**[JS中数组的操作](http://www.cnblogs.com/tianguook/archive/2010/09/30/1839648.html)**

**1、数组的创建**

var arrayObj = new Array();　//创建一个数组

var arrayObj = new Array([size]);　//创建一个数组并指定长度，注意不是上限，是长度

var arrayObj = new Array([element0[, element1[, ...[, elementN]]]]);　创建一个数组并赋值

    要说明的是，虽然第二种方法创建数组指定了长度，但实际上所有情况下数组都是变长的，也就是说即使指定了长度为5，仍然可以将元素存储在规定长度以外的，注意：这时长度会随之改变。

**2、数组的元素的访问**

var testGetArrValue=arrayObj[1]; //获取数组的元素值

arrayObj[1]= "这是新值"; //给数组元素赋予新的值

**3、数组元素的添加**

arrayObj. push([item1 [item2 [. . . [itemN ]]]]);// 将一个或多个新元素添加到数组结尾，并返回数组新长度

arrayObj.unshift([item1 [item2 [. . . [itemN ]]]]);// 将一个或多个新元素添加到数组开始，数组中的元素自动后移，返回数组新长度

arrayObj.splice(insertPos,0,[item1[, item2[, . . . [,itemN]]]]);//将一个或多个新元素插入到数组的指定位置，插入位置的元素自动后移，返回""。

**4、数组元素的删除**

arrayObj.pop(); //移除最后一个元素并返回该元素值

arrayObj.shift(); //移除最前一个元素并返回该元素值，数组中元素自动前移

arrayObj.splice(deletePos,deleteCount); //删除从指定位置deletePos开始的指定数量deleteCount的元素，数组形式返回所移除的元素

**5、数组的截取和合并**

arrayObj.slice(start, [end]); //以数组的形式返回数组的一部分，注意不包括 end 对应的元素，如果省略 end 将复制 start 之后的所有元素

arrayObj.concat([item1[, item2[, . . . [,itemN]]]]); //将多个数组（也可以是字符串，或者是数组和字符串的混合）连接为一个数组，返回连接好的新的数组

**6、数组的拷贝**

arrayObj.slice(0); //返回数组的拷贝数组，注意是一个新的数组，不是指向

arrayObj.concat(); //返回数组的拷贝数组，注意是一个新的数组，不是指向

**7、数组元素的排序**

arrayObj.reverse(); //反转元素（最前的排到最后、最后的排到最前），返回数组地址

arrayObj.sort(); //对数组元素排序，返回数组地址

**8、数组元素的字符串化**

arrayObj.join(separator); //返回字符串，这个字符串将数组的每一个元素值连接在一起，中间用 separator 隔开。

toLocaleString 、toString 、valueOf：可以看作是join的特殊用法，不常用

**二、数组对象的3个属性**

**1、length 属性**

    Length属性表示数组的长度，即其中元素的个数。因为数组的索引总是由0开始，所以一个数组的上下限分别是：0和length-1。和其他大多数语言不同的是，JavaScript数组的length属性是可变的，这一点需要特别注意。当length属性被设置得更大时，整个数组的状态事实上不会发生变化，仅仅是length属性变大；当length属性被设置得比原来小时，则原先数组中索引大于或等于length的元素的值全部被丢失。下面是演示改变length属性的例子：

var arr=[12,23,5,3,25,98,76,54,56,76];

//定义了一个包含10个数字的数组

alert(arr.length); //显示数组的长度10

arr.length=12; //增大数组的长度

alert(arr.length); //显示数组的长度已经变为12

alert(arr[8]); //显示第9个元素的值，为56

arr.length=5; //将数组的长度减少到5，索引等于或超过5的元素被丢弃

alert(arr[8]); //显示第9个元素已经变为"undefined"

arr.length=10; //将数组长度恢复为10

alert(arr[8]); //虽然长度被恢复为10，但第9个元素却无法收回，显示"undefined"

    由上面的代码我们可以清楚的看到length属性的性质。但length对象不仅可以显式的设置，它也有可能被隐式修改。JavaScript中可以使用一个未声明过的变量，同样，也可以使用一个未定义的数组元素（指索引超过或等于length的元素），这时，length属性的值将被设置为所使用元素索引的值加1。例如下面的代码：

var arr=[12,23,5,3,25,98,76,54,56,76];

alert(arr.length);

arr[15]=34;

alert(arr.length);

    代码中同样是先定义了一个包含10个数字的数组，通过alert语句可以看出其长度为10。随后使用了索引为15的元素，将其赋值为15，即arr[15]=34，这时再用alert语句输出数组的长度，得到的是16。无论如何，对于习惯于强类型编程的开发人员来说，这是一个很令人惊讶的特性。事实上，使用new Array()形式创建的数组，其初始长度就是为0，正是对其中未定义元素的操作，才使数组的长度发生变化。

    由上面的介绍可以看到，length属性是如此的神奇，利用它可以方便的增加或者减少数组的容量。因此对length属性的深入了解，有助于在开发过程中灵活运用。

**2、prototype 属性**

返回对象类型原型的引用。prototype 属性是 object 共有的。

objectName.prototype

objectName 参数是object对象的名称。

说明：用 prototype 属性提供对象的类的一组基本功能。 对象的新实例“继承”赋予该对象原型的操作。

    对于数组对象，以以下例子说明prototype 属性的用途。

    给数组对象添加返回数组中最大元素值的方法。要完成这一点，声明一个函数，将它加入 Array.prototype， 并使用它。

function array\_max( )

{

   var i, max = this[0];

   for (i = 1; i < this.length; i++)

   {

   if (max < this[i])

   max = this[i];

   }

   return max;

}

Array.prototype.max = array\_max;

var x = new Array(1, 2, 3, 4, 5, 6);

var y = x.max( );

该代码执行后，y 保存数组 x 中的最大值，或说 6。

**3、constructor 属性**

表示创建对象的函数。

object.constructor //object是对象或函数的名称。

说明：constructor 属性是所有具有 prototype 的对象的成员。它们包括除 Global 和 Math 对象以外的所有 JScript 固有对象。constructor 属性保存了对构造特定对象实例的函数的引用。

例如：

x = new String("Hi");

if (x.constructor == String) // 进行处理（条件为真）。

或

function MyFunc {

// 函数体。

}

y = new MyFunc;

if (y.constructor == MyFunc) // 进行处理（条件为真）。

对于数组来说：

y = new Array();