Convert JavaScript to TypeScript

Step 1: Set Up Your JavaScript Project

- Create a sample JavaScript project:

```
mkdir sample-js-project
cd sample-js-project
npm init -y
```

- Create some JavaScript files:
  Create an index.js file with the following content:

```
// index.js
const add = (a, b) => a + b;

console.log(add(2, 3));
```

- Install necessary npm packages:

```
npm install
```

Step 2: Install TypeScript

- Install TypeScript and its dependencies:

```
npm install typescript --save-dev
npm install @types/node --save-dev
```

- Initialize a TypeScript configuration file:

```
npx tsc --init
```

Step 3: Update Configuration Files

- Update tsconfig.json:
  Modify tsconfig.json to include the following settings:

```
{
  "compilerOptions": {
    "target": "ES6",
    "module": "commonjs",
    "strict": true,
    "esModuleInterop": true,
    "skipLibCheck": true,
    "forceConsistentCasingInFileNames": true,
    "outDir": "./dist"
  },
  "include": ["src"],
  "exclude": ["node_modules"]
}
```

Step 4: Rename JavaScript Files to TypeScript

- Rename your JavaScript files:

```
mkdir src
mv index.js src/index.ts
```

Step 5: Add Type Annotations

- Modify src/index.ts to include type annotations:

```
// src/index.ts
const add = (a: number, b: number): number => a + b;

console.log(add(2, 3));
```

Step 6: Handle Existing Code Patterns

- Address common issues:
  If you have more complex code, you may encounter issues such as:

  - ❖ Implicit any types.
  - ❖ Incorrect module imports/exports.
  - ❖ Missing type definitions.

- Example of converting an implicit any type:

```
// src/index.ts
const greet = (name: string): string => `Hello, ${name}`;

console.log(greet("World"));
```

Step 7: Compile and Run the Project

- Compile TypeScript to JavaScript:

```
npx tsc
```

- Run the compiled JavaScript:

```
node dist/index.js
```

Step 8: Address Any Remaining Issues

- Handle errors and warnings:
  TypeScript's strict mode will help catch potential issues. Fix them as they appear.

- Install type definitions for third-party libraries:
  For any third-party libraries, you might need to install type definitions.

```
npm install @types/express --save-dev
```

Step 9: Enjoy Your TypeScript Project

With these steps, you have successfully converted a JavaScript project to TypeScript. Your project now benefits from TypeScript's type checking, which can help prevent many common errors.

Summary

1. Set up the initial JavaScript project.
2. Install TypeScript and initialize configuration.
3. Rename .js files to .ts.
4. Add type annotations and handle TypeScript errors.
5. Compile and run the project.
6. Address any remaining issues.

Example Directory Structure

After conversion, your project structure should look like this: