



# Démarrer un projet TypeScript

➤ Thèmes	JavaScript TypeScript
≡ Description	Créer et démarrer son premier projet Typescript.
🕒 Date de création	@23 juin 2025 09:18
👤 Créée par	Kévin Wolff

## Chapitres

- 📖 [Démarrer un projet TypeScript simple](#)
- 📄 [Initialiser NPM et installer TypeScript](#)
- 📄 [Configuration de TypeScript](#)
- 📄 [Tester votre installation](#)

## Sur le même sujet

- 📖 [Installer Node.js et NVM](#)
- 📖 [Introduction à TypeScript](#)

## Démarrer un projet TypeScript simple

Le navigateur ne sait pas interpréter le code TypeScript. Il est donc nécessaire de *compiler* le code TypeScript en JavaScript afin qu'il soit compréhensible par le navigateur.

Cette opération est simple, à condition que votre configuration TypeScript soit correctement mise en place. Il suffira d'indiquer à TypeScript quels dossiers il doit compiler, ainsi que l'emplacement dans lequel il devra placer les fichiers JavaScript générés.

Veillez à bien organiser votre projet en séparant clairement le code source TypeScript du code compilé JavaScript.


Commencez par créer un dossier `<nom_de_votre_projet>` structuré comme suit :

```
├── dist
├── index.html
├── src
│   └── main.ts
└── style.css
```

Vous remarquerez que l'extension du fichier `main.ts` est bien `.ts`, ce qui signifie qu'il s'agit d'un fichier TypeScript. Vous pouvez donc y écrire librement du code TypeScript sans que votre IDE ne signale d'erreur.

Par ailleurs, vous êtes libre d'organiser le dossier `src` comme vous le souhaitez. L'unique contrainte est d'y placer tous vos fichiers `.ts`, car c'est dans ce dossier que TypeScript ira chercher les fichiers à compiler. Si des fichiers TypeScript se trouvent en dehors de ce dossier et que votre configuration n'est pas adaptée, ils ne seront tout simplement pas compilés.

## Initialiser NPM et installer TypeScript

Il est nécessaire d'installer la bibliothèque TypeScript dans votre projet afin de pouvoir l'utiliser. Pour cela, vous passerez par NPM (*Node Package Manager*), le gestionnaire de paquets fourni avec Node.js. Voir document  [Node et NPM](#) pour plus d'informations.

Pour commencer, initialiser NPM dans votre projet avec la commande suivante. Répondez aux questions en appuyant sur **Entrée** cela n'a pas d'importance.

```
npm init
```

Voici la commande à exécuter pour installer la librairie dans votre projet :

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

Le drapeau **--save-dev** indique que vous souhaitez installer la dépendance uniquement pour l'environnement de développement. Autrement dit, cette dépendance ne sera ni installée ni accessible lorsque le projet sera exécuté dans un autre environnement, comme la production.

Par exemple, il ne sera pas nécessaire d'inclure la dépendance TypeScript en production. En effet, un projet est déjà *buildé* et compilé lorsqu'il est déployé en production, et n'a donc plus besoin de TypeScript à ce stade.

## Configuration de TypeScript

Il est nécessaire de créer un fichier de configuration pour que TypeScript puisse fonctionner correctement dans votre projet.

Pour cela, créez un fichier nommé **tsconfig.json** à la racine du projet, puis ajoutez-y le contenu suivant :

```
{
  "compilerOptions": {
    "target": "ES6",      # dernière version de JavaScript
    "module": "ES6",      # nécessaire aux imports/exports
    "outDir": "./dist",   # fichier dans lequel ranger les fichiers compilés
    "rootDir": "./src",   # dossier racine du projet, ici `src`
    "strict": true
  },
  "include": ["src"],     # dossier(s) parcouru par TS pour la compilation
  "exclude": ["node_modules"] # dossier(s) à exclure de la compilation de TS
}
```

## Tester votre installation

Si vous avez suivis le document, vous devriez avoir un projet avec l'architecture suivante :

```
|— dist
|— node_modules
|— index.html
```

```
├── package.json
├── package-lock.json
├── src
│   └── main.ts
├── style.css
└── tsconfig.json
```

Ajouter le code TypeScript suivant dans votre fichier `src/main.ts` :

```
const test: string = "Hello World";
```

Pour compiler votre code, exécutez la commande suivante : `npx tsc` . Cette commande lancera le compilateur TypeScript, qui analysera les fichiers `.ts` présents dans le dossier `src` et les convertira en fichiers JavaScript.

Le résultat de la compilation sera généré dans le dossier `dist` . Par exemple, le fichier `src/main.ts` sera transformé en `dist/main.js` . Si vous ouvrez ce fichier, vous y retrouverez votre code, mais en version JavaScript.

Félicitations, votre configuration est fonctionnelle ! Vous êtes prêt à travailler sur ce nouveau projet TypeScript.

Pour aller plus loin, vous pouvez vous faciliter la vie en utilisant le *watcher* de TypeScript. Ce mode permet de surveiller les modifications de vos fichiers `.ts` et de les compiler automatiquement en tâche de fond.

Plus besoin de relancer manuellement la commande à chaque changement voici la commande à utiliser :

```
npx tsc --watch
```