

Js 字符串新特性: 多行字符串, 字符串模板 (ES6 也有)
自动拆分字符串 (你可以用一个字符串模板调用一个方法)

参数新特性: 1) 参数类型 2) 默认参数 3) 可选参数

eg: `var myname: string = "Lac"`
any 类型, 可以用任何值.

基本类型: `number, boolean, string, void` (不需要任何返回值), `any`

自定义类型:

eg:

```
class Person {  
    name: string  
    age: number  
}
```

```
var zhangsan: Person = new Person()  
zhangsan.name = "Bill"  
zhangsan.age = 18
```

带默认值参数声明在最后

~?

可选参数: `b?: string` (注意在方法体里处理)

函数新特性: 1) `rest and spread` 操作符 (ES6 应该都 OK 了)
2) 析构表达式

针对数组的: eg: `var arr = [1, 2, 3, 4]`

```
var [ num1, , num2 ] = arr
console.log(num1) // 1
console.log(num2) // 4
```

面向对象:

class 大体和 ES6 差不多.

可以使用 public, private protected
类的内部和子类

在 constructor 中 参数要 明确 声明 访问控制符.

super 关键字: 用来调用父类的构造函数
用来调用父类的其它方法

泛型:

eg: var workers: Array<Person> = []

接口 (Interface)

eg: interface IPerson {
 name: string
 age: number
}

1) 接口作为方法参数
的类型声明,
调用方法时, 会检查是否满足
接口定义

```
class Person {  
    constructor (Public config: IPerson) {  
    }  
}
```

var p1 = new Person ({ name: "zhan", age: 18 })



```
eg: interface Animal {  
    eat()  
}
```

2) 实现接口的方法

```
class sheep implements Animal {  
    eat() {  
        console.log("xxx")  
    }  
}
```

类型定义文件 (*.d.ts)