

# TypeScript

Michael X ☐ NATIS





Compétence visée : Comprendre l'articulation de TypeScript autour du JavaScript



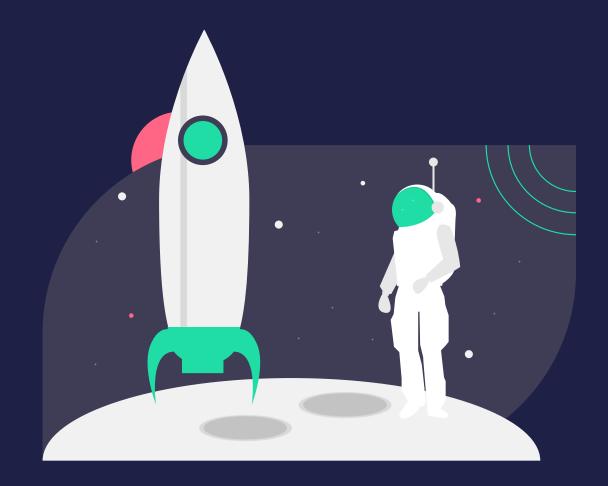
- 1. Utilité et avantages
- 2. Inconvénients

#### Transpilation

- 1. Fonctionnement
- 2. Configuration avec tsconfig.json
- 3. Les source maps
- 4. Pratique (hello world en typescript)

#### Les principes de la POO

- 1. Abstraction et interface
- 2. L'héritage
- 3. Le polymorphisme
- 4. L'encapsulation
- 5. Pratique (développer une API)







- 1. Utilité et avantages
- 2. Inconvénients

#### Transpilation

- Fonctionnement
- 2. Configuration avec tsconfig.json
- 3. Les source maps
- Pratique (hello world en typescript)

#### Les principes de la POC

- 1. Abstraction et interface
- 2. L'héritage
- 3. Le polymorphisme
- 4. L'encapsulation
- 5. Pratique (développer une API)

En termes simples, TypeScript est un langage de programmation open source développé par Microsoft qui compile en JavaScript.

Depuis sa sortie en 2012, le langage est resté en développement actif et continue de gagner en popularité chaque année.





- 1. Utilité et avantages
- 2. Inconvénients

#### Transpilation

- 1. Fonctionnement
- 2. Configuration avec tsconfig.json
- 3. Les source maps
- Pratique (hello world en typescript)

#### Les principes de la POC

- 1. Abstraction et interface
- 2. L'héritage
- 3. Le polymorphisme
- L'encapsulatior
- 5. Pratique (développer une API)

La prise en charge du typage fort ne fait pas partie de la norme ECMAScript et ne sera probablement jamais due à la nature interprétée au lieu de la nature compilée de JavaScript.





- 1. Utilité et avantages
- 2. Inconvénients

#### Transpilation

- Fonctionnement
- 2. Configuration avec tsconfig.json
- Les source maps
- Pratique (hello world en typescript)

#### Les principes de la POC

- 1. Abstraction et interface
- 2. L'héritage
- 3. Le polymorphisme
- L'encapsulatior
- 5. Pratique (développer une API)

L'augmentation clé que TypeScript apporte à JavaScript est un système de types (d'où le nom "TypeScript").





- 1. Utilité et avantages
- 2. Inconvénients

#### Transpilation

- 1. Fonctionnement
- 2. Configuration avec tsconfig.json
- 3. Les source maps
- Pratique (hello world en typescript)

#### Les principes de la POC

- 1. Abstraction et interface
- 2. L'héritage
- 3. Le polymorphisme
- L'encapsulatior
- 5. Pratique (développer une API)

Le système de types de TypeScript est incroyablement riche et comprend : des interfaces, des énumérations, des types hybrides, des génériques, des types d'union/intersection, des modificateurs d'accès et bien plus encore.





- 1. Utilité et avantages
- 2. Inconvénients

#### Transpilation

- 1. Fonctionnement
- 2. Configuration avec tsconfig.json
- 3. Les source maps
- Pratique (hello world en typescript)

#### Les principes de la POC

- 1. Abstraction et interface
- 2. L'héritage
- 3. Le polymorphisme
- L'encapsulation
- 5. Pratique (développer une API)

Le site officiel de TypeScript donne un aperçu de ces fonctionnalités. Le système de typage de Typescript est comparable à la plupart des autres langages typés comme le C# par exemple.





