学习javascript的时候，我曾经一度搞不清楚”数组”(array)和”对象”(object)的根本区别在哪里，两者都可以用来表示数据的集合。

比如有一个数组

a=[1,2,3,4]，还有一个对象

a={0:1,1:2,2:3,3:4}，然后你运行alert(a[1])，两种情况下的运行结果是相同的！这就是说，数据集合既可以用数组表示，也可以用对象表示，那么我到底该用哪一种呢？

我后来才知道，**数组表示有序数据的集合，而对象表示无序数据的集合**。如果数据的顺序很重要，就用数组，否则就用对象。

当然，数组和对象的另一个区别是，数组的数据没有”名称”（name），对象的数据有”名称”（name）。

但是问题是，很多编程语言中，都有一种叫做”**关联数组**”（associative array）的东西。这种数组中的数据是有名称的。

比如在javascript中，可以这样定义一个对象：

var a={“城市”:”北京”,”面积”:16800,”人口”:1600};

但是，也可以定义成一个关联数组：

a["城市"]=”北京”;

a["面积"]=16800;

a["人口"]=1600;

这起初也加剧了我对数组和对象的混淆，后来才明白，在Javascript语言中，**关联数组就是对象**，对象就是关联数组。

比如运行下面这段javascript：

var a=[1,2,3,4];

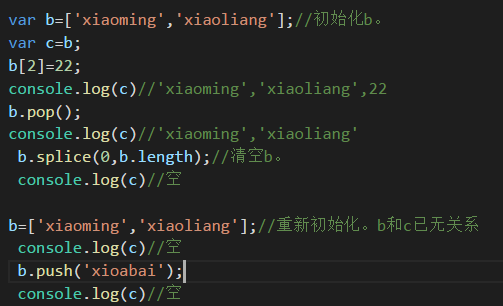
a['foo']=’Hello World’;

alert(a.length);

最后的结果是4，也就是说，数组a的元素个数是4个。

上面对其**引用类型**阐述的不够确切。

下面继续阐述：



解释：例子中首先声明数组 b 。然后 c=b。 因为数组属于引用类型。故两者的关系是：



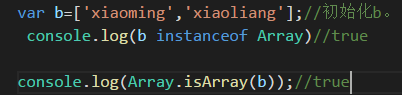
（图是借的，理会意思就好）。

因此，对数组b的操作会毫无保留的影响到c。

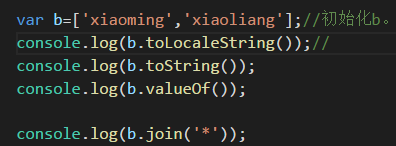
但是如果对b重新定义。那么c还是指向原来的那个b所在的位置。两者就没有关系了。

**数组的基本用法总结。**

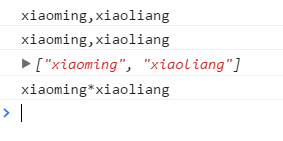
1. **检测数组（instanceof, isArray ）**



**2, 转换方法。**

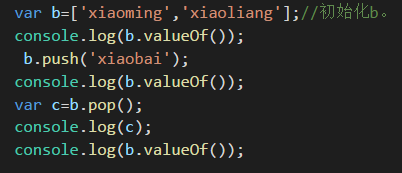


在上面例子中返回的情况：



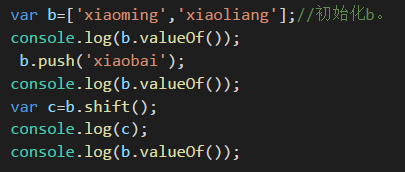
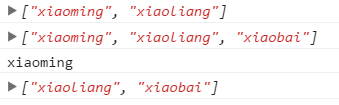
前两个返回以逗号间隔的字符串。第三个返回数组。第四个返回可以自定义间隔符号（比如\*）的字符串。

