

# TypeScript

## Installation dans VS Code

- Prérequis : installer Node.js

| <https://nodejs.org/en/>

## Créer un projet

- Créer un workspace par exemple : workspace-ts
- Dans le workspace créer un projet par exemple : my-dog
- Se positionner dans le projet en utilisant le terminal de VS Code :  

```
$ cd /chemin/workspace-ts/my-dog  
$
```
- Dans le projet my-dog créer un répertoire src qui contiendra les sources
- et un répertoire dist qui contiendra le projet compilé

# Première application

- Dans le répertoire src créer un fichier app.ts contenant le code suivant :

## app.ts

```
interface Dog {
  name: string;
  eating: boolean;
  sleeping: boolean;
}

class Dogs implements Dog {
  name: string;
  eating: boolean;
  sleeping: boolean;

  constructor(dog: Dog) {
    this.name = dog.name;
    this.eating = dog.eating;
    this.sleeping = dog.sleeping;
  }

  onCanEat() {
    if (this.eating) {
      console.log(`Mon chien ${this.name} mange`);
    } else {
      console.log(`Mon chien ${this.name} ne mange pas`);
    }
  }
}

const myDog = {
  name: "Brutus",
  eating: true,
  sleeping: false
};

const dogs = new Dogs(myDog);
dogs.onCanEat();
```

# Compilateur TypeScript

- Créez un fichier tsconfig.json à la racine du dossier du projet et ajoutez le contenu suivant :

tsconfig.json est le fichier de configuration du transpileur TS -> JS (TypeScript en JavaScript)

## tsconfig.json

```
{
  "compilerOptions": {
    "outDir": "./dist",
    "rootDir": "./src",
    "sourceMap": true,
    "moduleResolution": "node",
    "target": "es5"
  }
}
```

# Dépendances

Nous avons également besoin du fichier package.json.

package.json est le fichier de gestion des dépendances

- Exécutez simplement les commandes suivantes dans votre terminal pour le créer et pour ajouter les dépendances requises :

```
$ npm init --yes
```

```
$ npm install typescript --save-dev
```

- Le fichier package.json doit ressembler à :

## package.json

```
{
  "name": "my-dog",
  "version": "1.0.0",
  "description": "",
  "main": "app.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1",
    "start": "node dist/app.js",
    "prestart": "npm run build",
    "build": "tsc"
  }
}
```

# ESLint

- Installer le package ESLint

Comme son nom l'indique, ESLint est un linter. Il permet de passer au peigne fin les erreurs commises lors de la rédaction du code en se basant sur un certain nombre de règles. Contrairement à Prettier, ESLint ne s'occupe que de la qualité du code et non du formatage des règles d'indentations.

```
$ npm install eslint
```

## Déboguer

- Créer un répertoire à la racine du projet se nommant ".vscode"
- Dans le répertoire .vscode créer un fichier se nommant "launch.json"

Le fichier launch.json contient la configuration du déboguer de VS Code

### launch.json

```
{
  "version": "0.2.0",
  "configurations": [
    {
      "type": "node",
      "request": "launch",
      "name": "Build Project",
      "program": "${workspaceFolder}/src/app.ts",
      "preLaunchTask": "npm: build",
      "sourceMaps": true,
      "smartStep": true,
      "internalConsoleOptions": "openOnSessionStart",
      "outFiles": ["${workspaceFolder}/dist/**/*.js"]
    }
  ]
}
```