var wroxWin = window.open(...); wroxwin.opener = null;将opener属性设为null就是告诉浏览器新创建的标签页不需要与打开它的标签页通信 。要判断新窗口是不是被浏览器block,只要判断wroxWin是不是等于null;

调用setTimeout()之后，该方法会返回一个数值ID，表示超时调用。这个超时调用ID是计划执行代码的唯一标识符，可以通过它来取消超时调用。要取消尚未执行的超时调用计划，可以调用clearTimeout()方法并将相应的超时调用ID作为参数传递给它。

var timeoutId = setTimeout(function(){...},1000);

clearTimeout(timeoutId);//取消定时任务。

间歇调用：setInterval(方法,时间);setInterval(function(){},1000);

location.assign("http://....");使用assign方法并为其传递一个URL，就可以立即打开新URL并在浏览器的历史记录生成一条记录。href效果 一样. location.search 可以增加url后的?a=b&c=2查询条件。

navigator获得浏览器信息.

验证是不是数组：

function isArray(value){

   if (typeof Array.isArray === 'function') {

        return Array.isArray(value);

   } else {

      return Object.prototype.toString.call(value) === '[object Array]';

   }

}

判断一个属性是不是存在：

if ('count' in object){ ...}

配置数据：

将配置数据从代码中抽离出来的第一步是将配置数据拿到外部：

var config = {

   MSG\_INVAD\_VALUE: 'invalide value',

  ......

}

保存配置数据;

配置数据最好放在单独的文件中，以便清晰地分隔数据和应用逻辑。

当然你可以用java的property文件保存，然后通过json,jsonp返回。

JS运行时压缩，设置头信息 Content-Encoding: gzip

var person = new Person();

//给类加静态方法

Person.find= function(){...};

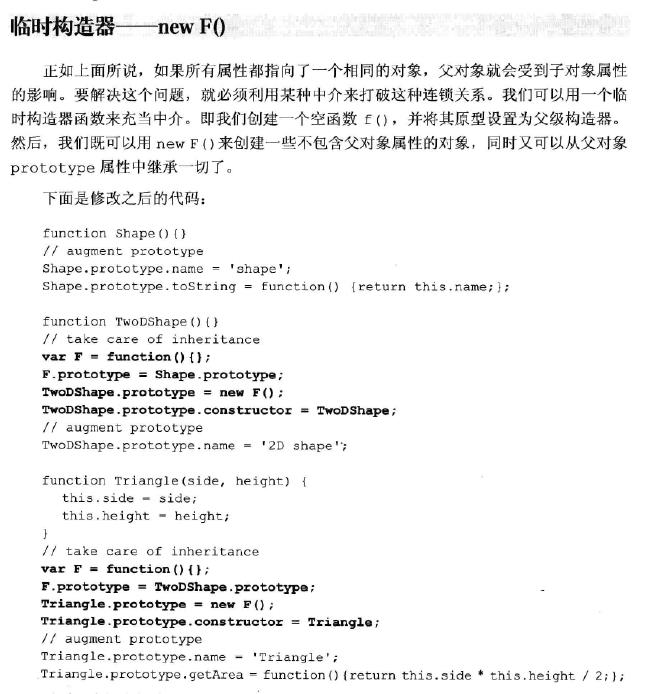
//增加实例方法

Person.prototype.fn=function(){}

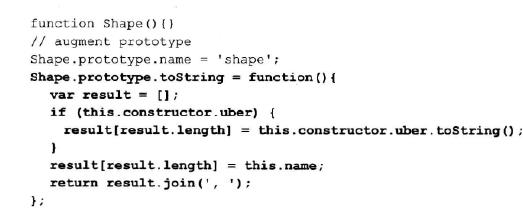
继承有原型继承：

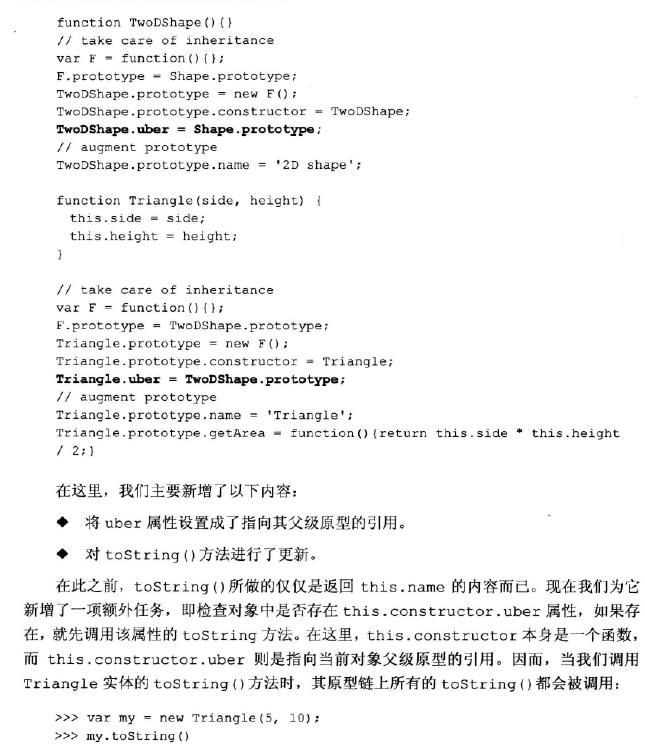
TwoDShape.prototype = Shape.prototype;

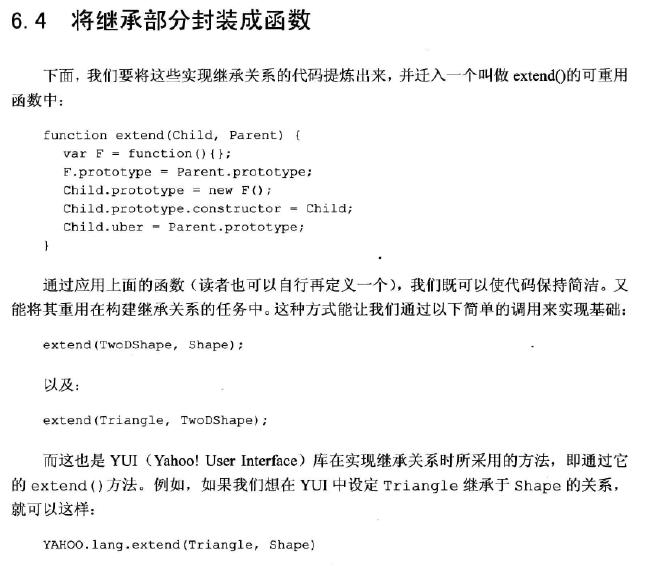
可以把一些不想让父类包括的属性发到临时构造器里.



Uber 子对象访问父对象的方式；







属性拷贝：

function extend2(Child, Parent){

    var p = Parent.prototype;

    var c = Child.prototype;

    for(var i in p){

        c[i] = p[i];

    }

   c.uber = p;

}

要小心处理引用copy;

