

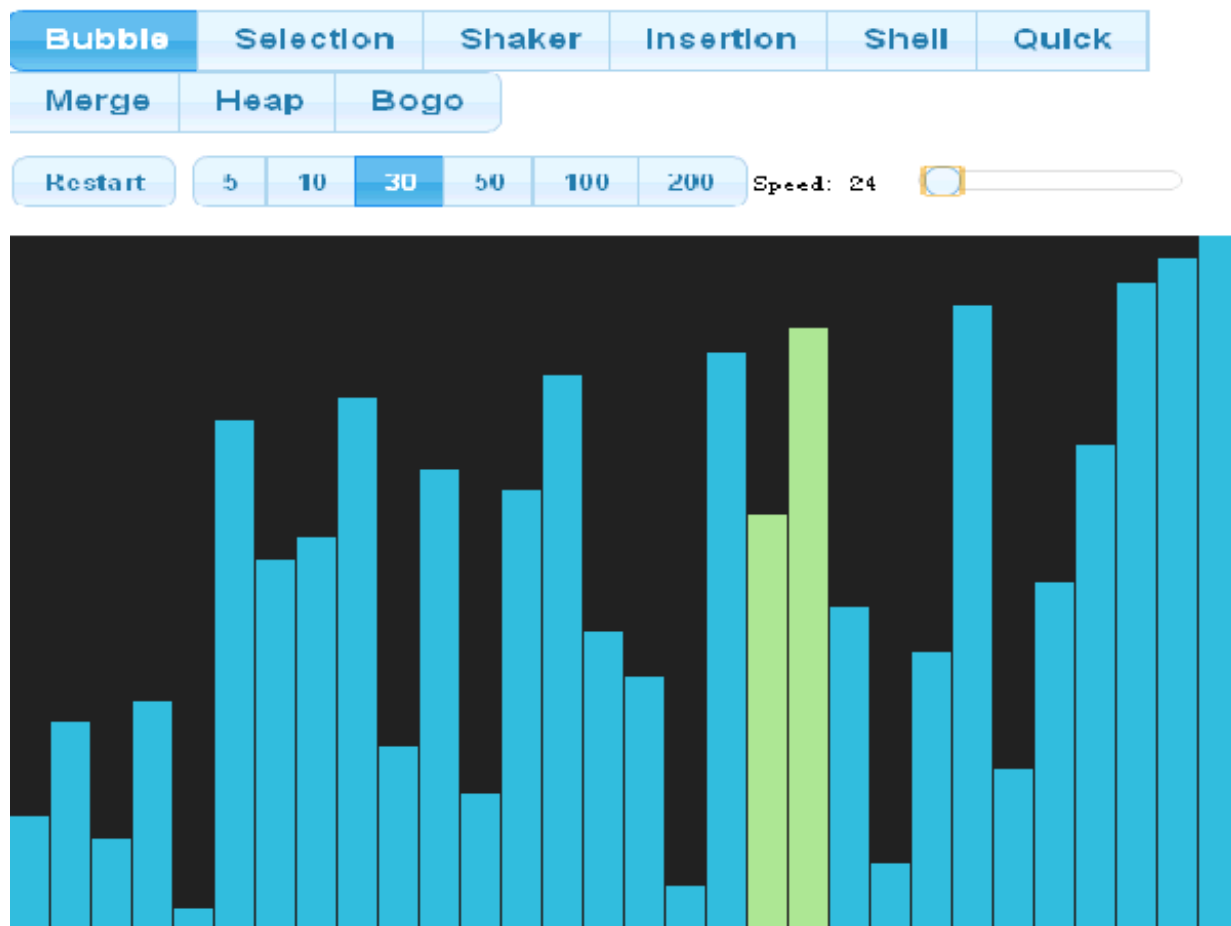
# 快速排序（Quicksort）的Javascript实现

作者：[阮一峰](#)

日期：[2011年4月4日](#)

日本程序员norahiko，写了一个排序算法的[动画演示](#)，非常有趣。

这个周末，我就用它当做教材，好好学习了一下各种排序算法。



**排序算法**（Sorting algorithm）是计算机科学最古老、最基本的课题之一。要想成为合格的程序员，就必须理解和掌握各种排序算法。

目前，最常见的排序算法大概有七八种，其中“**快速排序**”（Quicksort）使用得最广泛，速度也较快。它是图灵奖得主C. A. R. Hoare（1934--）于1960时提出来的。



