**TypeScript的视频笔记**

注意：官网有编译器，且可以对比ts和js

1. ES是JS语言的标准，JS是对ES标准的实现，标准和语言是独立存在的。在不加任何修饰的情况下，提到ES5、ES6默认都指JS语言的实现。

TS是JS的超集，在其基础上增加了静态类型的概念，本质上还是对ES标准的一种实现。

TS诞生于ES6标准发布之后，因此打一发布就支持ES6标准。把它和ES6紧密联系起来可以，但否认JS和ES6的关系这就是严重的误导了。（别不当回事，很多新手以为JS和jQuery是两种语言，一定程度就是受了误导）

2.compiler的原因：将typescript转成js，因为还有很多浏览器不支持tp

特性

一、字符串

1. 多行字符串 双撇号，即`` 可以生成多行字符串，不用像 js用+进行连接换行的字符串
2. 字符串模板

在多行字符串里用${}插入变量或方法的调用，只在多行字符串里才有效，若是””的字符串是无效的。注意：””表示普通的字符串

1. 自动拆分字符串

用字符串模板调用一个方法的时候，会自动将字符串和表达式的值传入方法中当做参数

二、参数类型

1. 用：指定变量的类型

var name: string = “abc”;

若是不指定类型，第一次赋值就认为它的类型是这个，后面改类型会报错

使用any，可以任意类型

类型： string number Boolean void(只能用于方法的返回类型，表示方法不需要任何返回值)

自定义类型

用class或接口声明自定义类型

用class声明的话，需要用new关键字来生成

1. 默认参数

参数的个数一定要和声明的个数一样才不会报错，但若指定了默认值的那个参数可以不传了。但带默认值的参数必须在最后声明

1. 可选参数

用？

b?: string

若没传可选参数，但方法里访问了这个参数，会报错

可选参数必须声明在必选参的后面

三、函数新特性

1.Rest and spread操作符：用来声明任意数量的方法参数

…args

// 反过来…args用来调用的话是es6的语法，目前ts不支持，js可以用的

2.generator函数

控制函数的执行过程，手工暂停和回复代码执行，如打断点

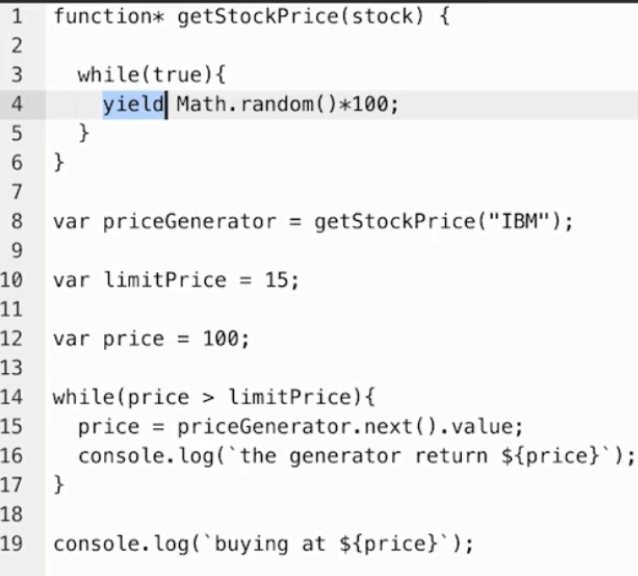
但ts还不支持，是es6的特性//可见不是所有的es6语法ts都支持啊

必须这样才能调用：



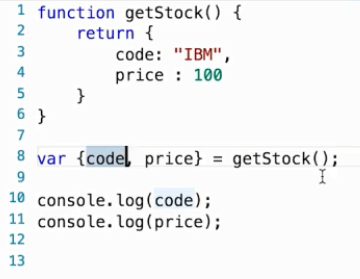
//获取0-100的随机数 怎么写？

//不太懂过程：



3.析构表达式

通过表达式将对象或数组拆解成任意数量的变量

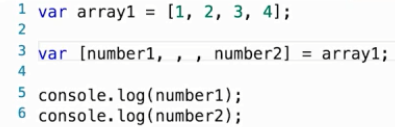


即将一个对象的属性分别赋给本地的表达式，注意，名字要相同，想名字不同就属性名:本地变量名的方式来实现

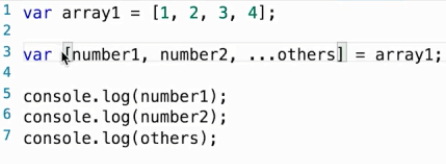
嵌套：



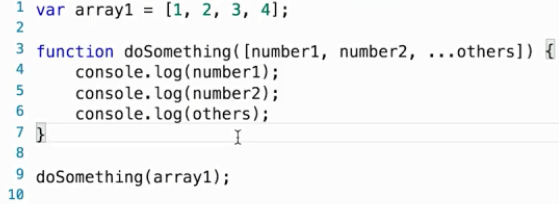
(2)从数组里取值：注意：这里是用的[],但不是，本地变量仍是变量，而没有变成数组哦！



