**ধাপ ১: TypeScript ইনস্টলেশন ও কনফিগারেশন**

1. **TypeScript ইনস্টল করুন:**
2. npm install -g typescript

**2.প্রোজেক্ট ইনিশিয়ালাইজ করুন:**

tsc --init

এটি tsconfig.json ফাইল তৈরি করবে যেখানে কম্পাইলার অপশনস থাকবে।

1. **কনফিগারেশন ফাইল (tsconfig.json):**

{   
"compilerOptions": {

"target": "ES6", *// JavaScript ভার্সন (ES3, ES5, ES6)*"module": "CommonJS", *// মডিউল সিস্টেম (CommonJS, ES6)*   
"outDir": "./dist", *// কম্পাইলড ফাইল যেখানে সেভ হবে*   
"rootDir": "./src", *// সোর্স কোডের লোকেশন*   
"strict": true, *// Strict টাইপ চেকিং* "esModuleInterop": true *// ES6 মডিউল কম্প্যাটিবিলিটি*   
} }

### ****ধাপ ২: TypeScript ডেটা টাইপ (Data Types)****

TypeScript নিম্নলিখিত ডেটা টাইপ সাপোর্ট করে:

1. **Primitive Types:**
   * 1. **number**: সংখ্যা (integer, float)
        1. let age: number = 25;

**string**: টেক্সট

let name: string = "John";

**boolean**: true/false

let isStudent: boolean = true;

**Array & Tuple:**

* **Array**:

let numbers: number[] = [1, 2, 3];

**Tuple** (ফিক্সড সাইজের অ্যারে):

let user: [string, number] = ["John", 25];

**Enum**: কাস্টম টাইপ

enum Color { Red, Green, Blue }

let bgColor: Color = Color.Red;

**Any & Unknown**: যেকোনো টাইপ

let dynamicValue: any = "Hello";

dynamicValue = 20; *// কোনো এরর নেই*

**Union Types**: একাধিক টাইপ

let id: string | number = "ABC123";

### ****ধাপ ৩: TypeScript OOP (Object-Oriented Programming)****

TypeScript OOP-এর ৪টি মূল নীতি সাপোর্ট করে:

1. **Encapsulation (ক্যাপসুলেশন):**

class Person {

private name: string; // প্রাইভেট ভেরিয়েবল

constructor(name: string) {

this.name = name;

}

public getName(): string { // পাবলিক মেথড

return this.name;

}

}

**Inheritance (ইনহেরিটেন্স):**

class Student extends Person {

private course: string;

constructor(name: string, course: string) {

super(name);

this.course = course;

}

}

**Polymorphism (পলিমরফিজম):**

class Animal {

makeSound(): void {

console.log("Some sound");

}

}

class Dog extends Animal {

makeSound(): void { // মেথড ওভাররাইডিং

console.log("Bark!");

}

}

**Abstraction (অ্যাবস্ট্রাকশন):**

abstract class Shape {

abstract area(): number; // অ্যাবস্ট্রাক্ট মেথড

}

class Circle extends Shape {

constructor(private radius: number) { super(); }

area(): number {

return Math.PI \* this.radius \*\* 2;

}

}

### ****ধাপ ৪: কম্পাইল ও রান করা****

1. **কম্পাইল করুন:**

tsc

1. এটি src ফোল্ডার থেকে dist ফোল্ডারে JS ফাইল জেনারেট করবে।
2. **আউটপুট রান করুন:**