1. 创建数组
2. Array构造函数

var arr=new Array();

var arr=Array();

var arr=new Array(5);

var arr=new Array("item1");

1. 数组字面量

var arr=[];

var arr=["item1","item2"];

在使用数组字面量法时不会调用Array构造函数(firefox3及以前除外)

1. 数组元素的添加删除
2. 栈方法

arr.push(2);//从队尾添加，返回新长度

arr.pop();//从队尾删除，返回删除的元素

1. 队列方法

arr.unshift(2);//从队头添加，返回新长度

arr.shift();//从队头删除，返回删除的元素

1. splice()

arr.splice(1);//删除从index=1到末尾所有元素

arr.splice(2,1);//删除从索引2开始的n个元素，返回被删除的元素组成的数组

arr.splice(2,0,"item1","item2");//从索引2的元素前面添加2个新元素，返回空数组

arr.splice(2,1,"a","b");//从index=2开始，删除1个，添加2个

1. 数组类型检测
2. ECMAScript5

Array.isArray(arr);//true/false

支持的浏览器：IE9+ Firefox4+ Safari5+ Opera10.5+ Chrome

1. ECMAScript5以前

arr instanceof Array

instanceof操作符的问题：它假定只有一个全局环境。如果网页中包含多个框架，那实际上就存在2个以上不同的全局执行环境，从而存在2个以上不同版本的Array构造函数。如果从一个框架向另一个框架传入数组，那么传入的数组与在第二个框架中原生创建的数组分别有不同的构造函数。

解决方案：检测对象的类属性

var isArray=Function.isArray||function(o){

return typeof o === "object" && Object.prototype.toString.call(o) === "[object Array]";

}

1. 数组与字符串转换
2. 数组转换为字符串

arr.toString();

arr.toLocaleString();

arr.valueOf();

arr.join();

arr.join("-");

1. 字符串转换为数组

var str="abc";

str.split();//["abc"]

str.split("");//["a","b","c"];

1. 方法
2. 重排序方法
   1. arr.sort()
   2. arr.reverse()
3. 操作方法
   1. arr.concat()
   2. arr.slice()
   3. arr.splice()
4. 位置方法

arr.indexOf()

arr.lastIndexOf()

1. 迭代方法

arr.every()

arr.some()

arr.filter()

arr.forEach()

arr.map()

1. 归并方法

arr.reduce()

arr.reduceRight()