- 1. Utilité et avantages
- 2. Inconvénients

#### Transpilation

- 1. Fonctionnement
- 2. Configuration avec tsconfig.json
- Les source maps
- Pratique (hello world en typescript)

#### Les principes de la POC

- 1. Abstraction et interface
- 2. L'héritage
- 3. Le polymorphisme
- L'encapsulation
- 5. Pratique (développer une API)

TypeScript vous permet de présenter plus rapidement de nouveaux développeurs à votre projet.





- 1. Utilité et avantages
- 2. Inconvénients

#### Transpilation

- 1. Fonctionnement
- 2. Configuration avec tsconfig.json
- 3. Les source maps
- Pratique (hello world en typescript)

#### Les principes de la POC

- 1. Abstraction et interface
- 2. L'héritage
- 3. Le polymorphisme
- L'encapsulatior
- 5. Pratique (développer une API)

Les structures de données et les annotations de type explicitement définies facilitent la compréhension des décisions prises par les ingénieurs qui ont écrit à l'origine votre code.





- 1. Utilité et avantages
- 2. Inconvénients

#### Transpilation

- Fonctionnement
- 2. Configuration avec tsconfig.json
- 3. Les source maps
- Pratique (hello world en typescript)

#### Les principes de la POC

- 1. Abstraction et interface
- 2. L'héritage
- 3. Le polymorphisme
- L'encapsulatior
- 5. Pratique (développer une API)

La configuration nécessaire peut être assez compliquée à comprendre ou a démystifier!





- Utilité et avantages
- 2. Inconvénients

#### Transpilation

- Fonctionnement
- 2. Configuration avec tsconfig.json
- Les source maps
- Pratique (hello world er typescript)

#### Les principes de la POC

- 1. Abstraction et interface
- 2. L'héritage
- 3. Le polymorphisme
- L'encapsulatior
- 5. Pratique (développer une API)

Cependant, ce n'est pas strictement un inconvénient de TypeScript, mais plutôt un inconvénient qui découle de son interopérabilité totale avec JavaScript, ce qui lui-même laisse encore plus de place aux complications.





- 1. Utilité et avantages
- 2. Inconvénients

#### Transpilation

- Fonctionnement
- Configuration avec tsconfig.json
- Les source maps
- Pratique (hello world en typescript)

#### Les principes de la POC

- 1. Abstraction et interface
- 2. L'héritage
- 3. Le polymorphisme
- L'encapsulatior
- 5. Pratique (développer une API)

Un autre argument contre TypeScript est qu'il nécessite une compilation, contrairement à JavaScript.





- 1. Utilité et avantages
- 2. Inconvénients

#### Transpilation

- Fonctionnement
- 2. Configuration avec tsconfig.json
- Les source maps
- Pratique (hello world en typescript)

#### Les principes de la POC

- 1. Abstraction et interface
- 2. L'héritage
- 3. Le polymorphisme
- 4. L'encapsulation
- 5. Pratique (développer une API)

Pour utiliser TypeScript, vous avez besoin d'un processus de construction pour compiler en code JavaScript.

Le processus de construction dépend de la taille de votre projet.





- 1. Utilité et avantages
- 2. Inconvénients

#### Transpilation

- 1. Fonctionnement
- 2. Configuration avec tsconfig.json
- 3. Les source maps
- Pratique (hello world en typescript)

#### Les principes de la POC

- 1. Abstraction et interface
- 2. L'héritage
- 3. Le polymorphisme
- 4. L'encapsulation
- 5. Pratique (développer une API)

Les transpilers sont également connus sous le nom de compilateurs source-àsource. Donc, essentiellement, ils sont un sous-ensemble de compilateurs qui prennent un fichier de code source et le convertissent en un autre fichier de code source dans une autre langue ou une version différente de la même langue.





- Utilité et avantages
- 2. Inconvénients

#### Transpilation

- 1. Fonctionnement
- Configuration avec tsconfig.json
- Les source maps
- Pratique (hello world en typescript)

#### Les principes de la POC

- 1. Abstraction et interface
- 2. L'héritage
- 3. Le polymorphisme
- L'encapsulatior
- 5. Pratique (développer une API)

La sortie est généralement compréhensible par un humain. Cette sortie doit encore passer par un compilateur ou un interpréteur pour pouvoir s'exécuter sur la machine.





