**Mémo TypeScript**

Le langage typescript qui est un langage de programmation libre et open source développé par Microsoft qui a pour but d'améliorer et de sécuriser la production de code JavaScript.

Il s'agit d'un sur-ensemble syntaxique de javascript, c’est-à-dire qu’il est destiné à enrichir le

langage javascript, et nécessite d’être “compilé” en javascript pour être interprété par les

navigateurs internet classiques.

Installez NodeJS (choisissez la LTS) : <https://nodejs.org/en/download>

vérifier dans un terminal :

node --version

npm --version

Installez typescript globalement sur votre machine (terminal en mode administrateur) :

**install -g typescript**

Créer le manifeste (sorte de manuel d’un projet node)

**npm init --Yes** ( l’option --Yes permet de répondre automatique aux questins)

package.json :

*{*

*"name": "decouverte-typescript",*

*"version": "1.0.0",*

*"description": "1er exercice typescript",*

*"main": "index.js",*

*"scripts": {*

*"test": "echo \"Error: no test specified\" && exit 1"*

*},*

*"keywords": [],*

*"author": "stephane chevrier",*

*"license": "ISC",*

*"dependencies": {*

*"express": "^4.18.2"*

*}*

*}*

Fichier **tsconfig.json** qui gère la configuration de la compilation typescript du projet.

{

"compilerOptions": {

/\* Visit https://aka.ms/tsconfig to read more about this file \*/

"target": "es2016",

"module": "commonjs", /\* Specify what module code is generated. \*/

"sourceMap": true, /\* Create source map files for emitted JavaScript files. \*/

"outDir": "dist", /\* Specify an output folder for all emitted files. \*/

"noEmitOnError": true, /\* Disable emitting files if any type checking errors are reported. \*/

"esModuleInterop": true, /\* Emit additional JavaScript to ease support for importing \*/

"forceConsistentCasingInFileNames": true, /\* Ensure that casing is correct in imports. \*/

"strictNullChecks": true, /\* When type checking, take into account 'null' and 'undefined'. \*/

}

}

Dans Intellij, **cliquer sur TypeScript** (en bas à droite) **pour compiler un fichier .ts**

**Syntaxe TypeScript**

**L’intérêt de TypeScript est d’éviter le coté très permissif de JavaScript**

Exemples variables :

**protected** owner : **string**;

protected articles : Articles[]; // List d’Articles, Articles est une interface

***interface*** *Articles {*

*id :* ***number****;*

*quantity :* ***number****;*

*name :* ***string****;*

*price :* ***number****;*

*}*

**public** addArticle(newArticle**:Articles**) **: void** {

}

la variable existingArticle est soit Articles soit undefined

**cons**t existingArticle : **Articles | undefined** = this.articles.find(

(article **: Articles**) **: boolean** => article.id === articleId);

Exemples modificateurs accès :

override pour indiquer clairement qu’il s’agit d’une surcharge d’une méthode

**override** addArticle(article:Articles) : void