

数组

数组API

- 查找元素的位置

```
var i = arr.indexOf(元素值, starti);
```

- 判断数组是否由规定的内容组成

- 判断是否所有元素都符合要求 **every**

```
var bool = arr.every(function(ele, i, arr){  
    //ele:当前元素值  
    //i:当前位置  
    //arr:当前数组对象  
    return 根据ele, i, arr判断当前元素是否符合要求;  
});
```

- 判断是否包含符合要求的元素 **some**

```
var bool = arr.some(function(ele, i, arr){  
    return 判断条件;  
});
```

- 遍历

- 对原数组中每个元素执行相同的操作 **forEach**

```
arr.forEach(function(ele, i, arr){  
    //对当前元素执行相同的操作  
});
```

- 依次取出原数组中每个元素值,执行相同操作后,放入新数组中返回 **map**

```
var evens = arr.map(function(ele, i, arr){  
    //操作
```

```
});
```

- 只关心内容,不关心位置和顺序 **for of**

```
for(var ele of arr){  
    //ele:arr的值  
}
```

* 总结:只是使用最佳选择

1. 遍历索引数组:forEach , for
2. 遍历数组对象:for
3. 遍历对象和关联数组:for in
4. 保护原数组,并生成新数组:map ---只有索引数组
5. 遍历对象和关联数组 for of

• 过滤和汇总

- 过滤:复制出原数组中符合条件的元素,组成新的子数组,原数组保持不变 **filter**

```
var subarr = arr.filter(function(ele,i,arr){  
    return 判断条件;  
});
```

- 汇总:对数组中的所有数组进行统计和分析,得出最终结论

```
var result = arr.reduce(function(prev,ele,i,arr){  
    //prev是截止目前的临时汇总值  
    return prev;  
});
```

数组常见API

- concat()

连接两个或更多的数组,并返回结果

```
var a = [1,2,3];  
document.write(a.concat(4,5));  
输出: 1,2,3,4,5
```

语法: `arrayObject.concat(arrayX,arrayX,...,arrayX);`

- `join()`

//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

```
var arr = new Array();
arr[0] = "George";
arr[1] = "John";
arr[2] = "Thomas";
document.write(arr.join())
//输出结果:George,John,Thomas
```

/*语法: `arrayObject.join(separator);`
separator 可选,指定分隔符,默认逗号分割*/

- `pop()`

//删除并返回数组的最后一个元素

```
var arr = new Array();
arr[0] = "George";
arr[1] = "John";
arr[2] = "Thomas";
document.write(arr); //George, John, Thomas
document.write(arr.pop()); //Thomas
document.write(arr); //George, John
```

/*语法: `arrayObject.pop()`
返回值:数组的最后一个元素
*/

- `push()`

//向数组的末尾添加一个或多个元素,并返回新的长度

```
var arr = new Array(3)
arr[0] = "George"
arr[1] = "John"
arr[2] = "Thomas"

document.write(arr); //George, John, Thomas
document.write(arr.push("James")); //4
document.write(arr); //George, John, Thomas, James
```

```
/*语法:arrayObject.push(newelement1,newelement2,...,newelementn);
newelement1是必须的,其他可选
*/
```

- reverse()

```
//颠倒数组中元素的顺序
var arr = new Array(3)
arr[0] = "George"
arr[1] = "John"
arr[2] = "Thomas"

document.write(arr);//George, John, Thomas
document.write(arr.reverse());//Thomas, John, George

/*语法:arrayObject.reverse()
该方法会改变原来的数组,而不会创建新的数组。
*/
```

- shift()

```
//删除并返回数组的第一个元素
var arr = new Array(3)
arr[0] = "George"
arr[1] = "John"
arr[2] = "Thomas"

document.write(arr);//George, John, Thomas
document.write(arr.shift());//George
document.write(arr);//John, Thomas

/*语法:arrayObject.shift();
返回值:数组原来的第一个元素
如果数组是空的,那么 shift() 方法将不进行任何操作,返回 undefined 值。请注意,该方法不创建新数组,
*/
```

- unshift()

```
//向数组的开头添加一个或更多元素,并返回数组的新长度
var arr = new Array()
arr[0] = "George"
arr[1] = "John"
```

```

arr[2] = "Thomas"

document.write(arr);//George,John,Thomas
document.write(arr.unshift("William"));//4
document.write(arr);//William,George,John,Thomas

/*语法:arrayObject.unshift(newele1,newele2,...,newelen);
newele1是必须的,其他随意
unshift() 方法将把它的参数插入 arrayObject 的头部,并将已经存在的元素顺次地移到较高的下标处,以便给
unshift() 方法不创建新的创建,而是直接修改原有的数组
*/

```

- slice()

```

//从某个已有的数组返回选定的元素
var arr = new Array(3)
arr[0] = "George"
arr[1] = "John"
arr[2] = "Thomas"

document.write(arr);//George,John,Thomas
document.write(arr.slice(1));//John,Thomas
document.write(arr);//George,John,Thomas

/*语法:arrayObject.slice(start,end)
返回值: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
*/

```

- sort()

```

//对数组的元素进行排序
var arr = new Array(6)
arr[0] = "George"
arr[1] = "John"
arr[2] = "Thomas"
arr[3] = "James"
arr[4] = "Adrew"
arr[5] = "Martin"

document.write(arr);//George,John,Thomas,James,Adrew,Martin
document.write(arr.sort());//Adrew,George,James,John,Martin,Thomas

/*
语法:arrayObject.sort(sortby);
sortby 可选,规定排序顺序,必须是函数

```

返回值:对数组的引用。请注意,数组在原数组上进行排序,不生成副本
*/

- splice()

```
//从数组中添加/删除项目,然后返回被删除的项目
var arr = new Array(6)
arr[0] = "George"
arr[1] = "John"
arr[2] = "Thomas"
arr[3] = "James"
arr[4] = "Adrew"
arr[5] = "Martin"

document.write(arr);//George,John,Thomas,James,Adrew,Martin
arr.splice(2,0,"William");
document.write(arr);//George,John,William,Thomas,James,Adrew,Martin

/*
语法:arrayObject.splice(index,howmany,item1,.....,itemX);
index 必需。整数,规定添加/删除项目的位置,使用负数可从数组结尾处规定位置。
howmany 必需。要删除的项目数量。如果设置为 0,则不会删除项目
*/
```

- toString()

```
//把数组转换为字符串,并返回结果
var arr = new Array(3)
arr[0] = "George"
arr[1] = "John"
arr[2] = "Thomas"

document.write(arr.toString());//George,John,Thomas

/*
语法:arrayObject.toString()
*/
```