freeCodeCamp - TS and Basic TS

08 November 2024

11:28

**Basics**

* Created at Microsoft to overcome drawbacks of JS
* Typescript is superset of JavaScript.
* Static typing(c++, c#, java) vs Dynamic typing(python, JS, Ruby)
* Static typing - once we assigned a value as int num=10; we can't assign it a new value as number ="a", whereas it is possible for dynamically typed languages.
* JS + TYPE checking = TS
* Vscode has TS inbuilt while we write JS. Thus giving the suggestions
* Adv of TS - static typing, code completion, debugging, refactoring, shorthand notations.
* All ts files has extension as .ts. To run a ts file from terminal the cmd is - tsc filename.ts
* We can specify the type of a variable, in this case as a number - let age : number = 20;

**TypeScript Compiler**

* TS compiler helps us to catch errors during compile time.
* The TS compiler converts the TS code to JS to run the same.
* We can make the TS compiler to point to a newer JS version. Cmd tsc --init which creates the tsconfig.json file.
* In the tsconfig.json, we can put 'target' value as any JS version such as ES2016.
* In the tsconfig.json, we can put 'rootDir' value as ./src to put all source ts files in same folder.
* In the tsconfig.json, we can configure 'outDir' value to a specific folder, so that the compiler generated JS file will get stored in that folder. For better folder structuring.
* In the tsconfig.json, we can configure 'removeComments' as true will prevent the comments written in the TS file from being generated in the compiler generated JS file.
* In the tsconfig.json, we can configure 'noEmitOnError' as true which on encountering any error while running the TS file won't generate the equivalent JS file.
* Once with above configurations, in terminal we can type just tsc to compile all the ts files in the project.

**Debugging TS**

* Launch.json file, parameters - preLaunchTask, outFiles.

**Typescript - Built-in Types**

* JS types - number, string, null, undefined, boolean, array, object.
* TS additional types - any, unknown, never, enum, tuple.
* Type 'any' - a variable is of this type when it is declared but not assigned any value. Avoid using 'any'.
* We can specify the type as number[] like below. But if it's all numbers then TS implicitly knows it’s a number array without explicitly specifying.
* Machine generated alternative text:
  1.et numbers: number[J  [1 2 31