TypeScript

1、TypeScript

1.1、以JavaScript为基础建立的语言。

1.2、一个JavaScript的超集。

1.3、可以在任何支持JavaScript的平台上运行。（不可以被js解析器直接执行，通过编译工具将TypeScript编译为JavaScript）

1.4、拓展了JavaScript，并且添加了类型。

1.5、支持ES的新特性，添加ES不具备的新特性，比如接口，继承这些。

2、开发环境搭建

2.1、开发环境搭建：

首先下载node，安装方法很简单，直接搜官网地址下载安装即可。<https://nodejs.org/zh-cn/>，安装node之后就可以使用npm拉取TypeScript了，win+r，输入cmd，进入命令行窗口，先输入node -v查看是否安装了node，之后输入npm i -g typescript，全局的TypeScript环境就拉取好了。

3、TypeScript类型声明

3.1、当申明变量的类型时，在一个TypeScript文件中不可赋值成其他形式的内容。当代码有错误的时候可以去编译，不过后期可以限制这样编译方式。

//定义一个变量a，并且设置其类型为number

let a: number;

//给a赋值

a = 10;

a = 30;

//当变量的类型声明之后，不可以通过赋值的形式将不同类型的值赋给该变量

a = 'bugdr'

3.2、如果在声明变量的同时进行变量赋值，那么TypeScript会自动的声明该变量的类型。

//给一个变量赋值为Boolean类型，ts默认将这个变量的类型声明为Boolean类型，

let b = false;

b = true

//报错

b = 123

3.3、声明函数时，函数参数可以使用类型声明，同时函数的返回值也是可以进行类型的声明。

//sum(a: number,b: number)a和b都声明了类型为number，函数的返回值为number类型

function sum(a: number,b: number): number{

    return a + b;

}

3.4、TypeScript中常用的类型。

|  |  |  |
| --- | --- | --- |
| **类型** | **例子** | **描述** |
| number | 1, -1, 0.0 | 任意数字 |
| string | 'hi', "hi", hi | 任意字符串 |
| boolean | true、false | 布尔值true或false |
| 字面量 | 其本身 | 限制变量的值就是该字面量的值 |
| any | \* | 任意类型 |
| unknown | \* | 类型安全的any |
| void | 空值（undefined） | 没有值（或undefined） |
| never | 没有值 | 不能是任何值 |
| object | {name:'孙悟空'} | 任意的JS对象 |
| array | [1,2,3] | 任意JS数组 |
| tuple | [4,5] | 元素，TS新增类型，固定长度数组 |
| enum | enum{A, B} | 枚举，TS中新增类型 |

3.4.1、number：

let A: number = 6;

let B: number = 0xf00d;

let C: number = 0b1010;

3.4.2、string

let D: string = '你好';

let E: string = 'bugdr';

let F: string = '学习TypeScript';

3.4.3、boolean

let G: boolean = true;

let H: boolean = false;

3.4.4、字面量

let I: 1 | 2 | 3;

let J: '我是' | 'bugdr';

3.4.5、any

let K: any;

K = 'bugdr';

K = 123;

3.4.6、unknown

//unknown类型

let L: unknown = 123;

L = 'bugdr';

3.4.7、void

//void类型

let M: void = undefined;

3.4.8、never

function Me(message: string): never {

throw new Error(message);

}

3.4.9、object

//对象

let object: object = {};

3.4.10、array（数组）

//数组

let N: string[] = ["123","bugdr","你好"];

let O: Array<number> = [1, 2, 3];

3.4.11、tuple（固定长度数组），语法表示如下。

//固定长度的数组

//[类型,类型,类型],长度可控制

let x: [string, number];

x = ["bugdr", 123];

3.4.12、enum

//枚举，定义枚举

enum Color {

    Red,

    Green,

    Blue,

}

let d: Color = Color.Green;