

数据类型

number,string,boolean

数字，字符串，布尔型

any

表示任何类型

unknown

类型安全的any，unknown的变量不能直接赋值给其他变量

```
let b: unknown;
let bb: string;

// b = 123; //success
// bb = b; //error

// b = 'xiaomi';
// bb = b; //error
// 如何成功
/* 类型断言：用来告诉解析器变量的实际类型
   语法：
       变量 as 类型
*/
b = 'xiaomi';
bb = b as string; //success
if (typeof b === 'string') bb = b //success

/* any类型，不报错,怎么设置都可以，可以赋值给任意变量，可见any真的不到万不得已不能使用 */
let c: any;
c = 999;
bb = c; //success

c = ['123'];
bb = c; //success
```

void

表示没有值或者undefined

```
function fn(): void {  
    // return '123'; //error  
    // return 'xiaomi'; //error  
    return; //success  
    // return undefined; //success  
    // return null; //success  
    //或者不写return  
}  
let data: void;  
// data = '123'; //error  
// data = [123]; //error  
// data = undefined; //success  
// data = null; //success
```

never

从不会出现的值，没有值，不能是任何值,永远不会有返回结果（返回“从不”的函数不能具有可访问的终结点）

```
let good: never;  
// good = '123'; //error  
// good = undefined; //error  
// good = null; //error  
  
function goodFun(): never {  
    // return '123'; //error  
    // return; //error  
    // return undefined; //error  
    // return null; //error  
    // console.log('123'); //error  
    throw new Error('error'); //success  
}
```

object

需要留意这两个特殊点

```
// 比如要求 name属性, salary属性这两个属性必须要有, 其他的属性随便添加修改
// job = { name: 'ios', salary: 11, company: '阿里巴巴' }; //error,
// 多设置了属性名, 此属性名并未在job中声明
// [propName: string]: type; type:any:表示任意类型的属性;
// type:string:表示属性的值只能为string 同理...
let job2: { name: string, salary: number, [propName: string]: any }
job2 = { name: 'ios', salary: 11, company: '阿里巴巴', boss: '马云'
}; //success
```

```
// 比如这个时候, 我们只想设置一个name, salary不想设置, 这个时候是报错的, 修改如下
// job = { name: 'ios' }; //error
let job1: { name: string, salary?: number } //属性 ? : type 表示此属性可有可无
job1 = { name: 'android' }; //success
```

array

两种声明方式

```
let a:number[];
let a:Array<number>;
```

tuple

固定长度的数组, 多了少了都不行

```
let a:[string,string]
a=["1","2"] //success
a=["1","2","3"] //error
```

enum

```
enum Gender {  
    woman,  
    man,  
    //woman = 0,  
    //man=1  
}  
  
let person: { name: string, gender: Gender };  
person = {  
    name: '科比',  
    gender: Gender.man  
}  
  
if (person.gender === Gender.man) console.log('男');  
if (person.gender === Gender.woman) console.log('女')
```

字面量

类型的就是值，不可变

```
let a:10;  
a = 1 //false  
a = '1' // false
```

特殊点

```
let a:'a'|123|false;  
a = 'a' //true  
a = 123 //true  
a = false //true
```