1、对象：

属性的无序集合（名：值），常见的对象散列（hash）、散列表（hashtable）、字典（dictionary）、关联数组（associative array）

**2、常见用法**：

创建（create）、设置（set）、查找（query）、删除（delete）、检查（test）、枚举（enumerate）

3、属性特性

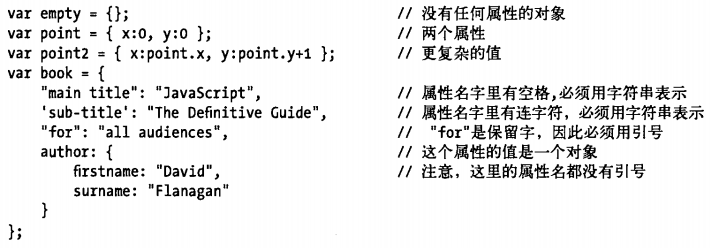
可写：是否可以设置属性的值

可枚举：可以通过for/in循环返回属性

可配置：是否可以删除或修改属性

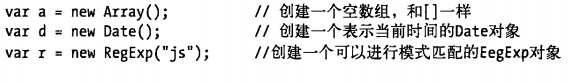
4、创建对象

大括号中使用冒号拼接{a:b}



**5、初始化对象或创建对象**

使用new关键词



6、原型

对象都是继承的原型，null除外，继承原型使用的方法（Object.prototype = function(){具体的操作方法，也是用“:”拼接}）

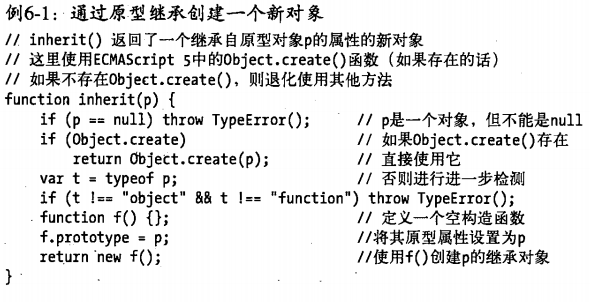
7、创建对象及继承

Object.create()



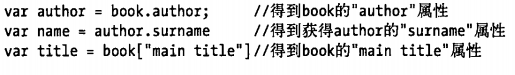






8、属性设置和查询

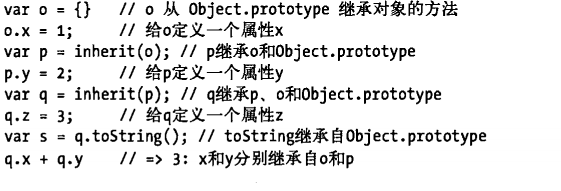
使用“.”或“[]”，点号后面不能连接关键词（for，foreach等），如果需要可以使用[‘for’]

