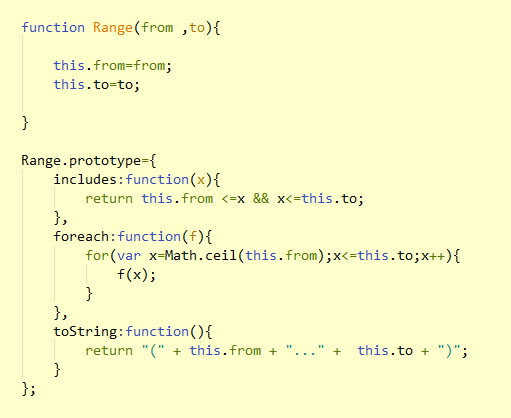
2017-12-27

JavaScript中，类的实现是基于其原型继承机制的。

往往是同一个构造函数创建并初始化的。

另外一个特点是——动态可继承，其根本原理在于JavaScript搜索方法和属性的顺序：先寻找私有的方法和属性，然后去其原型那里找。

使用构造函数的一个例子如下



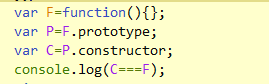
函数名首字母大写，构造函数规定写法

属性即方法，prototype中的属性和方法都被继承

并不需要return，因为使用new关键字调用

当且仅当两个对象继承自同一个原型对象的时候，他们才是属于同一个类的实例。

几乎所有JavaScript函数方法都拥有一个prototype属性，这个属性是一个对象，这个对象包含一个不可枚举属性constructor，constructor属性的值是一个函数对象。



对于任意函数，F.prototype.constructor==F

上面的例子说明，构造元素的原型中存在预先定义好的constructor属性，说明对象同城继承的constructor都是指它们的构造函数，这个属性为对象提供了“类”。

**JavaScript和Java的不同之处在于，JavaScript中函数都是以“值”的形式出现的，方法和字段之间并没有太大的区别！！**

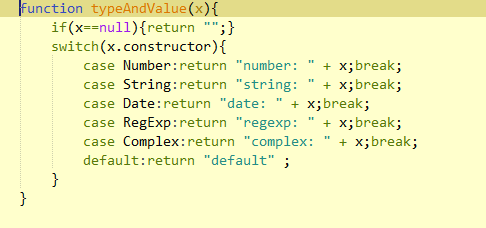
在JavaScript中模拟Java等语言的类继承，是一个分为3部的过程

1. 定义一个构造函数，初始化新对象的实例属性
2. 给构造函数的prototype对象定义实例方法
3. 给构造函数定义类字段和类属性

**JavaScript中的鸭式辩型。更加关注的是对象可以完成什么工作（它包含什么方法），而不是对象属于哪个类。**

检测任意对象的类的技术有

1. Instanceof运算符 实际上是检测对象的继承关系，而不是检测创建对象的构造函数。其缺点是：如果在不同的上下文环境中，例如多窗口和多框架情况下，不同窗口或框架之下的构造函数和其实例的instanceof运算返回的是false
2. Constructor属性



注意，constructor返回的是函数，case后面的也是函数

其缺点和instanceof运算符一样，依赖上下文条件。

1. 构造函数的名字

JavaScript中的面向对象技术