- Utilité et avantages
- 2. Inconvénients

#### Transpilation

- 1. Fonctionnement
- 2. Configuration avec tsconfig.json
- 3. Les source maps
- Pratique (hello world en typescript)

#### Les principes de la POC

- 1. Abstraction et interface
- 2. L'héritage
- 3. Le polymorphisme
- L'encapsulatior
- 5. Pratique (développer une API)

Le compilateur TypeScript prend en charge la compilation incrémentielle (option -- watch du compilateur) afin que toutes les modifications ultérieures puissent être compilées plus rapidement.





## La configuration du transpiler est consignée dans un fichier tsconfig.json

#### Utilité, avantages et inconvénients

- Utilité et avantages
- 2. Inconvénients

#### Transpilation

- Fonctionnement
- 2. Configuration avec tsconfig.json
- 3. Les source maps
- Pratique (hello world en typescript)

#### Les principes de la POC

- 1. Abstraction et interface
- L'héritage
- 3. Le polymorphisme
- L'encapsulatior
- 5. Pratique (développer une API)

```
projet > typescript > s tsconfig.json > {} compilerOptions
         "compilerOptions": {
           /* Basic Options */
           // "incremental": true,
           "target": "es6",
           "module": "es2015",
           "lib": ["ES2015", "DOM"],
           // "allowJs": true,
           // "checkJs": true,
           // "jsx": "preserve",
 11
           // "declaration": true,
           // "declarationMap": true,
           // "sourceMap": true,
           // "outFile": "./",
           "outDir": "./dist",
           // "rootDir": "./",
           // "composite": true,
           // "tsBuildInfoFile": "./",
           // "removeComments": true,
           // "noEmit": true,
           // "importHelpers": true,
 21
           // "downlevelIteration": true,
           // "isolatedModules": true,
```





- 1. Utilité et avantages
- 2. Inconvénients

#### Transpilation

- Fonctionnement
- 2. Configuration avec tsconfig.json
- Les source maps
- Pratique (hello world er typescript)

#### Les principes de la POC

- 1. Abstraction et interface
- 2. L'héritage
- 3. Le polymorphisme
- L'encapsulatior
- 5. Pratique (développer une API)

Chaque ligne de la configuration permet d'activer ou de désactiver un comportement du transpiler.





- Utilité et avantages
- 2. Inconvénients

#### Transpilation

- Fonctionnement
- 2. Configuration avec tsconfig.json
- 3. Les source maps
- Pratique (hello world en typescript)

#### Les principes de la POC

- 1. Abstraction et interface
- 2. L'héritage
- 3. Le polymorphisme
- L'encapsulatior
- 5. Pratique (développer une API)

Le compilateur TypeScript peut intégrer les informations de carte source dans les fichiers .js générés ou créer des fichiers .map distincts.





- 1. Utilité et avantages
- 2. Inconvénients

#### Transpilation

- 1. Fonctionnement
- 2. Configuration avec tsconfig.json
- 3. Les source maps
- Pratique (hello world en typescript)

#### Les principes de la POC

- 1. Abstraction et interface
- 2. L'héritage
- 3. Le polymorphisme
- L'encapsulation
- 5. Pratique (développer une API)

Les informations de carte source peuvent être utilisées par des utilitaires de débogage tels que Chrome DevTools et d'autres IDE pour relier les lignes du JavaScript à celles qui les ont générées dans le TypeScript.





- Utilité et avantages
- 2. Inconvénients

#### Transpilation

- Fonctionnement
- Configuration avec tsconfig.json
- 3. Les source maps
- 4. Pratique (hello world en typescript)

#### Les principes de la POC

- 1. Abstraction et interface
- 2. L'héritage
- 3. Le polymorphisme
- L'encapsulation
- 5. Pratique (développer une API)

# PRATIQUE







- Utilité et avantages
- 2. Inconvénients

#### Transpilation

- 1. Fonctionnement
- 2. Configuration avec tsconfig.json
- 3. Les source maps
- Pratique (hello world en typescript)

#### Les principes de la POO

- 1. Abstraction et interface
- 2. L'héritage
- Le polymorphisme
- L'encapsulatior
- 5. Pratique (développer une API)

### Et si on jouait un jeu?







« sinon ... on s'est fiche ... »



# Touche de clavier d'ordinateur



# Une batterie d'ordinateur



# Une souris d'ordinateur



## Un vidéo-projecteur



# Des hauts-parleurs



# Un microphone



# Une pédale de frein

- 1. Utilité et avantages
- 2. Inconvénients

#### Transpilation

- Fonctionnement
- Configuration avec tsconfig.json
- Les source maps
- Pratique (hello world en typescript)

