

# 前端算法面试题

2017-03-22 web前端

面试了两个星期，发现公司很喜欢考一些js算法题。当时我就烦这些公司，工作从来没用过，考来有什么用。经过多轮面试，终于认清楚了一个事实，是工作找你，不是你找工作。于是总结面试常考的一些算法题，供大家参考。

## 问题一：回文函数

```
function checkPalindrom(str) {  
    return str == str.split('').reverse().join('');  
}  
  
console.log(checkPalindrom('mamam'))
```

## 问题二：数组去重

```
let unique = function (arr) {  
    let hashTable = {};  
    let data = [];  
    for (let i = 0, l = arr.length; i < l; i++) {  
        if (!hashTable[arr[i]]) {  
            hashTable[arr[i]] = true;  
            data.push(arr[i]);  
        }  
    }  
    return data  
}  
  
console.log(unique([3, 4, 3, 2, 4, 3]));
```

## 问题三：字符串中出现最多的字符

```
function findMaxDuplicateChar(str) {  
    if (str.length == 1) {  
        return str;  
    }  
    let charObj = {};  
    for (let i = 0; i < str.length; i++) {  
        if (!charObj[str.charAt(i)]) {  
            charObj[str.charAt(i)] = 1;  
        } else {  
            charObj[str.charAt(i)] += 1;  
        }  
    }  
}
```

```
    }  
  }  
  let maxChar = '',  
      maxValue = 1;  
  for (var k in charObj) {  
    if (charObj[k] >= maxValue) {  
      maxChar = k;  
      maxValue = charObj[k];  
    }  
  }  
  return maxChar;  
}  
console.log(findMaxDuplicateChar('afjghdfraaaasdenas'))
```

微信搜索公众号 **vipweb8** 发送 **前端算法**，即可获得算法源码！

#### 问题四：冒泡排序

```
function bubbleSort(arr) {  
  for (var i = 0, l = arr.length; i < l - 1; i++) {  
    for (var j = i + 1; j < l; j++) {  
      if (arr[i] > arr[j]) {  
        var tem = arr[j];  
        arr[j] = arr[i];  
        arr[i] = tem;  
      }  
    }  
  }  
  return arr;  
}  
console.log(bubbleSort([3, 4, 5, 2, 3, 4, 4, 6766, 6]));
```

#### 问题五：快速排序

```
function quickSort(arr) {  
  if (arr.length <= 1) {  
    return arr;  
  }  
  let leftArr = [], rightArr = [], q = arr[0];  
  for (var i = 1, l = arr.length; i < l; i++) {  
    if (arr[i] > q) {
```

```
        rightArr.push(arr[i]);
    } else {
        leftArr.push(arr[i]);
    }
}

return [].concat(quickSort(leftArr), [q], quickSort(rightArr));
}

console.log(quickSort([3, 4, 5, 2, 3, 4, 4, 6766, 6]));
```

问题六：不借助临时变量，进行两个整数的交换

```
function swap(a, b) {
    b = b - a;
    a = a + b;
    b = a - b;
    return [a, b];
}

console.log(swap(4, 8));
```

问题七：找出下列正数组的最大差值

```
function getMaxProfit(arr) {
    var minPrice = arr[0], maxProfit = 0;
    for (var i = 0, l = arr.length; i < l; i++) {
        var currentPrice = arr[i];
        minPrice = Math.min(minPrice, currentPrice);
        var potentialProfit = currentPrice - minPrice;
        maxProfit = Math.max(maxProfit, potentialProfit);
    }
    return maxProfit;
}

console.log(getMaxProfit([3, 4, 5, 2, 3, 4, 4, 6766, 6]));
```

问题八：随机生成指定长度的字符串

```
function randomString(n) {
    let str = 'abcdefghijklmnopqrstuvwxyz9876543210';
    let tmp = '', l = str.length;
    for (var i = 0; i < n; i++) {
        tmp = tmp + str.charAt(Math.floor(Math.random() * l));
    }
    return tmp;
}
```

```
}  
console.log(randomString(3));
```

### 问题九：二分查找算法

```
function binarySearch(array, start, stop, num) {  
    if (stop - start == 1) {  
        if (array[stop] == num) {  
            return stop;  
        }  
        if (array[start] == num) {  
            return start;  
        }  
        return -1;  
    }  
    var center = Math.floor((stop + start) / 2);  
    if (array[center] != num) {  
        return num > array[center] ? binarySearch(array, center, stop, num) : binarySearch(array, start, center, num);  
    }  
    return center;  
}  
  
var array = [1, 3, 4, 6, 8, 9]  
console.log(binarySearch(array, 0, array.length, 9))
```

### 问题十：使用闭包获取每个li的index

```
var list = document.querySelectorAll('#test li');  
for (var i = 0, l = list.length; i < l; i++) {  
    list[i].onclick = (function (num) {  
        return function () {  
            console.log(num)  
        }  
    })(i)  
}  
  
<ul id="test">  
    <li>第一</li>  
    <li>第二</li>  
    <li>第三</li>  
</ul>
```

微信搜索公众号 **vipweb8** 发送 **前端算法**，即可获得算法源码！

