**Generic Information**

TypeScript is a programming language that is built on top of JavaScript. So, every JavaScript file is a valid typescript file. Any JavaScript code is valid in TypeScript.   
TypeScript is essentially JavaScript with type checking.

TypeScript is a free and open-source high-level programming language developed by Microsoft that adds static typing with optional type annotations to JavaScript.

**Benefits:**

* Static typing:  
  There is two types of programming languages are:  
  Statically-Typed & Dynamically-Typed  
    
  Statically-Typed languages like (C++, C#, Java), here we know the types of variables at compile time or while coding.  
  For Ex: int number = 10;  
    
  Dynamically-Typed languages like (JavaScript, Python, Ruby), here the types of variables is dynamic. So it is determined at runtime and we can only caught error at runtime instead of compile time.  
  For ex:   
  let number = 10;  
  number = “a”;
* Code completion
* Refactoring
* Shorthand notations

**Drawbacks:**

* Compilation: There is always a completion step involved because at this time browser don’t understand typescript code.   
  So, we’ve to give typescript code to the typescript compiler to compile and translate into javascript. This process is called transpilation.
* Discipline in coding: We’ve to a bit more discipline in writing code.

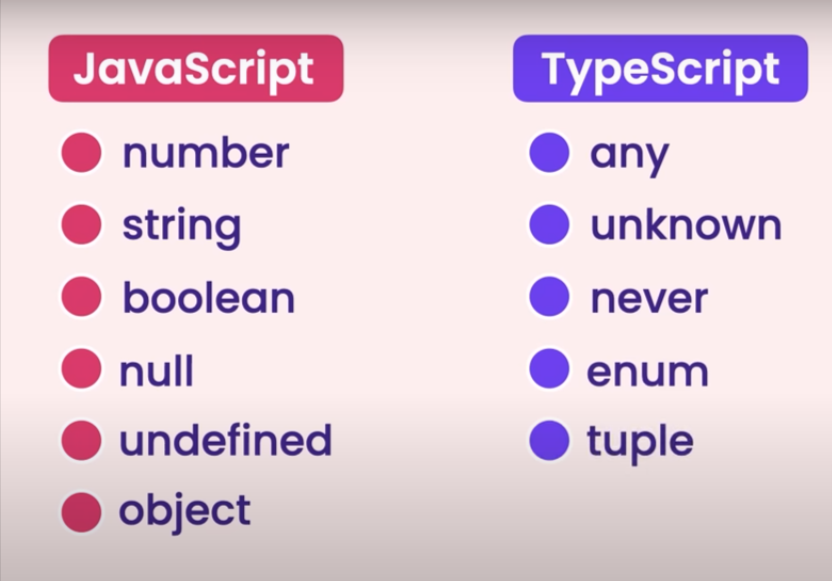
**Configuring TypeScript Compiler:**

By default, typescript compiler converted its code to an older version of javascript i.e. ES5

So to make ts compiler to convert its ts code to modern js, we’ve to follow the below steps:

* *“tsc –init”*Initializing typescript. This will create a typescript configuration file i.e. tsconfig.json.
* In tsconfig.json file,   
  Under Language and Environment section, by default,   
  “target”: “es2016”  
  That is ES6 (Valid in all browsers)  
  So, in this target property we can set the JavaScript language version for emitted JavaScript and include compatible library declarations as we want.
* Under Modules Section:  
  uncomment, “rootDir”: “./”  
  and provide the path value where you’ll contain your TypeScript files.  
  For ex: “rootDir”: “./src”  
  Source folder
* Under Emit Section:  
  uncomment, “outFile”: “./”  
  and provide the path value where you’ll contain your JavaScript files.  
  So, whenever we compile our .ts file, our ts compiler will generate .js file at the above specified path of “outFile”.  
  For ex: “outFile”: “./dist”  
  Distributable folder
* Under Emit Section:  
  uncomment, “removeComments”: true  
  This will avoid generation of commented javascript code from typescript by ts compiler
* Under Emit Section:  
  uncomment, “noEmitOnError”: true  
  This will avoid generation javascript file if we encounter with any error at compile time.
* Under Emit Section:  
  uncomment, “sourceMap”: true  
  sourceMap is a file that specifies how each line of our typescript code maps to the generated javascript code.  
  This basically help to debug our typescript code.

**Built-in Datatypes in Javascript and extended types in TypeScript:-**

****

**Note:**

* For simple projects vanilla javascript is best. For medium to large projects typescript is good to go.
* We can compile TypeScript code by running the following command in the terminal:  
  *“tsc <file\_name>.ts”*

Or

We can compile all TypeScript file by running the following command in the terminal: *“tsc”*After execution of the above compilation command, compiler will automatically create a javascript file with the same name with .js extension and add its converted typescript code to it i.e. (javascript code).  
  
  
So, we can’t write manually any random js code to javascript file by creating it manually with the same name as .ts file before compilation.  
Because after compilation, ts compiler will removed that js code to the manually created .js file with same as .ts file and override its javascript code that is converted from its own typescript code.