复杂一点：

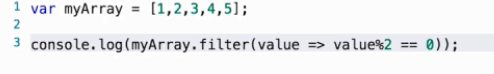


作为函数参数：



四、表达式和循环

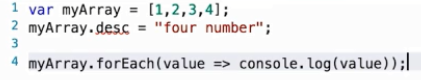
1.箭头函数



//filter函数要求传入一个箭头函数，满足条件的返回，不满足的过滤掉，这个例子运行结果是返回一个数组

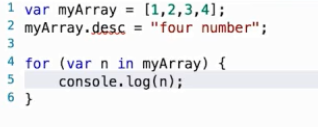
2.forEach(), for in 和for of

Ts里不能给数组指定一个属性，但js里可以

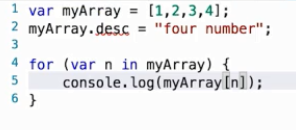


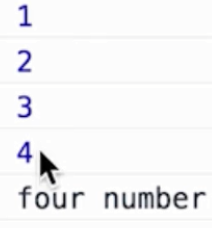
ForEach做循环不能用那个yield来中断，还有会忽略数组的属性值

For in

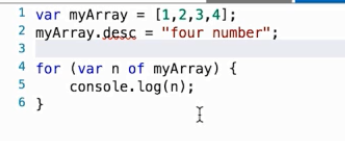


//结果是数组的下标和属性的名字，js里任何对象都是键值对，for in会输出所有的键（在这里是数组下标和属性名字）要想打出值，应该这样写：



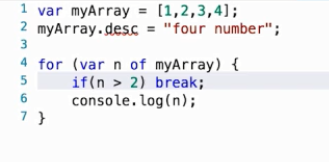


For of 循环出值，和for each比，不同点是可以打断点的，其他和for each一样，比如忽略属性

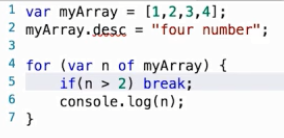




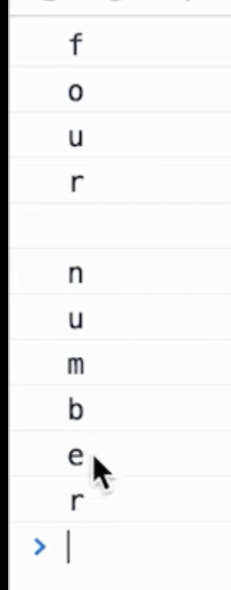
用break打断：



For of可以用在任何类型上，如对象，数组，字符串



结果是打出每个字符来：



面向对象的特性

类

一．

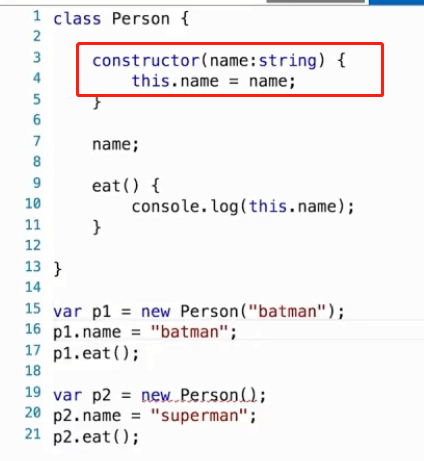
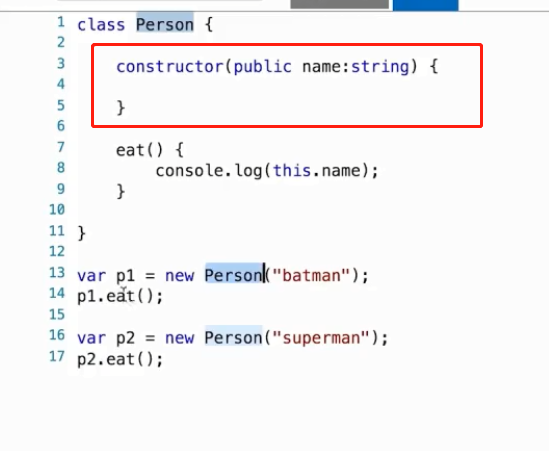
1. 类名好像要大写，注意类没有（）里写参数这一步了，若要传参则在constructor函数里定义参数
2. 实例化 用new
3. 访问控制符，在类的外部能不能被访问，默认是public