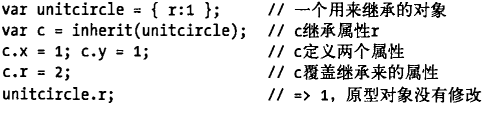




9、继承属性

原型链式指向，但和其它语言的类继承相同，不改变父级的属性

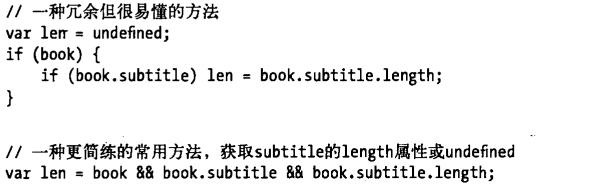




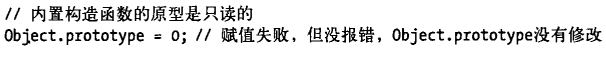
10、属性访问错误

返回o.x 返回undefined（null和undefined是无属性的，返回的状态是false）

避免访问错误的

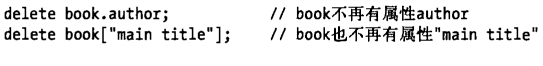


有些属性是不可复制的，如prototype

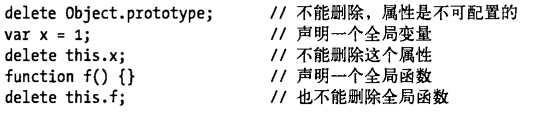


11、删除属性

delete，删除成功或没有任何副作用返回true，可以理解为永远返回true，不能删除不可配置性的返回fasle

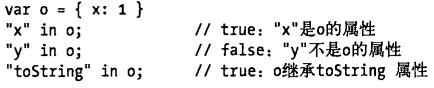


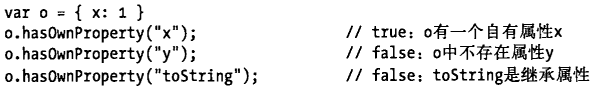


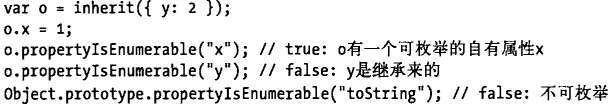


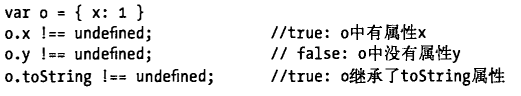
12、检查属性

in、hasOwnPreperty()、propertyIsEnumerable()【更高级的属性，可以判断出属性是否是可枚举的，自身建立的】、!==【可以区分undefined和null】

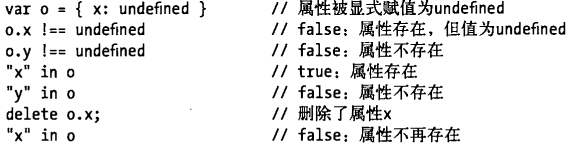


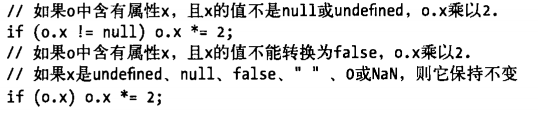






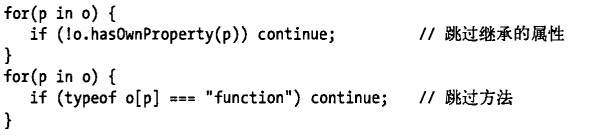
只能使用in不能使用其他的方法：in可以区分不存在的属性和存在但值为undefined的属性

