

# 数组方法

2017年7月17日 19:04

方法	参数	作用	返回值
<code>arr.push ()</code>	一个或多个元素	向数组的末尾添加一项到多项	返回值是push之后数组的长度
<code>arr.unshift()</code>	一个或多个元素	向数组的开头添加一项到多项	返回值是unshift之后数组的长度
<code>arr.pop()</code>	无	删除数组的末尾	返回值是删除的那一项
<code>arr.shift()</code>	无	删除数组的开头	返回值是删除的那一项
<code>concat()</code>	valueN需要与原数组合并的数组或非数组值.	将传入的数组或非数组值与原数组合并	合并后的新数组
<code>reverse()</code>	无	颠倒数组中元素的位置	颠倒顺序后的数组，改变了原来的数组
<code>join()</code>	separator 可选，用于指定连接每个数组元素的分隔符。分隔符会被转成字符串类型；如果省略的话，默认作为一个逗号。如果 seprator 是一个空字符串，那么数组中的所有元素将被直接连接。	将数组中的所有元素连接成一个字符串	拼接后的字符串
<code>slice()</code>	begin 起始位置 end 结束位置	截取数组中一部分，并返回这个新的数组。	截取后的新的数组 不影响原数组
<code>splice()</code>	start 起始位置 deleteCount 删除长度 item 添加的内容 不指定参数，不截取，数组还是原来的数组，arr.splice为空	用新元素替换旧元素，以此修改数组的内容 array.splice(start, deleteCount[, item1[, item2[, ...]]])	由被删除的元素组成的一个数组会影响原数组

arr.forEach()

循环每一项的时候，都会执行传入的回调函数

执行一次，就把每一项以及下标作为这个回调函数的参数传过来

```
var a = arr.forEach(function(item,index,array){  
    console.log(this); // 通过forEach的第二参数，改变这个回调函数this的指向  
}, "abc");
```

方法forEach和for循环

相对for循环来说：

forEach不能选择从哪个位置开始循环

forEach不能停止和跳过循环

arr.map()

循环数组

返回值是一个新数组,会把执行回调函数返回值放在新数组中

```
var arr2 = [1,2,3];  
// 1+","+"2+","+"3  
// 数组转字符串，每一项拼接一个逗号转成字符串  
console.log(arr2.toString());
```

Arr.filter()

根据某个条件来过滤 成立或不成立

// 条件成立 把这一项放在新数组；不成立，就忽略，不放入新数组中

回调函数的返回值为true，那么就把循环到的这一项放在新数组中

```
var filterArr = arr.filter(function (item,index,array){  
    //console.log(item,index,array);  
    // 回调函数的返回值为true，那么就把循环到的这一项放在新数组中  
    return item < 3;  
});
```

`arr3.concat()`没有参数，复制一个数组