"快速排序"的思想很简单，整个排序过程只需要三步：

- ( 1 ) 在数据集之中，选择一个元素作为"基准" ( pivot )。
- ( 2 ) 所有小于"基准"的元素，都移到"基准"的左边；所有大于"基准"的元素，都移到"基准"的右边。
- ( 3 ) 对"基准"左边和右边的两个子集，不断重复第一步和第二步，直到所有子集只剩下一个元素为止。

举例来说，现在有一个数据集{85, 24, 63, 45, 17, 31, 96, 50}，怎么对其排序呢？

第一步，选择中间的元素45作为"基准"。（基准值可以任意选择，但是选择中间的值比较容易理解。）

85 24 63 **45** 17 31 96 50

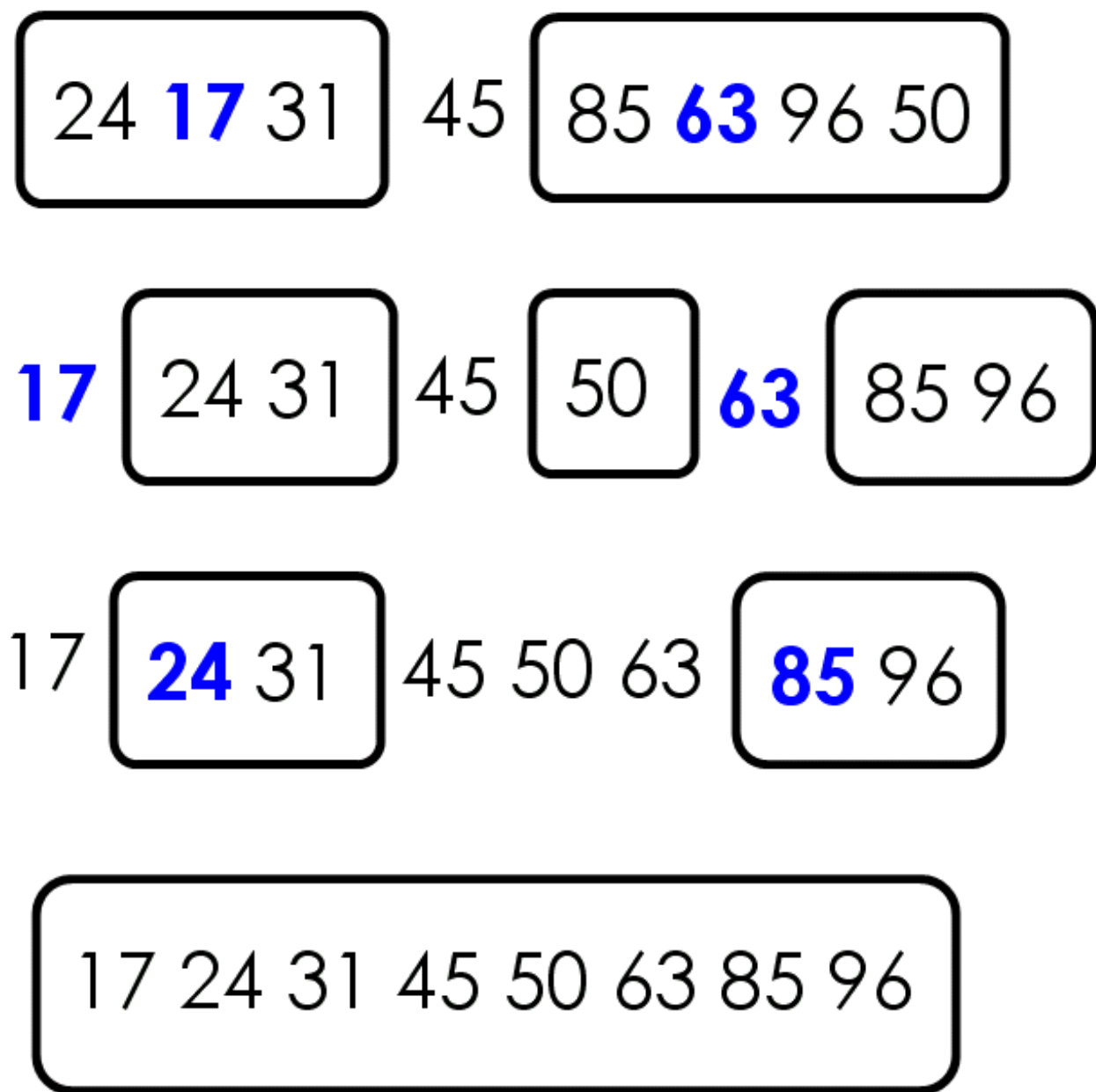
第二步，按照顺序，将每个元素与"基准"进行比较，形成两个子集，一个"小于45"，另一个"大于等于45"。

