

简介：typescript具有类型系统，且是JavaScript的超集。它可以编译成普通的JavaScript代码。

TypeScript支持任意浏览器，任意环境，任意系统并且是开源的。

安装：

```
npm install -g typescript
```

编译：

```
tsc greeter.ts
```

类型注解 `variable: datatype`

接口：声明一个对象，设置数据类型

```
interface Person {  
    firstName: string;  
    lastName: string;  
}
```

类：类比es6新特性，可设置属性类型，公有属性(`public`) 或私有属性(`private`)

入门

1.1 基础类型

数字，字符串，结构体，布尔值；支持与Js几乎相同的数据类型，枚举类型

模板字符串： `` I'm string!`` `${string}`

数组泛型： `Array<eleType>`

元组： `Tuple`元组类型允许表示一个已知元素数量和类型的数组，各元素的类型不必相同

enum：JavaScript标准数据类型的补充，可以为一组数值赋予友好的名字。

```
enum Color {Red, Green, Blue}
```

任意值: any

void: 空值, void类型像是与any类型相反, 它表示没有任何类型。

null, undefined: 默认情况下null和undefined是所有类型的子类型。

never: 表示永不存在的值的类型。never类型是任何类型的子类型, 也可以赋值给任何类型; 然而, 没有类型是never的子类型或可以赋值给never类型 (除了never本身之外)。即使any也不可以赋值给never

类型断言 : 不进行特殊的数据检查和解构。没有运行时的影响, 只是在编译阶段起作用。

<> as

在TypeScript里使用JSX时, 只有as语法断言是被允许的。

1.2 变量声明

var: 声明一个变量 函数作用域

let: js新的变量生命方式 块作用域

const: 声明一个被赋值后不可改变的常量, 引用的值不可变

块作用域:

块作用域中的变量不能在声明之前读写 (暂时性死区)

可以在一个拥有块作用域变量被声明前获取它, 但无法调用该函数, 直到变量声明

在包含块之前或for循环外不可访问

catch声明变量也有同样的规则

重定义及屏蔽:

多次用var声明同个变量只会得到一个变量且不会报错

用let声明后的变量再次用var或let声明会报错

块级作用域变量需要在明显不同的块里声明

在一个嵌套作用域里引入一个新名字的行为称做屏蔽。

解构

数组解构

对象解构

展开