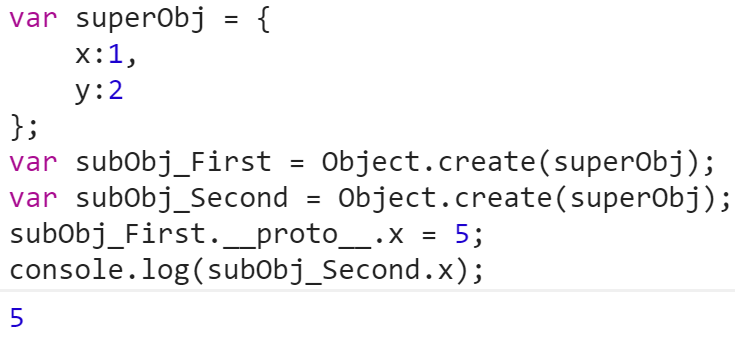
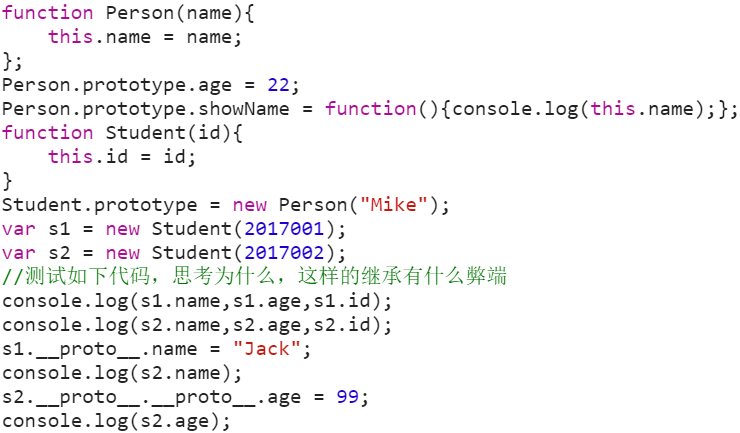
**第十七章 深入理解JS的继承方式**

**JS原型继承的方式及其优缺点**

* **JavaScript的原型继承是对象-对象的继承**- 每个对象都有一个原型对象（可动态的指定原型，来改变继承关系，最原始的原型是null）  
  - 思考并回答三种方式创建的对象的原型都是什么？  
  - 多个对象继承于一个原型时，存在原型共享（节省内存如共享方法，但也带来了共享问题）



**构造函数实现的对象-对象的原型继承的原型共享问题**

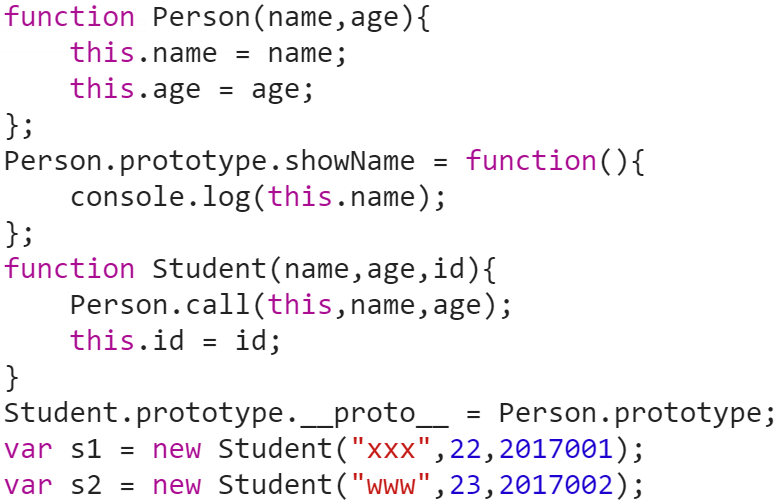
左侧的代码有什么问题

思考共享的弊端，如何给每个Studnent对象添加自有的name属性，s1.name =”Jack”，和原型的name属性什么关系，思考这样的话是否造成内存的浪费，具体参见下页图解

图片包含 文字, 地图

描述已自动生成

**模拟类-类继承的形式 一**



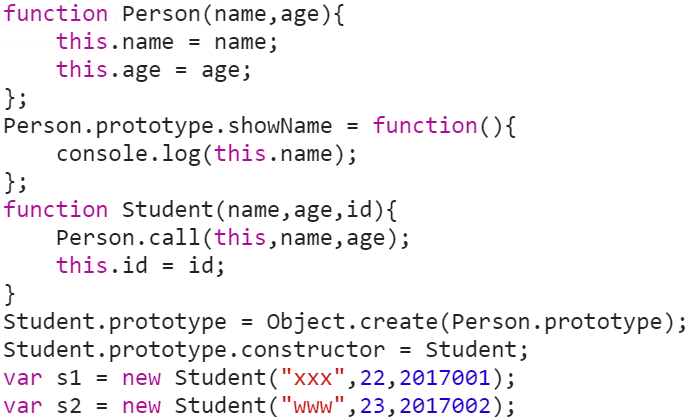
思考：name属性添加到哪个对象上了？Person.prototype、Student.prototype还是实例化的对象上？

推荐：将方法添加到对象的原型上（即构造函数的prototype上）便于共享，节省内存

图片包含 文字, 地图

描述已自动生成

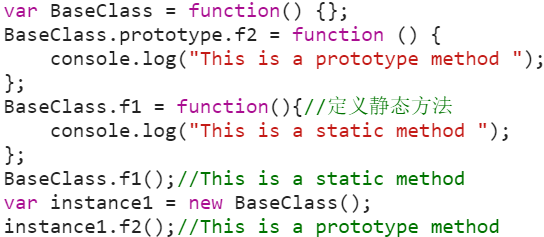
模拟类-类继承的形式 二



图片包含 文字, 地图

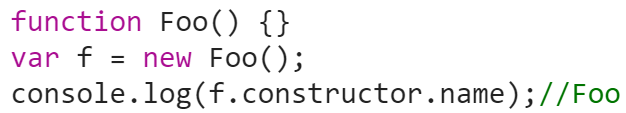
描述已自动生成

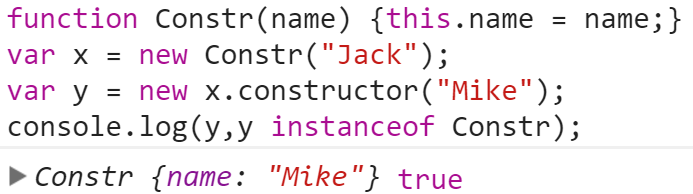
**静态方法与原型方法的区别**- 静态方法是构造器函数对象（类）的方法，原型方法是实例化对象（对象）的原型的方法  
- 使用形式有什么不同，区别在哪里？（属性共享） 直接定义在构造函数上为静态方法 在构造函数的prototype上的方法为原型方法、



**再谈对象原型的constructor属性**

* - 因为对象实例从原型中继承了constructor，所以可以通过constructor得到实例的构造函数  
  - 确定对象的构造函数名、创建相似对象、constructor可用于指定构造函数





**对象的公有属性、私有属性（回顾闭包）**