24 17 31

**45**

85 63 96 50

第三步，对两个子集不断重复第一步和第二步，直到所有子集只剩下一个元素为止。



下面参照网上的资料（[这里](#)和[这里](#)），用Javascript语言实现上面的算法。

首先，定义一个quickSort函数，它的参数是一个数组。

```
var quickSort = function(arr) {  
  ;  
};
```

然后，检查数组的元素个数，如果小于等于1，就返回。

```
var quickSort = function(arr) {  
  if (arr.length <= 1) { return arr; }  
};
```

接着，选择"基准"（pivot），并将其与原数组分离，再定义两个空数组，用来存放一左一右

的两个子集。

```
var quickSort = function(arr) {  
    if (arr.length <= 1) { return arr; }  
    var pivotIndex = Math.floor(arr.length / 2);  
    var pivot = arr.splice(pivotIndex, 1)[0];  
    var left = [];  
    var right = [];  
};
```

然后，开始遍历数组，小于"基准"的元素放入左边的子集，大于基准的元素放入右边的子集。

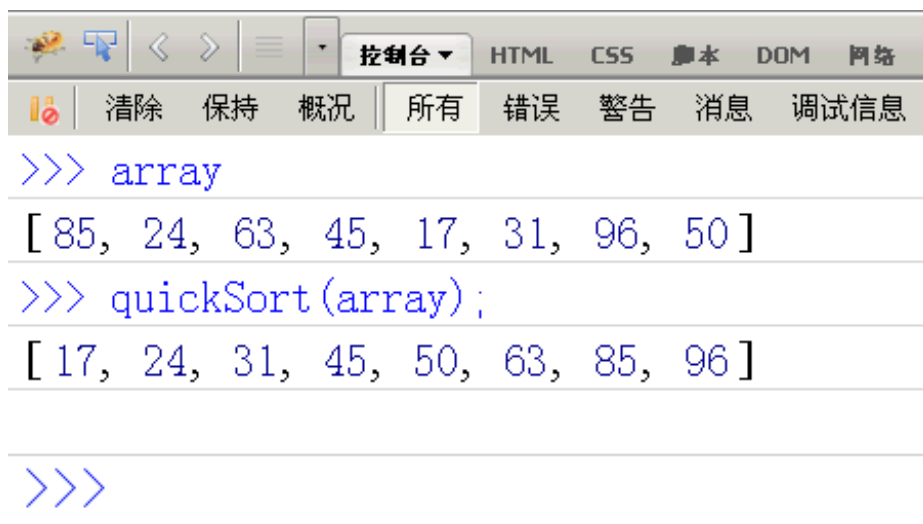
```
var quickSort = function(arr) {  
    if (arr.length <= 1) { return arr; }  
    var pivotIndex = Math.floor(arr.length / 2);  
    var pivot = arr.splice(pivotIndex, 1)[0];  
    var left = [];  
    var right = [];  
    for (var i = 0; i < arr.length; i++){  
        if (arr[i] < pivot) {  
            left.push(arr[i]);  
        } else {  
            right.push(arr[i]);  
        }  
    }  
};
```

最后，使用递归不断重复这个过程，就可以得到排序后的数组。

```
var quickSort = function(arr) {  
    if (arr.length <= 1) { return arr; }  
    var pivotIndex = Math.floor(arr.length / 2);  
    var pivot = arr.splice(pivotIndex, 1)[0];
```

```
var left = [];  
var right = [];  
for (var i = 0; i < arr.length; i++){  
    if (arr[i] < pivot) {  
        left.push(arr[i]);  
    } else {  
        right.push(arr[i]);  
    }  
}  
  
return quickSort(left).concat([pivot], quickSort(right));  
};
```

使用的时候，直接调用quickSort()就行了。



The screenshot shows a web browser's developer console. The top toolbar includes icons for a bug, a magnifying glass, and navigation arrows, along with tabs for '控制台' (Console), 'HTML', 'CSS', '脚本' (Script), 'DOM', and '网络' (Network). Below the toolbar, there are buttons for '清除' (Clear), '保持' (Keep), '概況' (Overview), '所有' (All), '错误' (Errors), '警告' (Warnings), '消息' (Messages), and '调试信息' (Debugging Information). The console output shows the following sequence of commands and results:

```
>>> array  
[ 85, 24, 63, 45, 17, 31, 96, 50 ]  
>>> quickSort(array);  
[ 17, 24, 31, 45, 50, 63, 85, 96 ]  
  
>>>
```

(完)



Leanote  
Upgrade Account