

Daily_Learning(2019-11-28)

冒泡排序如何实现，时间复杂度是多少，还可以如何改进？

```
1 function bubbleSort(arr) {
2     for (let i = 0; i < arr.length; i++) {
3         for (let j = 0; j < arr.length - i - 1; j++) {
4             if (arr[j] > arr[j + 1]) {
5                 const temp = arr[j];
6                 arr[j] = arr[j + 1];
7                 arr[j + 1] = temp;
8             }
9         }
10    }
11    console.log(arr);
12 }
13 时间复杂度:  $O(n^2)$ 
14 // 改进冒泡排序
15 function bubbleSort1(arr) {
16     let i = arr.length - 1;
17
18     while (i > 0) {
19         let pos = 0;
20         for (let j = 0; j < i; j++) {
21             if (arr[j] > arr[j + 1]) {
22                 pos = j;
23                 const temp = arr[j];
24                 arr[j] = arr[j + 1];
25                 arr[j + 1] = temp;
26             }
27         }
28         i = pos;
29     }
30     console.log(arr);
31 }
```

某公司 1 到 12 月份的销售额存在一个对象里面

如下：{1:222, 2:123, 5:888}，请把数据处理为如下结构：[222, 123, null, null, 888, null, null, null, null, null, null, null]。

```
1 let obj = {1:222, 2:123, 5:888};  
2 const result = Array.from({ length: 12 }).map((ele, index) => obj[index +  
  1] || null);  
3 console.log(result)
```

拓展：如何快速生成一个1-12的数组。

```
1 const arr = Array(12).fill().map( (_,i)=>i+1);
```