var arr=['a','b','c'];

　　若要删除其中的'b',有两种方法：

**1.delete方法:delete arr[1]**

　　这种方式数组长度不变,此时arr[1]变为undefined了,但是也有好处原来数组的索引也保持不变,此时要遍历数组元素可以才用

for(index in arr)  
{  
　　 document.write('arr['+index+']='+arr[index]);  
}

      这种遍历方式跳过其中undefined的元素

　　\* 该方式IE4.o以后都支持了

**2.数组对象splice方法:arr.splice(1,1);**

　　这种方式数组长度相应改变,但是原来的数组索引也相应改变

　　splice参数中第一个1,是删除的起始索引(从0算起),在此是数组第二个元素

　　第二个1,是删除元素的个数,在此只删除一个元素,即'b';

　　此时遍历数组元素可以用普通遍历数组的方式,比如for,因为删除的元素在

　　数组中并不保留

　　\* 该方法IE5.5以后才支持

　　值得一提的是splice方法在删除数组元素的同时,还可以新增入数组元素

　　比如arr.splice(1,1,'d','e'),d,e两个元素就被加入数组arr了

　　结果数组变成arr:'a','d','e','c'

　　<big>外一篇：</big>

　　JavaScript通过设置数组的length属性来截断数组是惟一一种缩短数组长度的方法.如果使用delete运算符来删除数组中元素,虽然那个元素变成未定义的,但是数组的length属性并不改变两种删除元素,数组长度也改变的方法.

　 /\*  
　 \*　方法:**Array.remove(dx)**　 \*　功能:删除数组元素.  
　 \*　参数:dx删除元素的下标.  
　 \*　返回:在原数组上修改数组  
　 \*/  
　　  
　//经常用的是通过遍历,重构数组.

[复制代码](javascript:void(0);)

Array.prototype.remove=function(dx)  
　{  
　　if(isNaN(dx)||dx>this.length){return false;}  
　　for(var i=0,n=0;i<this.length;i++)  
　　{  
　　　　if(this[i]!=this[dx])  
　　　　{  
　　　　　　this[n++]=this[i]  
　　　　}  
　　}  
　　this.length-=1  
　}  
　a = ['1','2','3','4','5'];  
　alert("elements: "+a+"nLength: "+a.length);  
　a.remove(0); //删除下标为0的元素  
　alert("elements: "+a+"nLength: "+a.length);

[复制代码](javascript:void(0);)

[复制代码](javascript:void(0);)

/\*  
　 \*　方法:Array.baoremove(dx)  
　 \*　功能:删除数组元素.  
　 \*　参数:dx删除元素的下标.  
　 \*　返回:在原数组上修改数组.  
　 \*/  
　　  
　//我们也可以用splice来实现.  
　　  
　Array.prototype.baoremove = function(dx)  
　{  
　　if(isNaN(dx)||dx>this.length){return false;}  
　　this.splice(dx,1);  
　}  
　b = ['1','2','3','4','5'];  
　alert("elements: "+b+"nLength: "+b.length);  
　b.baoremove(1); //删除下标为1的元素  
　alert("elements: "+b+"nLength: "+b.length);

[复制代码](javascript:void(0);)

     我们知道，在IE5或更低的版本中，JavaScript的Array（数组）对象并未提供现成的删除数组元素的方法。在IE5.5+的版本中，虽然有splice方法，但是并不是删除某一项（或几项），而仅仅是将某一项（或几项）的值清除，也就是说该项仍然存在，数组的长度并没有改变。

　　事实上，我们可以自己为数组增加一个删除方法（注意，这里指的是将数组的某一项真正的从数组成员中移除）。或许你会想到用循环来为数组重新赋值，这样做当然可以，但效率很低。

　　下面我们介绍利用Array对象的两个方法slice、concat来自定义删除数组的方法。

　　具体代码如下，请注意里面的注释。

--------------------------------------------------------------

[复制代码](javascript:void(0);)

Array.prototype.del=function(n) {　//n表示第几项，从0开始算起。  
//prototype为对象原型，注意这里为对象增加自定义方法的方法。  
　if(n<0)　//如果n<0，则不进行任何操作。  
　　return this;  
　else  
　　return this.slice(0,n).concat(this.slice(n+1,this.length));  
　　/\*  
　　　concat方法：返回一个新数组，这个新数组是由两个或更多数组组合而成的。  
　　　　　　　　　这里就是返回this.slice(0,n)/this.slice(n+1,this.length)  
　　 　　　　　　组成的新数组，这中间，刚好少了第n项。  
　　　slice方法： 返回一个数组的一段，两个参数，分别指定开始和结束的位置。  
　　\*/  
}  
//我们来试一试这个自己增加的方法  
var test=new Array(0,1,2,3,4,5);  
test=test.del(3);　//从0算起，这里也就是删除第4项。  
alert(test);

[复制代码](javascript:void(0);)