

sort () 方法

定义和用法：

sort () 方法用于对数组的元素进行排序

语 法： arrObject.sort

(function) **参数：**

function	可选，规定数组的排序顺序
a , b	fnction 的参数，代表每次排序时比较时的两个数组的项。sort () 排序时每次比较两个数组的项都会执行这个参数，并把两个比较的数组项作为参数传递给函数。当返回值为 1 的时候就交换两个数组项的顺序，否则就不交换。

参数说明 (两种情况)：

(1) 如果不使用参数，那么会按照字符编码 (感兴趣的同学可以去网上下载一个字符编码表，了解一下) 的顺序进行比较。它会将数组里面的项转化为字符串，然后去进行比较。

比如下列数组[2,23,3,33,1,12]，我们对它进行排序，如果按照我们正常的理解，排序后它的结果应该是[1,2,3,12,23,33],但是我们进行sort 排序后，将这些项转换成字符串，对字符串进行 unicode 编码转换：

2——50

23——5051

3——51

33——5151

1——49

12——4950

我们通过对比,发现: $49 > 4950 > 50 > 5051 > 51 > 5151$, 所以正确答案是:

[1,12,2,23,3,33]

(2) 如果想根据需求进行升序排序或者降序排序,那么我们就需要用到参数了。

它的参数是一个函数,该函数会对两个值进行比较,然后返回一个用于说明这两个值的相对顺序的数字。

升序排序: `arrObject.sort (function (a,b) {return a-b;})`

当 $a=b$ 时,返回值是 0, a 和 b 不用交换位置

当 $a>b$ 时,返回值是 1, a 放置在 b 后面

当 $a<b$ 时,返回值是-1, a 和 b 不用交换位置

降序排序: `arrObject.sort (function (a,b) {return b-a;})`

当 $a=b$ 时,返回值是 0, a 和 b 不用交换位置

当 $a>b$ 时,返回值是-1, a 和 b 不用交换位置

当 $a<b$ 时,返回值是 1, a 放置在 b 后面

比如数组: `arr = [1,15,8,9,85]`, 进行升序排序

Sort 函数在排序的时候,传递两个数进去

第一次排序:

$a=1, b=15$, 显然 $a<b$, 为-1, 那么不用交换位置, 结果是: [1,15,8,9,85]

第二次排序:

$a=15, b=8$, 显然 $a>b$, 为 1, 那么交换位置, 结果是: [1, 8,15,9,85]

第三次排序:

$a=15, b=9$, 显然 $a>b$, 为 1, 那么交换位置, 结果是: [1, 8,9,15,85]

第四次排序:

$a=15, b=85$, 显然 $a < b$, 为-1 , 那么不用交换位置 , 结果是 : [1, 8, 9, 15, 85]

返回值 :

在原数组上进行排序 , 原数组会发生变化 , 不会生成新的数组

案例 (代码如下):

没有参数 :

// 没有参数

```
<script>
  var arr=[1,2,0,21,15,6,34,9,45];
  var arrSort=arr.sort();
  console.log(arrSort); // [0, 1, 15, 2, 21, 34, 45, 6, 9]
</script>
```

有参数 :

//有参数 , 进行升序排列

```
<script>
  var arr=[1,2,0,21,15,6,34,9,45];
  var arrSort=arr.sort(function(a,b){return a-b});
  console.log(arrSort); // [0, 1, 2, 6, 9, 15, 21, 34, 45]
</script>
```