Structure de base TypeScript console

1. Initialisation du projet

```
mkdir mon-projet-console
cd mon-projet-console
npm init -y
npm install typescript @types/node ts-node --save-dev
npx tsc --init
```

2. Structure des fichiers

```
mon-projet-console/
├── src/
│   └── index.ts
├── dist/              (généré à la compilation)
├── package.json
├── tsconfig.json
└── node_modules/
```

3. Configuration tsconfig.json

```
{
  "compilerOptions": {
    "target": "ES2020",
    "module": "CommonJS",
    "outDir": "./dist",
    "rootDir": "./src",
    "strict": true,
    "esModuleInterop": true,
    "skipLibCheck": true,
    "forceConsistentCasingInFileNames": true,
    "resolveJsonModule": true
  },
  "include": ["src/**/*"],
  "exclude": ["node_modules", "dist"]
}
```

4. Fichier principal src/index.ts

```
// Point d'entrée du programme
function main(): void {
  console.log("Hello, World depuis TypeScript !");

  // Récupérer les arguments de la ligne de commande
  const args: string[] = process.argv.slice(2);
```

```
  if (args.length > 0) {
    console.log("Arguments reçus :", args);
  }

  // Exemple d'interaction basique
  const nom: string = "Utilisateur";
  saluer(nom);
}

function saluer(nom: string): void {
  console.log(`Bonjour, ${nom} !`);
}

// Lancement du programme
main();
```

5. Scripts dans package.json

```
{
  "name": "mon-projet-console",
  "version": "1.0.0",
  "scripts": {
    "build": "tsc",
    "start": "node dist/index.js",
    "dev": "ts-node src/index.ts",
    "watch": "tsc --watch"
  },
  "devDependencies": {
    "typescript": "^5.0.0",
    "@types/node": "^20.0.0",
    "ts-node": "^10.9.0"
  }
}
```

6. Commandes utiles

```
# Exécution directe en développement (sans compiler)
npm run dev

# Compiler puis exécuter
npm run build
npm start

# Compilation automatique en continu
npm run watch
```

7. Bonus : lecture d'entrée utilisateur (interactif)

Pour un programme interactif type CLI, utilisez le module `readline` :

```
import * as readline from 'readline';

const rl = readline.createInterface({
  input: process.stdin,
  output: process.stdout
});

rl.question("Quel est votre nom ? ", (reponse: string) => {
  console.log(`Bonjour, ${reponse} !`);
  rl.close();
});
```

Astuce

Pour des CLI plus avancées (avec options, sous-commandes, etc.), pensez à des librairies comme :

- **commander** – gestion d'arguments et sous-commandes
- **inquirer** – prompts interactifs élégants
- **chalk** – colorer la sortie console

```
npm install commander inquirer chalk
```

From:
http://slamwiki2.kobject.net/ - **SlamWiki 2.1**

Permanent link:
http://slamwiki2.kobject.net/cnam/utc503/ts

Last update: **2026/08/24 14:25**