什么是编译

JS的运行环境(浏览器和nodeJS)不认识TS代码，所以TS想运行，必须先转化为JS代码，这个转化就叫做编译。

TS官方提供了编译器，编译时会将类型声明和类型相关的代码全部删除，只留下能运行的JS代码，不会改变JS的运行结果。

注意点:

TS的类型检查是编译时的类型检查，而不是代码运行时的类型检查，一旦代码编译为JS，运行时就不再进行检查了。

编译器

TS官方提供的编译器叫做tsc，它可以将ts代码编译成js代码，如果想编译ts代码，必须先安装tsc。

根据约定，ts脚本文件使用`.ts`后缀名,js脚本文件使用`.js`后缀名，tsc作用就是把`.ts`脚本转为`.js`脚本。

1. 安装

是一个npm模块，可以全局安装，也可以在项目中安装为一个依赖模块。

```
npm install -g typescript
```

检测是否安装成功,`-v`或`--version`参数检测当前安装的tsc版本。

```
tsc -v
Version 5.1.6
```

`-h`或`--help`参数输出基本帮助信息，`--all`参数查看完整帮助信息。

```
tsc --all
```

2. 编译脚本

tsc命名后面，加上ts脚本文件，就可以将其编译成js脚本。

会直接在当前目录下生成一个.js脚本文件，这个脚本就是完全编译后生成的js代码，

也可以一次编译多个ts脚本。

```
tsc app.ts
tsc file1.ts file2.ts file3.ts
```

tsc后还可以添加参数，调整编译行为。

- --outFile

将多个ts脚本编译成一个js文件

```
tsc file1.ts file2.ts --outFile app.js
```

- --outDir

编译的时候默认保存到当前目录，使用`--outDir`可以指定保存到其他目录。

```
tsc app.ts --outDir dist
```

- --target

指定编译后JS的版本，建议使用es2015

因为为了保证编译的结果能在各种JS引擎运行，tsc默认会将ts代码编译成低版本的JS，默认是es3版本。

```
tsc --target es2015 app.ts
```

3. 编译错误的处理

编译过程中，如果没有报错，tsc命令不会显示任何信息。如果编译报错，则会显示报错信息，但是默认依然会编译生成JS脚本。

假如想一旦报错就停止编译，不生产编译产物，可以使用`--noEmitOnError`参数。

```
tsc --noEmitOnError app.ts
```

`--noEmit` 只检查类型是否正确,不生成JS文件。

```
tsc --noEmit app.ts
```

4. tsconfig.json

tsc的编译参数不仅可以写在命令行里，也能写在配置文件中。[tsconfig.json]