mémoire  
Garantir une cohérence dans votre code

TypeScript :

```
var a:int = 3;
```



Une formation  
**Alphorm**.com

## Rappel

### JavaScript (es5)

```
1  
2   a=2;
```

```
function uneFonction(){  
    var a = 2;  
}
```

### JavaScript (es6) TypeScript

```
10  function uneAutreFonction(){  
11      var a = 2;  
12      if(true){  
13          let a = 3;  
14      }  
15  }
```

```
18  const a = 3;  
19  a=4;  
20  console.log(a)  
21  
22
```





Une formation  
**Alphorm**.com

# Les types fondamentaux

## Les chaînes

```
67 function doAlert(): void {  
68     alert("une alerte");  
69 }  
63 console.log(uneVariable);
```



Une formation  
**Alphorm**.com

# L'affirmation de type

Indiquer à TypeScript que « nous savons ce que nous faisons »

Proche du **cast** de type dans d'autres langages

```
76  
77 let uneChaine: any = "Ceci est une chaîne";  
78 let tailleChaine_1: number = (<string>uneChaine).length;  
79 let tailleChaine_2: number = (uneChaine as string).length;
```

# Merci



Une formation  
**Alphorm**.com

## Les fonctions sous TypeScript



Frédéric GAURAT



Une formation  
**Alphorm**.com

## Plan

Rappel sur les fonctions  
Les types appliqués aux fonctions  
Les paramètres  
Lambdas et this  
Surcharge de fonction



Une formation  
**Alphorm**.com

## Rappel sur les fonctions

Fonction nommée

```
3  function somme(a, b) {  
4      return a + b;  
5  }
```

Fonction anonyme

```
8  let sommeAnonyme = function(a, b) { return a+b; };
```



Une formation  
**Alphorm**.com

## Démonstrations

Les types appliqués aux fonctions

Les paramètres optionnels

Les paramètres par défaut

Nombre de paramètres variable

Lambdas et this

Surcharge de fonction



# Merci



Une formation  
**Alphorm**.com

# Les Classes sous TypeScript



Frédéric GAURAT



Une formation  
**Alphorm**.com

## Plan

Rappel sur les classes  
Démonstrations



Une formation  
**Alphorm**.com

## Rappel sur les classes

JavaScript s'appuie sur les fonctions pour la création de composant

JavaScript s'appuie sur les prototypes pour spécialiser ces composants

JavaScript ne propose pas de notion de classe



Une formation  
**Alphorm**.com

## Rappel sur les classes

TypeScript propose d'utiliser les notions de programmation orientée objet.

```
1  class Personne{
2      name : string;
3
4      constructor(a:string){
5          this.name = a;
6      }
7
8      direbonjour(){
9          console.log("bonjour je m'appelle "+this.name);
10     }
11 }
12
13 |
14 let a:Personne;
15 a= new Personne("Robert");
16 a.direbonjour();
```



Une formation  
**Alphorm**.com

## Démonstrations

- Le constructeur
- L'encapsulation
- Les accesseurs (es5)
- L'héritage
- Les propriétés static
- Les classes abstraites
- Les classes et interfaces



# Merci



Une formation  
**Alphorm**.com

# Les interfaces sous TypeScript



Frédéric GAURAT



Une formation  
**Alphorm**.com

## Plan

Les interfaces et le typage structurel  
Démonstrations





Une formation  
**Alphorm**.com

# Les interfaces et le typage structurel

## Wikipédia :

Une interface est la couche limite par laquelle ont lieu les échanges et les interactions entre deux éléments.

Exemple : Interface Homme Machine

Dans la plupart des langages une interface est un **contrat**.

En JavaScript : pas d'interface.



Une formation  
**Alphorm**.com

## Démonstrations

Les interfaces et le typage structurel  
Attributs optionnels  
Attributs en lecture seule  
Attribut de type indexable



# Merci



Une formation  
**Alphorm**.com

## Les types génériques sous TypeScript



Frédéric GAURAT



Une formation  
**Alphorm**.com

## Plan

Introduction aux **Generics**

Les fonctions génériques

Les classes génériques

Ajouter des contraintes sur les types



Une formation  
**Alphorm**.com

## Introduction aux Generics

L'ajout de type à JavaScript offre une sécurité aux développeurs

**Mais** limite le passage des paramètres aux seuls types déclarés

**La solution** est de déclarer une autre variable qui sera initialisée avec le type



Une formation  
**Alphorm**.com

## Démonstrations

Les fonctions génériques

Les classes génériques

Ajouter des contraintes sur les types



# Merci



Une formation  
**Alphorm**.com

# Les Enums sous TypeScript



Frédéric GAURAT



Une formation  
**Alphorm**.com

## Plan

Notion d'énumération  
Syntaxe et utilisation



Une formation  
**Alphorm**.com

## Notion d'Enum

- Définir un ensemble de valeurs numériques constantes
- Utiliser le mot-clé **enum**
- Bidirectionnelles :
  - nom => valeur
  - valeur => nom



Une formation  
**Alphorm**.com

## Démonstrations

Syntaxe et utilisation



# Merci



Une formation  
**Alphorm**  
com

## Les Symbols sous TypeScript



Frédéric GAURAT



Une formation  
**Alphorm**.com

## Plan

Présentation des Symbols

Un exemple simple d'utilisation



Une formation  
**Alphorm**.com

## Présentation des Symbols

Un Symbol est un type de données  
apparu en ES6

Type primitif avec des caractéristiques  
particulières :

- Initialisé à partir d'un constructeur,
- Unique et non modifiable,
- Utilisable comme attribut d'objet.



# Merci



Une formation  
**Alphorm**  
com

## Les Itérateurs et générateurs sous TypeScript



Frédéric GAURAT



Une formation  
**Alphorm**.com

## Plan

Un objet itérable  
Implémentation d'un itérateur  
Les générateurs



Une formation  
**Alphorm**.com

## Un objet itérable

Supporte les boucles **for..of** ou **for..in**  
Implémente la propriété  
**Symbol.iterator**

# Merci



Une formation  
**Alphorm**  
com

## Les Modules sous TypeScript



Frédéric GAURAT



Une formation  
**Alphorm**.com

# Plan

Notion de module  
Démonstrations



Une formation  
**Alphorm**.com

## Notion de module

Les modules permettent d'organiser du code plus facilement

Les modules sont stockés dans des fichiers distincts (un module par fichier)

Les relations entre modules sont spécifiées en terme d'import / export

Le chargement des modules dépend du contexte d'exécution du script (navigateur ou node.js)



Une formation  
**Alphorm**.com

## Démonstrations

Déclarer un module et l'exporter  
Importer un module ou une partie  
Les chargeurs de module  
Configuration de TypeScript



# Merci



Une formation  
**Alphorm**.com

# Les Namespaces sous TypeScript



Frédéric GAURAT



Une formation  
**Alphorm**.com

## Plan

Présentation des namespaces  
Démonstrations



Une formation  
**Alphorm**.com

## Présentation des namespaces

Organiser le code

Eviter les collisions de nom

Limiter la portée des éléments  
déclarés



Une formation  
**Alphorm**.com

## Démonstrations

Déclaration d'un namespace

Namespace sur plusieurs fichiers

Directive triple-slash

Namespaces et modules



# Merci



Une formation  
**Alphorm**.com

## Les décorateurs



Frédéric GAURAT





Une formation  
**Alphorm**.com

# Plan

Présentation des décorateurs  
Démonstrations



Une formation  
**Alphorm**.com

## Présentation des décorateurs

Des déclarations particulières appliquées  
à des : *classes, méthodes, accesseurs, propriétés,*  
*paramètres.*

Préfixés d'un @

Exécutés au moment du runtime



Une formation  
**Alphorm**.com

# Implémenter un décorateur

## Une fonction :

Appel immédiat pas d'initialisation possible

## Une factory :

Appel immédiat mais permet de paramétrer le décorateur



Une formation  
**Alphorm**.com

# Démonstrations

Implémenter un décorateur



# Merci



Une formation  
**Alphorm**  
com

## Les mixins



Frédéric GAURAT



Une formation  
**Alphorm**.com

## Plan

Notion de mixins

Implémenter un mixin en TypeScript



Une formation  
**Alphorm**.com

## Notion de mixins

**mixer** plusieurs classes dans une seule

**L'héritage multiple** (non supporté, TypeScript)

# Merci



Une formation  
**Alphorm**  
com

## Les fichiers de déclaration de type



Frédéric GAURAT



Une formation  
**Alphorm**.com

# Plan

Présentation des fichiers  
Démonstrations



Une formation  
**Alphorm**.com

## Présentation des fichiers

Utiliser des bibliothèques externes en bénéficiant de la vérification de type proposé par TypeScript

TypeScript permet la prise en compte des différentes « formes » de bibliothèque (Web, Node.js)



Une formation  
**Alphorm**.com

## Démonstrations

Créer un fichier de déclaration de type  
Utilisation d'un fichier de déclaration de type  
Utiliser TypeScript avec d'autres Bibliothèques



# Merci



Une formation  
**Alphorm**.com

# Le projet TodoList



Frédéric GAURAT



Une formation  
**Alphorm**.com

## Plan

Le projet Todo List  
Installation des outils  
Développement du projet





Une formation  
**Alphorm**.com

## Le projet Todo List

Todo	dueDate	complete	Action
Todo 1	01/012017	<input checked="" type="checkbox"/>	<a href="#">Delete</a> / <a href="#">Update</a>
Todo 2	01/012017	<input type="checkbox"/>	<a href="#">Delete</a> / <a href="#">Update</a>
Todo 3	01/012017	<input checked="" type="checkbox"/>	<a href="#">Delete</a> / <a href="#">Update</a>
Todo 4	01/012017	<input type="checkbox"/>	<a href="#">Delete</a> / <a href="#">Update</a>



Une formation  
**Alphorm**.com

## Installation des outils

Node/npm

MongoDB / Deployd



Une formation  
**Alphorm**.com

# Démonstration

Préparation du projet  
Installation des dépendances

# Merci



Une formation  
**Alphorm**.com

## Conclusion



Frédéric GAURAT



Une formation  
**Alphorm**.com

## Plan

Ce qui a été couvert  
Ce qui reste à aborder



Une formation  
**Alphorm**.com

## Ce qui a été couvert

- Présentation de la formation
- Présentation du projet de la formation
- TypeScript présentation installation et utilisation
- Les éléments du Langage TypeScript
- L'organisation du code TypeScript
- Les particularités



Une formation  
**Alphorm**.com

## Ce qui reste à aborder





**Merci**