private 只能内部访问,protected可在内部和子类里被访问

//js里，类和继承都是通过这个关键字来实现的proptype

但在构造函数的参数声明时加public和不加是不一样的，若不加访问控制符，则后面使用这个变量会报错，意思是没有声明变量

1. coonstructor是被实例化时被调用一次的函数
2. 若构造函数有参数，则实例化时必须要传参进去才能不报错
3. 以下是相等的：

二、继承

1. Extense

继承父类的方法，还可以自己定义自己的属性和方法

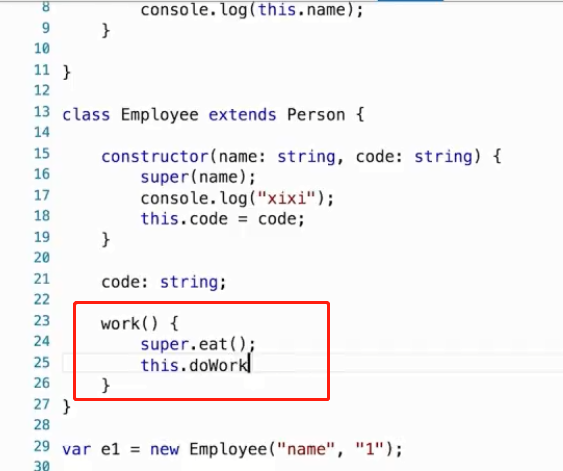
1. Super调用父类的构造函数或方法

子类的构造函数必须调用父类的构造函数，即在子类的constructor函数里用super方法

（2）super的第二种用法

调用父类的其他方法用super.func();

注意调用自己的函数要加this.关键字：



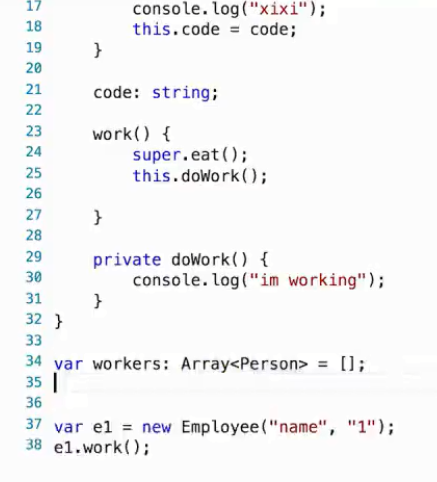
注意这里类的里面的方法的写法是啥都不声明直接写方法，和对象里的方法的写法也不一样，对象都是要属性：function fun(){} 这样的方式来写的。但类的属性写法和对象是一样写的

