1. 类型注解 类型会自动推断，但是不好

:String any bool number 类名 变量类型 方法类型 method:string()

编译时类型检查 let 数组 any[][] 元组类型 let x:[1,2,””] ，就是属性

枚举

类 接口

Class A (

Property:type;

Constructor( )

)

箭头函数的this 是不会穿透的 ， 替换匿名函数

Es6属性语法： class a:number; 可以用分号结束 this.

方法不用写function

访问修饰符 public private private color:string;

派生类 extends 重写方法 return

模板字符串 let const 插值变量${} 插值函数

参数默认值 :””

可选参数 ?:string 扩展运算符 ... 把数组展开成参数

生成器，手动控制代码的执行步骤 …… 在任何地方断点 function \*

承诺promise 包装了似乎更好 async await

**析构，把对象和数组拆解为变量 ,快速拆分出变量来，不用麻烦地使用各种循环啊**

箭头表达式，匿名函数运行在函数环境中，会导致this setInterval丢失**…………用来在对象中调用匿名函数 所有需要用到匿名回调的函数都是如此**

forEahch ((value)=) ) 遍历并处理

for in（遍历键名或者索引）， for(let i of **) 遍历值 可以用于数组，对象，break，字符串上**

**es6式写法 ：属性 a:sting; 函数private work() {}** 没有 返回值就不用写方法类型

局部变量使用let或者var

泛型，约束集合 :array<String>  **接口**

**模块 导出 导入 export 类 变量 函数 所有**

注解（装饰器)：给框架使用的，让框架实现自动扫描注解来注释

附加在类，方法， 属性 ，参数 ，访问符

装饰器求值 监视修改替换类定义 （运行时装饰器就是个函数）

类装饰器：参数是构造函数的名字 sealed 冻结类

方法装饰器 参数： 构造或者原型……成员方法的名字，属性描述对象

@enumerable 修改成可枚举

访问符装饰器 configurable

属性装饰器 ：可以修改属性行为

参数装饰器

类型定义文件 d.ts