Atitit 数组与数字 文字排序

## **数字排序**

默认地，sort() 函数按照**字符串**顺序对值进行排序。

该函数很适合字符串（"Apple" 会排在 "Banana" 之前）。

不过，如果数字按照字符串来排序，则 "25" 大于 "100"，因为 "2" 大于 "1"。

正因如此，sort() 方法在对数值排序时会产生不正确的结果。

我们通过一个**比值函数**来修正此问题：

### **实例**

var points = [40, 100, 1, 5, 25, 10];

points.sort(function(a, b){return a - b});

[亲自试一试](https://www.w3school.com.cn/tiy/t.asp?f=js_array_sort_numeric_1" \t "https://www.w3school.com.cn/js/_blank)

使用相同的技巧对数组进行降序排序：

### **实例**

var points = [40, 100, 1, 5, 25, 10];

points.sort(function(a, b){return b - a});

[亲自试一试](https://www.w3school.com.cn/tiy/t.asp?f=js_array_sort_numeric_2" \t "https://www.w3school.com.cn/js/_blank)

## **比值函数**

比较函数的目的是定义另一种排序顺序。

比较函数应该返回一个负，零或正值，这取决于参数：

function(a, b){return a-b}

当 sort() 函数比较两个值时，会将值发送到比较函数，并根据所返回的值（负、零或正值）对这些值进行排序。

### **实例：**

当比较 40 和 100 时，sort() 方法会调用比较函数 function(40,100)。

该函数计算 40-100，然后返回 -60（负值）。

排序函数将把 40 排序为比 100 更低的值。

您可以使用下面的代码片段来测试数值和字母排序：

## 排序对象数组

JavaScript 数组经常会包含对象：

### **实例**

var cars = [

{type:"Volvo", year:2016},

{type:"Saab", year:2001},

{type:"BMW", year:2010}];

即使对象拥有不同数据类型的属性，sort() 方法仍可用于对数组进行排序。

解决方法是通过比较函数来对比属性值：

### **实例**

cars.sort(function(a, b){return a.year - b.year});