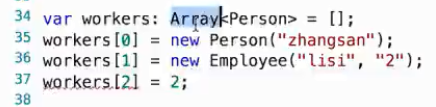
三、泛型

参数化的类型，一般用来限制集合的内容

指定类型里必须放的东西，用<＞

//比如这里表示数组里必须放person类型的数据，person是自己写的类

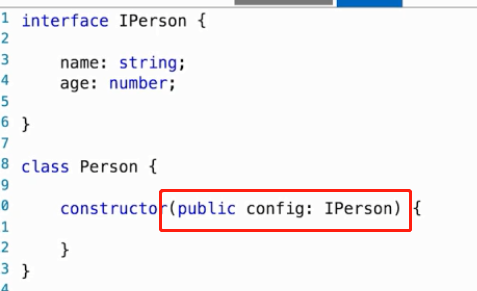


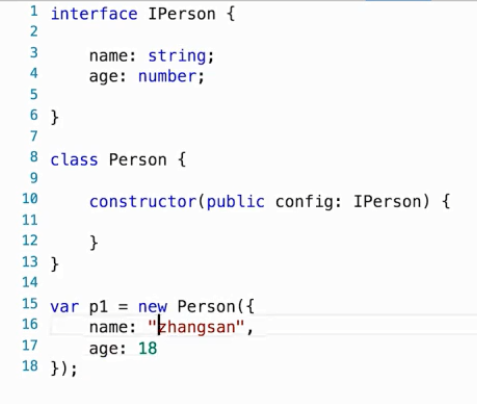


四、接口

在js里没有接口的概念，但ts里有

1. 作为一个方法的参数的类型声明，这里是用接口声明属性，注意没有参数





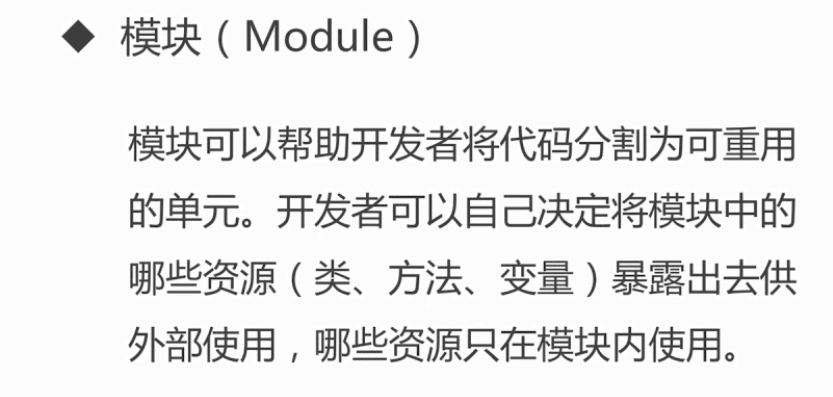
1. 接口声明方法



用一个类来实现该接口，类里必须实现接口里的那个方法，注意关键字implements



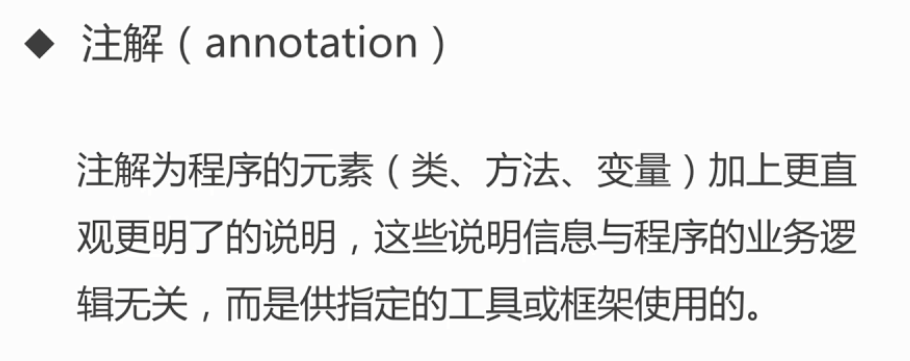
五、模块



一个模块就是一个文件

<https://www.imooc.com/video/13539>

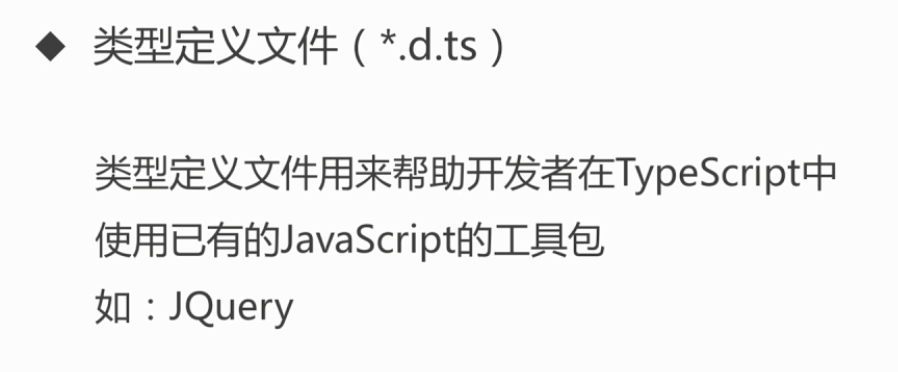
六、注解



告诉你的框架怎么来理解ts的一些类或者属性等

<https://www.imooc.com/video/13540>

七、类型定义文件



作用：把你要用的那些包里的类和方法以ts的模块的方式暴露出来，供你import

<https://www.imooc.com/video/13541>

这个工具用于安装类型定义文件

