TYPESCRIPT

TypeScript is a **syntactic superset of JavaScript** which adds static typing. This basically means that TypeScript adds syntax on top of JavaScript, allowing developers to add types.

TypeScript being a "Syntactic Superset" means that it shares the same base syntax as JavaScript, but adds something to it.

JavaScript is a loosely typed language. It can be difficult to understand what types of data are being passed around in JavaScript.

Why to use TypeScript?

In JavaScript, function parameters and variables don't have any information! So, developers need to look at documentation, or guess based on the implementation. TypeScript allows specifying the types of data being passed around within the code, and has the ability to report errors when the types don't match. For example, TypeScript will report an error when passing a string into a function that expects a number. JavaScript will not.

A common way to use TypeScript is to use the official TypeScript compiler, which transpiles TypeScript code into JavaScript.

TypeScript being converted into JavaScript means it runs anywhere that JavaScript runs!

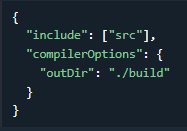
To install typescript compiler –

npm install typescript --save-dev

npx tsc ( to start compiler)

By default the TypeScript compiler will print a help message when run in an empty project. The compiler can be configured using a tsconfig.json file. You can have TypeScript create tsconfig.json with the recommended settings with: 

BY adding this in your tsconfig.json – This will configure the TypeScript compiler to transpile TypeScript files located in the src/ directory of your project, into JavaScript files in the build/ directory.



TypeScript Simple Types

TypeScript supports some simple types (primitives) you may know like Boolean, number and string.

When creating a variable, there are two main ways TypeScript assigns a type: Explicit & Implicit

Explicit - writing out the type: 

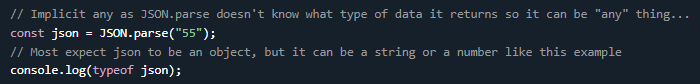
Implicit - TypeScript will "guess" the type, based on the assigned value: 

Having TypeScript "guess" the type of a value is called infer.

Error In Type Assignment -> TypeScript will throw an error if data types do not match.

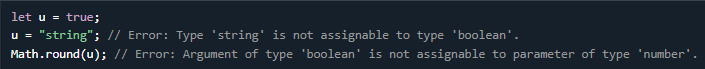
 Will give an error

NOTE - TypeScript may not always properly infer what the type of a variable may be. In such cases, it will set the type to any which disables type checking. A really good example is is json parsing –

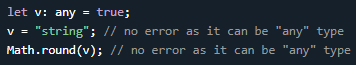


This behavior can be disabled by enabling noImplicitAny as an option in a TypeScript's project tsconfig.json. That is a JSON config file for customizing how some of TypeScript behaves.

Type : ANY -> any is a type that disables type checking and effectively allows all types to be used. The example below does not use any and will throw an error: Typical and predictable



Setting any to the special type any disables type checking:



any can be a useful way to get past errors since it disables type checking, but TypeScript will not be able provide type safety, and tools which rely on type data, such as auto completion, will not work. Remember, it should be avoided at "any" cost...

Type: unknown ->

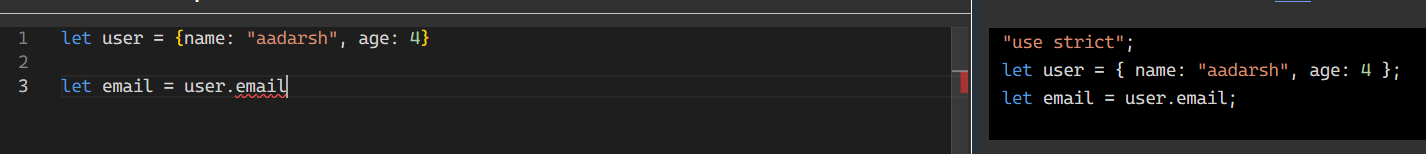
Video Course –

Typescript allows you to write JS in precise mannerso that code faces lot less error in run time. In fact all the code written on typescript is eventually compiled into JS. Type script is all about typeSafty.

For eg 2 + “2” = ‘22’. Now this is not type safty, js should have stopped the addition of a number and a string. Similarly you are allowed to add null + 2, or undefined + 2, et…..

Typescript involves STATIC CHECKING - > entire langualage parser or syntaxes of the code are analyzed by the IDE. But this is not the case with js. Hence TS is all about Static checking.TS is more lengthy for same code than JS.

TypeScript is a top layer on JS, It’s a super set of JS. It is a development tool. -> Helps you write better code with less problems and more scalability.Its like a wrapper around the js.



TS is entirely dependent on JS, not a standalone language.

.ts for typescript file and tsx -> for a component level thing, just like js and jsx.

So whole idea of TS to prevent from the errors one might easily fall into.