记一次面试题

作者：huazhi
原文链接：https://ld246.com/article/1550132778782
来源网站：链滴

从事前端开发一年半，下面是我做的一份面试题，分享一下，
希望大家指出我实现过程中的不足。

```
/**

extensions is an Array and each item has such format:

{firstName: 'xxx', lastName: 'xxx', ext: 'xxx', extType: 'xxx'}

lastName, ext can be empty, extType can only has "DigitalUser", "VirtualUser","FaxUser","Dept
,"AO".

**/



/**

Question 1: sort extensions by "firstName" + "lastName" + "ext" ASC

**/

function sortExtensionsByName(extensions) {

    // reserved the type of sort

    let type = String(arguments[1]).toLowerCase() ===  'desc'  ?  'desc'  :  'asc';


    if (Array.isArray(extensions)) {

        extensions.sort(function(a, b) {

            let firstLength = Math.max(a.firstName.length, b.firstName.length),

                lastLength = Math.max((a.lastName ||  '').length, (b.lastName ||  '').length),

                extLength = Math.max((a.ext ||  '').length, (b.ext ||  '').length),

                aStr,

                bStr;


            aStr = a.firstName.padEnd(firstLength, ' ') + (a.lastName ||  '').padEnd(lastLength, ' ') + (a
ext ||  '').padEnd(extLength, ' '),

bStr = b.firstName.padEnd(firstLength, ' ') + (b.lastName ||  '').padEnd(lastLength, ' ') + (b.ext ||
 '').padEnd(extLength, ' ');

return type ===  'asc'  ? aStr.localeCompare(bStr) : bStr.localeCompare(aStr);
```

原文链接：记一次面试题

```
        });
    } else {
        console.log("error: The first parameter should be an Array, e.g., [{firstName: 'xxx', lastNa
e: 'xxx', ext: 'xxx', extType: 'xxx'}]");
    }
}



/**
Question 2: sort extensions by extType follow these orders ASC
DigitalUser < VirtualUser < FaxUser < AO < Dept.
**/
function sortExtensionsByExtType(extensions) {
    // reserved the type of sort
    let type = String(arguments[1]).toLowerCase() ===  'desc'  ?  'desc'  :  'asc',
        sortObj = {
            'Dept': 0,
            'AO': 1,
            'FaxUser': 2,
            'VirtualUser': 3,
            'DigitalUser': 4
        };
    if (Array.isArray(extensions)) {
        extensions.sort(function(a, b) {
            let aNum = sortObj[a.extType],
                bNum = sortObj[b.extType];
            return type ===  'asc'  ? aNum - bNum : bNum - aNum;
        });
```

```
    } else {
        console.log("error: The first parameter should be an Array, e.g., [{firstName: 'xxx', lastNa
e: 'xxx', ext: 'xxx', extType: 'xxx'}]");
    }
}



/**
saleItems is an Array has each item has such format:
{
  month: n, //[1-12],
  date: n, //[1-31],
  transationId: "xxx",
  salePrice: number
}
**/


/**
Question 3: write a function to calculate and return a list of total sales (sum) for each quarter,
xpected result like:
[
  {quarter: 1, totalPrices: xxx, transactionNums: n},
  {....}
]
**/


function sumByQuarter(saleItems) {
    let list = [
```

```
    {quarter: 1, totalPrices: 0, transactionNums: 0},
    {quarter: 2, totalPrices: 0, transactionNums: 0},
    {quarter: 3, totalPrices: 0, transactionNums: 0},
    {quarter: 4, totalPrices: 0, transactionNums: 0}
];
if (Array.isArray(saleItems)) {
    saleItems.forEach(function(item) {
        let totalItem = {},
            num = item.salePrice,
            totalNum = 0,
            baseNum1 = 0,
            baseNum2 = 0,
            baseNum = 0;

        if(item.month < 4) {
            totalItem = list[0];
        } else if (item.month < 7) {
            totalItem = list[1];
        } else if (item.month < 10) {
            totalItem = list[2];
        } else {
            totalItem = list[3];
        }

        totalItem.transactionNums++;
        totalNum = totalItem.totalPrices;
        // precision handling
```

```
        try {
            baseNum1 = num.toString().split(".")[1].length;
        } catch (e) {
            baseNum1 =  0;
        }
        try {
            baseNum2 = totalNum.toString().split(".")[1].length;
        } catch (e) {
            baseNum2 =  0;
        }
        baseNum = Math.pow(10, Math.max(baseNum1, baseNum2));

        totalItem.totalPrices = (num * baseNum + totalNum * baseNum) / baseNum;
    });
} else {
    console.log(`
        error: The first parameter should be an Array, e.g.,
        [
          {
              month: n, //[1-12],
              date: n, //[1-31],
              transationId: "xxx",
              salePrice: number
          }
      ]
    `);
}
```

```
    return list;
}


/**
Question 4: write a function to calculate and return a list of average sales for each quarter, ex
ected result like:
[
{quarter: 1, averagePrices: xxx, transactionNums: n},
{....}
]
**/


function averageByQuarter(saleItems) {
    // reused the 'sumByQuarter'
    let list = sumByQuarter(saleItems);

    list.forEach(function(item) {
        let num = item.totalPrices,
            totalNum = item.transactionNums,
            baseNum = 0;
        if(totalNum === 0) {
            item.averagePrices = 0;
        } else {
            // precision handling
            try {
                baseNum = num.toString().split(".")[1].length;
            } catch (e) {
                baseNum = 0;
```

```
            }
            item.averagePrices = (Number(num.toString().replace(".", "")) / totalNum) / Math.pow(
0, baseNum);
        }
        delete item.totalPrices;
    });
    return list;
}



/**
Question 5: please create a tool to generate Sequence
Expected to be used like:
var sequence1 = new Sequence();
sequence1.next() --> return 1;
sequence1.next() --> return 2;
in another module:
var sequence2 = new Sequence();
sequence2.next() --> 3;
sequence2.next() --> 4;
**/

// ES5
var Sequence;
(function(){
    var unique;
    Sequence =  function(){
        if(unique){
```

```
            return unique
        }
        unique = this;
        this.index = 1;
        this.next = function() {
            return this.index++;
        };
    }
}());

// ES6
class Sequence {
    next() {
        return Sequence.index++;
    }
}
Sequence.index = 1;


/**
Question 6:
AllKeys: 0-9;
usedKeys: an array to store all used keys like [2,3,4];
We want to get an array which contains all the unused keys,in this example it would be: [0,1,5
6,7,8,9]
**/

function getUnUsedKeys(allKeys, usedKeys) {
```

```
    //TODO
    if (Array.isArray(allKeys) && Array.isArray(usedKeys)) {
        let newArr = allKeys.concat(usedKeys),
            unusedKeys = [];

        // sort the values, and then remove duplicate values in loop
       newArr.sort();
       for(let i =  0; i < newArr.length; i++){
           if(newArr[i] !== newArr[i+1]) {
               unusedKeys.push(newArr[i]);
           } else {
              i++;
           }
        }
        return unusedKeys;
    } else {
        console.log('error: The first parameter and the second parameter should be an Array');
    }
}
```