**Array方法总结**

# 检测数组

Array.isArray()

# 转换方法

1. Array.toString()方法，返回数组的字符串表示，每个值的字符串
2. 表示拼接成了一个字符串，中间以逗号分隔。
3. Array.join()方法，可以使用不同的分隔符来构建这个字符串。join()方
4. 法只接收一个参数，即用作分隔符的字符串，然后返回包含所有数组项的字符串。如果不给 join()方法传入任何值，或者给它传入 undefined，则使用逗号作为分隔符。

# 栈方法

push()方法可以接收任意数量的参数，把它们逐个添加到数组末尾，并返回修改后数组的长度。而

pop()方法则从数组末尾移除最后一项，减少数组的 length 值，然后返回移除的项。

# 队列方法

shift()，它能够移除数组中的第一个项并返回该项,unshift()在数组前端添加任意个项并返回新数组的长度

# 重排序方法

sort()方法按升序排列数组项——即最小的值位于最前面，最大的值排在最后面。

为了实现排序，sort()方法会调用每个数组项的 toString()转型方法，然后比较得到的字符串，以

确定如何排序。即使数组中的每一项都是数值，sort()方法比较的也是字符串

var values = [0, 1, 5, 10, 15];

values.sort();

alert(values); //0,1,10,15,5

这种排序方式在很多情况下都不是最佳方案。因此 sort()方法可以接收一个比较函数作为参

数，以便我们指定哪个值位于哪个值的前面。

比较函数接收两个参数，如果第一个参数应该位于第二个之前则返回一个负数，如果两个参数相等

则返回 0，如果第一个参数应该位于第二个之后则返回一个正数。

function compare(value1, value2){

return value2 - value1;

}

var values = [0, 1, 5, 10, 15];

values.sort(compare);

# 迭代方法

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

 filter()：对数组中的每一项运行给定函数，返回该函数会返回 true 的项组成的数组。

 forEach()：对数组中的每一项运行给定函数。这个方法没有返回值。

 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。

 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。

以上方法都不会修改数组中的包含的值。

# 归并方法

reduce()和 reduceRight()的函数接收 4 个参数：前一个值、当前值、项的索引和数组对象。这

个函数返回的任何值都会作为第一个参数自动传给下一项。第一次迭代发生在数组的第二项上，因此第

一个参数是数组的第一项，第二个参数就是数组的第二项。

使用 reduce()方法可以执行求数组中所有值之和的操作，比如：

var values = [1,2,3,4,5];

var sum = values.reduce(function(prev, cur, index, array){

return prev + cur;

});

alert(sum); //15