**Converting a JavaScript project to TypeScript**

Transitioning from JavaScript to TypeScript involves renaming files, updating import/export syntax, adding type annotations, and ensuring your project is properly configured to compile TypeScript. This step-by-step process allows for a gradual and manageable transition, enabling you to leverage the benefits of TypeScript, such as type safety and enhanced IDE support.

1. **Initialize a Node.js Project**

* npm init

1. **Install TypeScript and necessary packages:**

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

1. **Initialize a TypeScript Configuration File:**

* npx tsc –init

Convert JavaScript to TypeScript ->

Rename .js files to .ts:

src/index.js → src/index.ts

src/utils.js → src/utils.ts

Update Module Imports and Exports:

* Use ES module syntax instead of CommonJS.

For Ex->

import { greet } from './utils';

const name: string = 'World';

console.log(greet(name));

1. **Update tsconfig.json**

Ensure the “tsconfig.json” includes necessary settings for module resolution and output.

Here is json code ->

{

"compilerOptions": {

"target": "ES6",

"module": "commonjs",

"rootDir": "./src",

"outDir": "./dist",

"strict": true,

"esModuleInterop": true

},

"include": ["src"]

}

1. **Compile TypeScript**

Compile the TypeScript code:

* npx tsc

1. **Run the Compiled Code**

Execute the compiled JavaScript:

* node dist/index.js