#### Les principes de la POC

- 1. Abstraction et interface
- 2. L'héritage
- 3. Le polymorphisme
- L'encapsulatior
- 5. Pratique (développer une API)

#### Abstraction

La capacité d'utiliser des composants sans connaître leurs détails techniques





- 1. Utilité et avantages
- 2. Inconvénients

#### Transpilation

- Fonctionnement
- Configuration avec tsconfig.json
- Les source maps
- Pratique (hello world en typescript)

#### Les principes de la POO

- 1. Abstraction et interface
- 2. L'héritage
- 3. Le polymorphisme
- L'encapsulation
- 5. Pratique (développer une API)

Une interface permet de lister les fonctionnalités attendues d'un composant

On dit qu'une classe implémente une interface

Cette relation entre la classe et l'interface est implémentée grâce au mot-clé inplements





- Utilité et avantages
- 2. Inconvénients

#### Transpilation

- Fonctionnement
- 2. Configuration avec tsconfig.json
- Les source maps
- Pratique (hello world en typescript)

#### Les principes de la POC

- 1. Abstraction et interface
- 2. L'héritage
- 3. Le polymorphisme
- L'encapsulatior
- 5. Pratique (développer une API)

L'héritage permet une factorisation du code avec une réflexion de classification.

Exemple : 1 chose est une chose, mais 1 chose peut-elle être plusieurs choses ?







#### **CHIEN**

- + age
- + groupe sanguin
- + taille
- + poids
- + mange()
- + dort()
- + aboie()

#### CHAT

- Coussinets
- + age
- + groupe sanguin
- + taille
- + poids
- + mange()
- + dort()
- + miauler()



# ???? + age - groupe sanguin # taille + poids + mange() + dort()

CHIEN
+ aboie()

- coussinets
+ miauler()



#### ANIMAL

- + age
- + groupe sanguin
- + taille
- + poids
- + mange()
- + dort()

#### **CHIEN**

+ aboie()

#### **CHAT**

- coussinets
- + miauler()

- Utilité et avantages
- 2. Inconvénients

#### Transpilation

- Fonctionnement
- 2. Configuration avec tsconfig.json
- 3. Les source maps
- Pratique (hello world en typescript)

#### Les principes de la POC

- 1. Abstraction et interface
- 2. L'héritage
- 3. Le polymorphisme
- L'encapsulatior
- 5. Pratique (développer une API)

La relation d'héritage entre 2 classes est implémenté par le mot-clé extends





- 1. Utilité et avantages
- 2. Inconvénients

#### Transpilation

- Fonctionnement
- 2. Configuration avec tsconfig.json
- Les source maps
- Pratique (hello world en typescript)

#### Les principes de la POC

- 1. Abstraction et interface
- 2. L'héritage
- 3. Le polymorphisme
- 4. L'encapsulation
- 5. Pratique (développer une API)

Le polymorphisme est l'utilisation d'un seul et même symbole pour représenter plusieurs structures différentes.

Le concept est emprunté à un principe en biologie où un organisme ou une espèce peut avoir de nombreuses formes ou stades différents







#### ANIMAL

- + age
- + groupe sanguin
- + taille
- + poids
- + mange()
- + dort()

#### **CHIEN**

- + aboie()
- + mange()

#### **CHAT**

- coussinets
- + miauler()
- + mange()

- Utilité et avantages
- 2. Inconvénients

#### Transpilation

- 1. Fonctionnement
- Configuration avec tsconfig.json
- Les source maps
- Pratique (hello world en typescript)

#### Les principes de la POC

- 1. Abstraction et interface
- 2. L'héritage
- 3. Le polymorphisme
- 4. L'encapsulation
- 5. Pratique (développer une API)

L'encapsulation est le concept de restreindre l'accès direct à certains des composants d'un objet







#### ANIMAL

- + age
- + groupe sanguin
- + taille
- + poids
- sang
- + mange()
- # dort()

#### **CHIEN**

- + aboie()
- + mange()

#### **CHAT**

- coussinets
- + miauler()
- + mange()

- 1. Utilité et avantages
- 2. Inconvénients

#### Transpilation

- Fonctionnement
- 2. Configuration avec tsconfig.json
- 3. Les source maps
- Pratique (hello world er typescript)

#### Les principes de la POC

- 1. Abstraction et interface
- L'héritage
- 3. Le polymorphisme
- L'encapsulatior
- 5. Pratique (développer une API)

# PRATIQUE









## Pour aller plus loin

https://www.typescriptlang.org/