## JavaScript常用数组操作方法

数组是用于储存多个相同类型数据的集合，平时在数据的处理中用到最多，JavaScript 中常用的操作方法

**1、concat()**

concat() 方法用于连接两个或多个数组。该方法不会改变现有的数组，仅会返回被连接数组的一个副本。

var arr1 = [1,2,3];

var arr2 = [4,5];

var arr3 = arr1.concat(arr2);

console.log(arr1); //[1, 2, 3]

console.log(arr3); //[1, 2, 3, 4, 5]

**2、join()**

join() 方法用于把数组中的所有元素放入一个字符串。元素是通过指定的分隔符进行分隔的，默认使用','号分割，不改变原数组。

var arr = [2,3,4];

console.log(arr.join());  //2,3,4

console.log(arr.join('.')); //[2.3.4]

**3、push()**

push() 方法可向数组的末尾添加一个或多个元素，并返回新的长度。末尾添加，返回的是长度，会改变原数组。

var a1 = [2,3,4];

var a2 = a1.push(5);

console.log(a1);  //[2,3,4,5]

console.log(a2);  //4

**4、pop()**

pop() 方法用于删除并返回数组的最后一个元素。返回最后一个元素，会改变原数组。

var arr = [2,3,4];

console.log(arr.pop()); //4

console.log(arr);  //[2,3]

**5、shift()**

shift() 方法用于把数组的第一个元素从其中删除，并返回第一个元素的值。返回第一个元素，改变原数组。

var arr = [2,3,4];

console.log(arr.shift()); //2

console.log(arr);  //[3,4]

**6、unshift()**

unshift() 方法可向数组的开头添加一个或更多元素，并返回新的长度。返回新长度，改变原数组。

var arr = [2,3,4,5];

console.log(arr.unshift(10,100)); //6

console.log(arr); // [10, 100, 2, 3, 4, 5]

**7、slice()**

返回一个新的数组，包含从 start 到 end （不包括该元素）的 arrayObject 中的元素。返回选定的元素，该方法不会修改原数组。

var arr = [2,3,4,5];

console.log(arr.slice(1,3));  //[3,4]

console.log(arr);  //[2,3,4,5]

**8、splice()**

splice() 方法可删除从 index 处开始的零个或多个元素，并且用参数列表中声明的一个或多个值来替换那些被删除的元素。如果从 arrayObject 中删除了元素，则返回的是含有被删除的元素的数组。splice() 方法会直接对数组进行修改。

语法：array.splice(*index*,*howmany*,*item1*,.....,*itemX*)

Index必需。规定从何处添加/删除元素

howmany必需。规定应该删除多少元素

item1可选。要添加到数组的新元素

var a = [5,6,7,8];

console.log(a.splice(1,0,9)); //[]

console.log(a);  // [5, 9, 6, 7, 8]

var b = [5,6,7,8];

console.log(b.splice(1,2,3));  //[6, 7]

console.log(b); //[5, 3, 8]

**9、substring() 和 substr()**

**相同点：**如果只是写一个参数，两者的作用都一样：都是是截取字符串从当前下标以后直到字符串最后的字符串片段。  
substr(startIndex);  
substring(startIndex);

var str = '123456789';

console.log(str.substr(2));    //  "3456789"

console.log(str.substring(2)) ;//  "3456789"

**不同点：**第二个参数  
substr（startIndex,lenth）： 第二个参数是截取字符串的长度（从起始点截取某个长度的字符串）；  
substring（startIndex, endIndex）： 第二个参数是截取字符串最终的下标 （截取2个位置之间的字符串,‘含头不含尾’）。

console.log("123456789".substr(2,5));    //  "34567"

console.log("123456789".substring(2,5)) ;//  "345"

**10、sort 排序**

按照 Unicode code 位置排序，默认升序

var a = ['red', 'blue', 'green'];

a.sort(); // ["blue", "green", "red"]

var num = [1, 10, 21, 2];

num.sort(); // [1, 10, 2, 21]

**11、reverse()**

reverse() 方法用于颠倒数组中元素的顺序。返回的是颠倒后的数组，会改变原数组。

var arr = [2,3,4];

console.log(arr.reverse()); //[4, 3, 2]

console.log(arr);  //[4, 3, 2]

**12、indexOf 和 lastIndexOf**

都接受两个参数：查找的值、查找起始位置  
不存在，返回 -1 ；存在，返回位置。indexOf 是从前往后查找， lastIndexOf 是从后往前查找。  
**indexOf**

var a = [2, 9, 9];

a.indexOf(2); // 0

a.indexOf(7); // -1

if (a.indexOf(7) === -1) {}

**lastIndexOf**

var numbers = [2, 5, 9, 2];

numbers.lastIndexOf(2);     // 3

numbers.lastIndexOf(7);     // -1

numbers.lastIndexOf(2, 3);  // 3

numbers.lastIndexOf(2, 2);  // 0

numbers.lastIndexOf(2, -2); // 0

numbers.lastIndexOf(2, -1); // 3

**13、every**

对数组的每一项都运行给定的函数，每一项都返回 ture,则返回 true

function fun(element) {

return element < 10;

}

[2, 5, 8, 3, 4].every(fun); // true

function fun(element) {

return element < 10;

}

[2, 5, 11, 3, 4].every(fun); // false

**14、filter**

对数组的每一项都运行给定的函数，返回 结果为 ture 的项组成的数组

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

var a = arr.filter(function(item){

return item != 3;

});

console.log(a) //[1,2,4,5]

**15、forEach 数组遍历**

var arr= [1, 2, 3];

var arr2 = [];

arr.forEach(function(item){

arr2.push(item+1)

});

console.log(arr2) // [2, 3, 4]