数组 Array

创建Array对象

使用Array构造函数

```
var colors = new Array();

var colors = new Array(20);

var colors = new Array("red","blue","green");

省略new操作符：var colors =Array();
```

数组字面量表示法

```
var colors = ["red","blue","green"];
```

属性和方法

属性

length 设置或返回数组中元素的个数

转换字符串方法

toString() 把数组转换为字符串，并返回结果。元素之间用逗号分隔。

toLocaleString() 把数组转换为本地字符串，并返回结果。元素之间用逗号分隔。

join(分隔符(可选)) 把数组的所有元素放入一个字符串。元素通过指定的分隔符进行分隔。如果省略该参数，则使用逗号作为分隔符。如果要让字符串没有分隔符，则参数为一个空字符串("") 。

重排序方法

reverse() 颠倒数组中元素的顺序。

sort() 对数组的元素进行排序

- sort() 按升序排列数组项，比较的是字符串（即使每一项都是数值）。
- sort(function(a,b)) 参数：比较函数，该函数接收两个参数 a 和 b，其返回值如下：

若 a 应该出现在 b 之前，则返回一个负数。

若 a 等于 b，则返回 0。

若 a 应该出现在 b 之后，则返回一个正数。

这些方法会改变原来的数组，而不会创建新的数组。

例：升序排列

```
function sortNumber(a,b)
{
return a - b
}
```

连接方法

concat(arrayX,arrayX,.....,arrayX) 连接两个或更多的数组，并返回被连接数组的一个副本。调用后原数组不变。

提取元素方法

slice(start,end(可选)) 返回一个新数组，包含从 start 到 end （不包括该元素）的元素。调用后原数组不变。

参数：

start　如果是负数，那么它规定从数组尾部开始算起的位置。也就是说，-1 指最后一个元素，-2 指倒数第二个元素，以此类推。
end 如果没有指定该参数，那么切分的数组包含从 start 到数组结束的所有元素。如果这个参数是负数，那么它规定的是从数组尾部开始算起的元素。

插入删除替换方法

splice(起始位置，要删除的项数，要插入的项(可选)) 删除、插入、替换元素。

- splice(要删除的第一项位置，项数) 删除
- splice(起始位置，0，要插入的项) 插入
- splice(起始位置，要删除的项数，要插入的项)) 替换

push(newelement1,newelement2,....,newelementX) 向数组的末尾添加一个或多个元素，并返回新的长度。
pop() 删除并返回数组的末尾元素
shift() 删除并返回数组的首个元素
unshift(newelement1,newelement2,....,newelementX) 向数组的开头添加一个或多个元素，并返回新的长度
调用后原数组被修改。

查找方法

indexOf(要查找的项，查找的起点位置(可选)) 从数组开头向后查找，返回要查找的项在数组中首次出现的位置。如果没有找到，则返回 -1。
lastIndexOf(要查找的项，查找的起点位置(可选)) 从数组末尾向前查找，返回要查找的项在数组中首次出现的位置。如果没有找到，则返回 -1。

迭代方法

map(function(item,index,array),作用域(可选)) 对数组中的每一项运行函数，返回每次函数调用的结果组成的数组。
filter(function(item,index,array),作用域(可选)) 对数组中的每一项运行函数，返回该函数会返回true的项组成的数组。
every(function(item,index,array),作用域(可选)) 对数组中的每一项运行函数，如果该函数对每一项都返回true，则返回true。
some(function(item,index,array),作用域(可选)) 对数组中的每一项运行函数，如果该函数对任意一项都返回true，则返回true。
forEach(function(item,index,array),作用域(可选)) 对数组中的每一项运行函数，没有返回值。本质上与使用for迭代数组一样。
参数：
item 数组项的值
index 该项在数组中的位置
array 数组本身

归并方法

reduce(function(prev,cur,index,array)，作为归并基础的初始值(可选)) 迭代数组的所有项，返回结果。function()返回的任何值都会作为prev传递给下一项。第一次迭代发生在array的第二项上，因此prev是第一项，cur是第二项。