1.typescript需要nodejs环境，先安装nodejs

2.利用npm命令全局安装：

|  |
| --- |
| npm install typescript -g |

3.测试安装：

|  |
| --- |
| 在命令行界面输入：tsc  出现如下界面：  C:\Users\Administrator.WIN-9JP9A443C48>tsc  Version 4.6.2  tsc: The TypeScript Compiler - Version 4.6.2  TS  COMMON COMMANDS  tsc  Compiles the current project (tsconfig.json in the working directory.)  tsc app.ts util.ts  Ignoring tsconfig.json, compiles the specified files with default compiler options.  tsc -b  Build a composite project in the working directory.  tsc --init  Creates a tsconfig.json with the recommended settings in the working directory.  tsc -p ./path/to/tsconfig.json  Compiles the TypeScript project located at the specified path.  tsc --help --all  An expanded version of this information, showing all possible compiler options  tsc --noEmit  tsc --target esnext  Compiles the current project, with additional settings.  COMMAND LINE FLAGS  --help, -h Print this message.  --watch, -w Watch input files.  --all Show all compiler options.  --version, -v Print the compiler's version.  --init Initializes a TypeScript project and creates a tsconfig.json file.  --project, -p Compile the project given the path to its configuration file, or to a folder with a 'tsconfig.json'.  --build, -b Build one or more projects and their dependencies, if out of date  --showConfig Print the final configuration instead of building.  COMMON COMPILER OPTIONS  --pretty Enable color and formatting in TypeScript's output to make compiler errors easier to read  type: boolean  default: true  --target, -t Set the JavaScript language version for emitted JavaScript and include compatible library decla rations.  one of: es3, es5, es6/es2015, es2016, es2017, es2018, es2019, es2020, es2021, es2022, esnext  default: es3  --module, -m Specify what module code is generated.  one of: none, commonjs, amd, umd, system, es6/es2015, es2020, es2022, esnext, node12, nodenext  default: undefined  --lib Specify a set of bundled library declaration files that describe the target runtime environment .  one or more: es5, es6/es2015, es7/es2016, es2017, es2018, es2019, es2020, es2021, es2022, esnext, dom, dom.i terable, webworker, webworker.importscripts, webworker.iterable, scripthost, es2015.core, es201 5.collection, es2015.generator, es2015.iterable, es2015.promise, es2015.proxy, es2015.reflect, es2015.symbol, es2015.symbol.wellknown, es2016.array.include, es2017.object, es2017.sharedmemor y, es2017.string, es2017.intl, es2017.typedarrays, es2018.asyncgenerator, es2018.asynciterable/ esnext.asynciterable, es2018.intl, es2018.promise, es2018.regexp, es2019.array, es2019.object, es2019.string, es2019.symbol/esnext.symbol, es2020.bigint/esnext.bigint, es2020.promise, es2020 .sharedmemory, es2020.string, es2020.symbol.wellknown, es2020.intl, es2021.promise/esnext.promi se, es2021.string, es2021.weakref/esnext.weakref, es2021.intl, es2022.array/esnext.array, es202 2.error, es2022.object, es2022.string/esnext.string, esnext.intl  default: undefined  --allowJs Allow JavaScript files to be a part of your program. Use the `checkJS` option to get errors fro m these files.  type: boolean  default: false  --checkJs Enable error reporting in type-checked JavaScript files.  type: boolean  default: false  --jsx Specify what JSX code is generated.  one of: preserve, react, react-native, react-jsx, react-jsxdev  default: undefined  --declaration, -d Generate .d.ts files from TypeScript and JavaScript files in your project.  type: boolean  default: `false`, unless `composite` is set  --declarationMap Create sourcemaps for d.ts files.  type: boolean  default: false  --emitDeclarationOnly Only output d.ts files and not JavaScript files.  type: boolean  default: false  --sourceMap Create source map files for emitted JavaScript files.  type: boolean  default: false  --outFile Specify a file that bundles all outputs into one JavaScript file. If `declaration` is true, als o designates a file that bundles all .d.ts output.  --outDir Specify an output folder for all emitted files.  --removeComments Disable emitting comments.  type: boolean  default: false  --noEmit Disable emitting files from a compilation.  type: boolean  default: false  --strict Enable all strict type-checking options.  type: boolean  default: false  --types Specify type package names to be included without being referenced in a source file.  --esModuleInterop Emit additional JavaScript to ease support for importing CommonJS modules. This enables `allowS yntheticDefaultImports` for type compatibility.  type: boolean  default: false  You can learn about all of the compiler options at <https://aka.ms/tsconfig-reference>  说明安装成功。 |

4.建议使用vscode作为typescript的开发工具。

5.创建一个文件夹如typescriptproj，用vscode打开，新建一个文件如hello.ts.

输入如下代码：

|  |
| --- |
| function hello(){      let web:string="Hello typeScript!!!"      console.log(web);    }  hello() |

6.打开一个终端，输入命令tsc hello.ts 将ts转换为js，

|  |
| --- |
| tsc hello.js |

如果出现如下错误，需要配置powershell

|  |
| --- |
| tsc : 无法加载文件 C:\Users\Administrator.WIN-9JP9A443C48\AppData\Roaming\npm\tsc.ps1，因为在此系  统上禁止运行脚本。 |

7.以管理的身份打开powershell，输入如下命令：set-ExecutionPolicy RemoteSigned回车，出现询问，答Y

8.重新执行步骤6，成功生成js文件。可以执行js文件。

**9.如果想直接运行ts文件，需要安装一个叫做ts-node的组件。**

|  |
| --- |
| npm install –g ts-node |

10.安装完ts-node后，运行ts-node hello.ts还是报错，说找不到console，

|  |
| --- |
| hello.ts:3:5 - error TS2584: Cannot find name 'console'. Do you need to change your target library? |

11.解决办法：

|  |
| --- |
| npm install @types/node --save-dev |