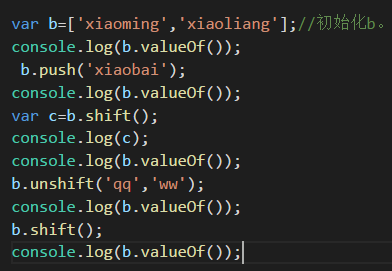
**3，当栈用：( push,pop )和c++特别像**。

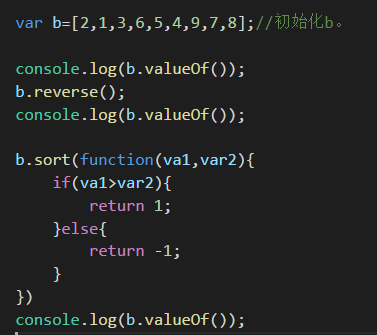
**4，当队列用：（push, shift）还有一个unshift.**

他是在前面插入任意个元素。



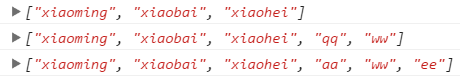
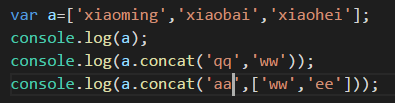
**5，重排序（reverse 和sort）**：



Reverse是数组逆转，没什么好说的。

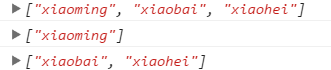
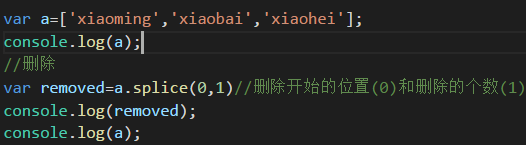
主要是sort. sort(function(va1,va2){})接受的函数有两个参数。排序的原则是：如果要逆序（前一个和后一个交换位置）则返回 1.

**6. 操作方法（concat(拼接)，slice(最强大)）：**

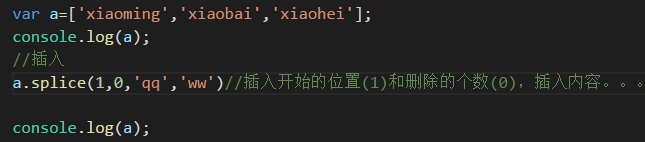


Slice.

删除



插入：

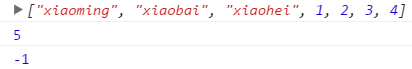
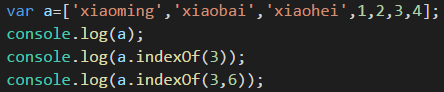


替换。和插入一样。

**7，位置方法：**

indexOf(参数1，参数2)。参数1表示查找的项。第二个参数可选，表示查找开始的位置。其是从数组头开始查。

lastIndexOf()其他的一样，是从数组尾开始查。



**8，迭代方法：**

every, some , filter, forEach, map.

every() 对数组中的每一项，运行给定函数，如果函数对每一项都返回true，则返回true。

some() 对数组中的每一项运行给定函数，如果该函数对任一项返回true，则返回true。

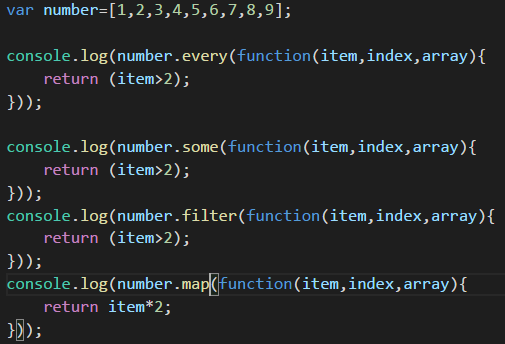
filter() 对数组中每一项运行给定函数，返回该函数返回true值的数组。

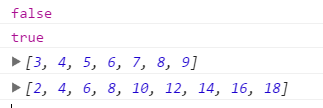
forEach() 对数组中的每一项运行给定函数，该方法不返回值。

map() 对数组中的每一项运行给定函数，返回每次函数调用的结果组成的数组。

以上方法都不会改变数组中包含的值。

给定的函数接受三个参数。数组项的值，该项在数组中的位置，数组本身。



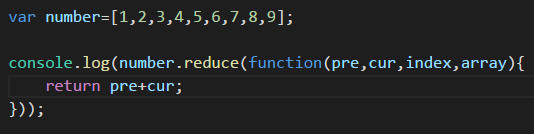


forEach()，不甚清楚，这里不做讨论。

**9, 归并方法**

reduce()和reduceRight().

这两个方法会迭代所有项，然后构建一个最终返回值。reduce()是从第一项开始，逐个遍历到最后。而reduceRight()是从最后一项开始。

//45