Day 17	SPA & Working with NPM SPA : Single page application SPA loads a single HTML page and dynamically updates the content without re-loading. NPM is a package manager for javascript	
	Type Script JSX	Var-hoisting Functional Score Classes & Methods & Constructor
		Maps Iterators Interfaces Declarations and Annotations
		Anytype Enumeration Decorator , Arrays & tuples

DEmo 4: Flexbox layout and semantic HTML(responsive web design)
Demo 5 : Bootstrap component (NavBar, Cards, Buttons and Alerts) A landing page using bootstrap grid and components.
Demo 6: Javascript form Validation + DOM Manipulation

Features	**JavaScript**	**TypeScript**
	Scripting lang for web development	**Superset of javascript**
Typed System	**Dynamically typed**	**Statically types(optional)**
Compilation	**Interpreted by by browser**	**Needs to be compiled to javascript using Typescript compiler(tsc)**
Error Checking	**Runtime errors**	**Compile-time error checking before execution**
Support ES6+	**Fully supported ES6+**	**Supports ES6+ and beyond with additional features**
OOP	**Supports OOP but less structures**	**Full supports classes, interfaces and access modifiers**
IDE Supporting	**Limited support for auto completion**	**Enhances tooling with intelliSense, better refactoring.**
Easeness to master	**Easier for beginners**	**Slightly steeper learning curve due to added**

		features and strictness
Code readability	Harder in large apps	Easier due to static types and self documenting code.
Uses	Front end scripting(browser) & NodeJs	Large-scale front-end/backend applications (Angular, NestJs)

Typical folder structure of typescript project : -

```
your-project-folder/  ( Rooth Direcory )
├── src/                ( all Typescript source files)
│   └── student.ts —> Where we write code using typescript
├── package.json  —--> (Project configuration/manifest file)
└── tsconfig.json —---> ( Typescript compiler configuration
```

Overall Steps for typescript program :

Step 1: We write code in src/student.ts file

Step 2: tsconfig.jon tells the compiler:

- **Take files from src/**
- **Output compiled JS to dist/**
- **Use these compiler rules**

Step 3: Package.json provides:

- **Command to run (npm start)**
- **Required dependencies(Typescript)**

Step 1: et upi VS CODE Environment

Step 2: Initialize a NodeJs Project (npm init -y) via CLI

Step 3: install typescript

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

Step 4: Create a typescript configuration via tsconfig.json

```
npx tsc --init
```

Step 5: Create a project structure

```
your-project-folder/
├── src/
│   └── student.ts
├── package.json
└── tsconfig.json
```

Step 6: Write a Student class

Step 7: Compile Typescript to javascript (npx tsc)

Step 8: Run the javascript file

```
Node dist/student.js
```