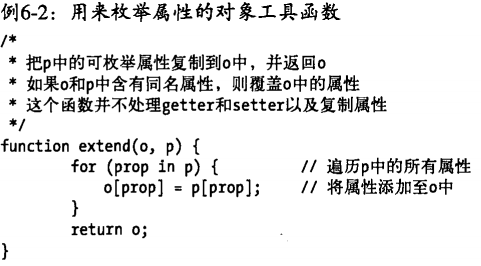


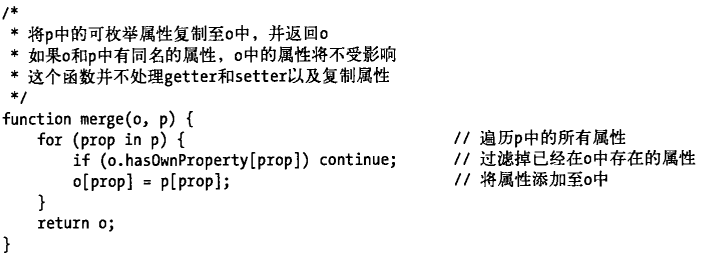


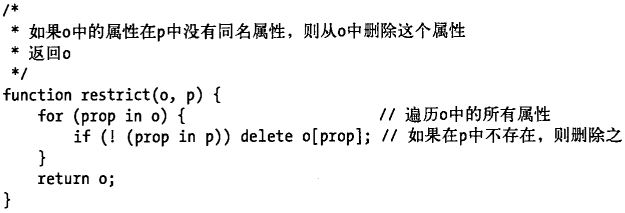
13、枚举属性

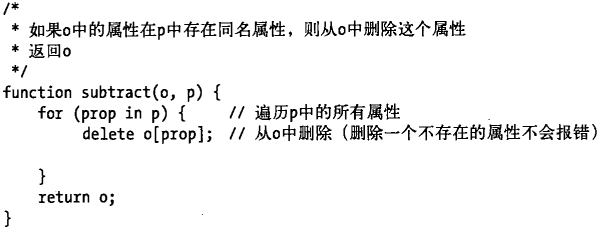
自身建立的属性

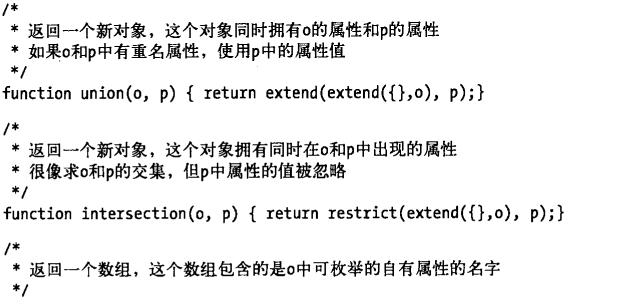


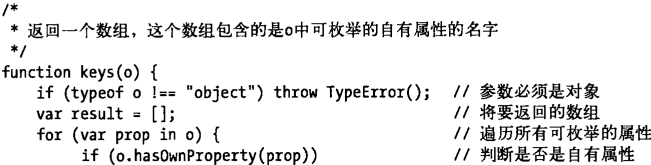


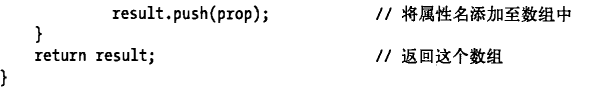






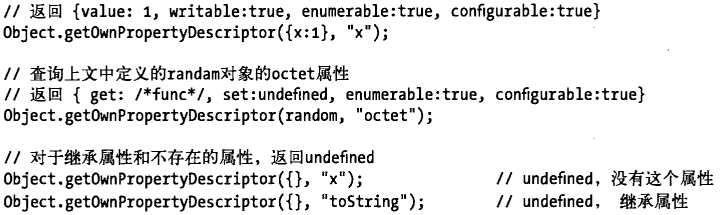






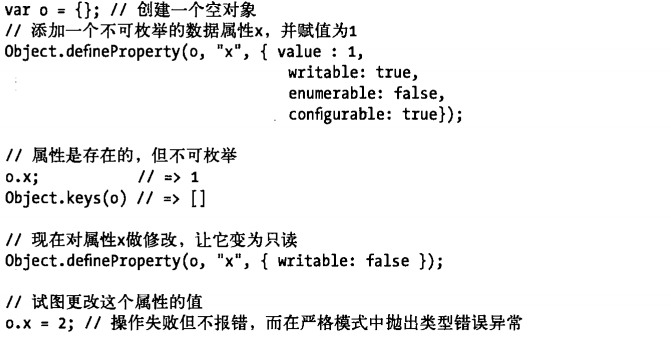
14、查看属性特性

Object.getOwnPropertyDescriptor()

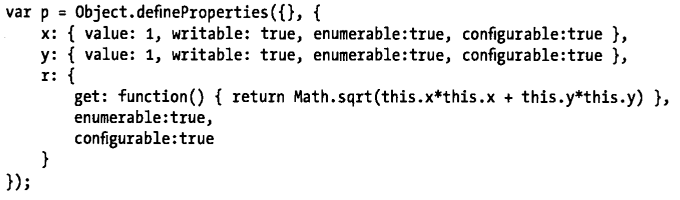


15、设置属性具有某种特性：Object.definePeoperty()，传入对应的属性设置

修改单个属性

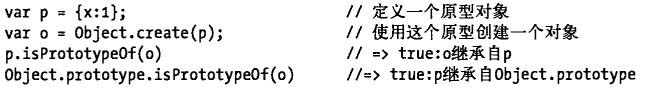


修改多个属性：第一个参数（修改的对象）、第二个参数（映射表）

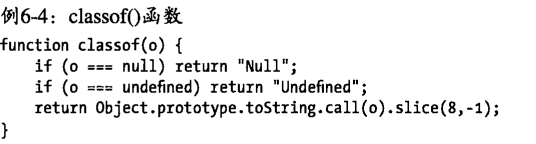


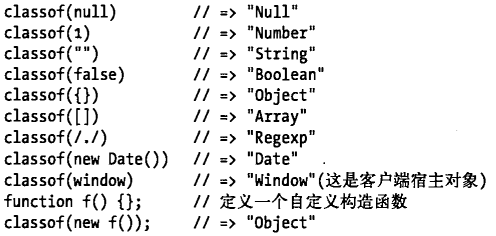
16、原型属性

上一级对象，检查继承关系isPrototypeOf()



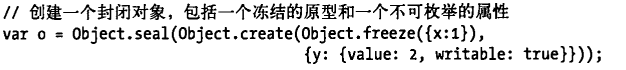
17、类属性





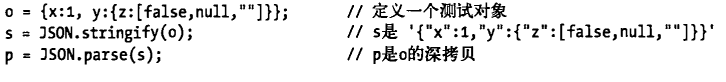
18、可扩展性

Object.esExtensible()判断是否可扩展，目的将对象锁定。Object.seal()：将对象设置为不可扩展，还可以设置为不可配置的



19、序列化对象

JSON.stringify()【序列化】，JSON.parse()【还原对象】



20、对象方法

a）toString()：返回对象的字符串，使用+拼接字符串的时候需要，几乎所有类都带这个属性



b）toLocaleString()：返回这个对象的本地化的字符串，根据本地电脑显示

c）toJSON()：JSON.stringify()方法会调用toJSON()方法

d）valueOf()：和toString()差不多