{

"compilerOptions": {

"incremental": true, // TS编译器在第一次编译之后会生成一个存储编译信息的文件，第二次编译会在第一次的基础上进行增量编译，可以提高编译的速度

"tsBuildInfoFile": "./buildFile", // 增量编译文件的存储位置

"diagnostics": true, // 打印诊断信息

"target": "ES5", // 目标语言的版本

"module": "CommonJS", // 生成代码的模板标准

"outFile": "./app.js", // 将多个相互依赖的文件生成一个文件，可以用在AMD模块中，即开启时应设置"module": "AMD",

"lib": ["DOM", "ES2015", "ScriptHost", "ES2019.Array"], // TS需要引用的库，即声明文件，es5 默认引用dom、es5、scripthost,如需要使用es的高级版本特性，通常都需要配置，如es8的数组新特性需要引入"ES2019.Array",

"allowJS": true, // 允许编译器编译JS，JSX文件

"checkJs": true, // 允许在JS文件中报错，通常与allowJS一起使用

"outDir": "./dist", // 指定输出目录

"rootDir": "./", // 指定输出文件目录(用于输出)，用于控制输出目录结构

"declaration": true, // 生成声明文件，开启后会自动生成声明文件

"declarationDir": "./file", // 指定生成声明文件存放目录

"emitDeclarationOnly": true, // 只生成声明文件，而不会生成js文件

"sourceMap": true, // 生成目标文件的sourceMap文件

"inlineSourceMap": true, // 生成目标文件的inline SourceMap，inline SourceMap会包含在生成的js文件中

"declarationMap": true, // 为声明文件生成sourceMap

"typeRoots": [], // 声明文件目录，默认时node\_modules/@types

"types": [], // 加载的声明文件包

"removeComments":true, // 删除注释

"noEmit": true, // 不输出文件,即编译后不会生成任何js文件

"noEmitOnError": true, // 发送错误时不输出任何文件

"noEmitHelpers": true, // 不生成helper函数，减小体积，需要额外安装，常配合importHelpers一起使用

"importHelpers": true, // 通过tslib引入helper函数，文件必须是模块

"downlevelIteration": true, // 降级遍历器实现，如果目标源是es3/5，那么遍历器会有降级的实现

"strict": true, // 开启所有严格的类型检查

"alwaysStrict": true, // 在代码中注入'use strict'

"noImplicitAny": true, // 不允许隐式的any类型

"strictNullChecks": true, // 不允许把null、undefined赋值给其他类型的变量

"strictFunctionTypes": true, // 不允许函数参数双向协变

"strictPropertyInitialization": true, // 类的实例属性必须初始化

"strictBindCallApply": true, // 严格的bind/call/apply检查

"noImplicitThis": true, // 不允许this有隐式的any类型

"noUnusedLocals": true, // 检查只声明、未使用的局部变量(只提示不报错)

"noUnusedParameters": true, // 检查未使用的函数参数(只提示不报错)

"noFallthroughCasesInSwitch": true, // 防止switch语句贯穿(即如果没有break语句后面不会执行)

"noImplicitReturns": true, //每个分支都会有返回值

"esModuleInterop": true, // 允许export=导出，由import from 导入

"allowUmdGlobalAccess": true, // 允许在模块中全局变量的方式访问umd模块

"moduleResolution": "node", // 模块解析策略，ts默认用node的解析策略，即相对的方式导入

"baseUrl": "./", // 解析非相对模块的基地址，默认是当前目录

"paths": { // 路径映射，相对于baseUrl

// 如使用jq时不想使用默认版本，而需要手动指定版本，可进行如下配置

"jquery": ["node\_modules/jquery/dist/jquery.min.js"]

},

"rootDirs": ["src","out"], // 将多个目录放在一个虚拟目录下，用于运行时，即编译后引入文件的位置可能发生变化，这也设置可以虚拟src和out在同一个目录下，不用再去改变路径也不会报错

"listEmittedFiles": true, // 打印输出文件

"listFiles": true// 打印编译的文件(包括引用的声明文件)

}

}

