前端面试问题

ES6

## 1.自我介绍

## 2.es6新特性

1. 不一样的变量声明：const和let

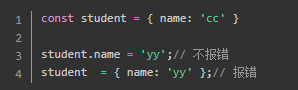
ES6推荐使用let声明局部变量，



let表示声明变量，而const表示声明常量，两者都为块级作用域；const 声明的变量都会被认为是常量，意思就是它的值被设置完成后就不能再修改了。



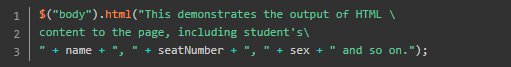
如果const的是一个对象，对象所包含的值是可以被修改的。抽象一点儿说，就是对象所指向的地址没有变就行。



* let 和 const 声明只在最靠近的一个块中（花括号内）有效
* 当使用常量 const 声明时，请使用大写变量，如：CAPITAL\_CASING
* const 在声明时必须被赋值

1. 模板字符串

传统的字符串模板：在传统的Javascript中，如果我们对DOM进行操作，字符串模板通常采用加号( + )进行字符串拼接。

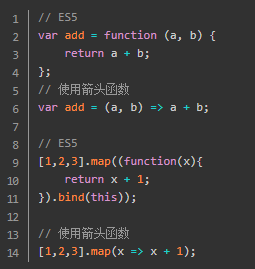


ES6模板字符串 ：以反引号( ` )表示普通的字符串，也可以表示多行字符串，同时还可以插入变量（代替传统的加号拼接）。



1. 箭头函数

* 不需要 function 关键字来创建函数
* 省略 return 关键字
* 继承当前上下文的 this 关键字



注：当你的函数有且仅有一个参数的时候，是可以省略掉括号的。当你函数返回有且仅有一个表达式的时候可以省略{} 和 return；

1. 函数的参数默认值



1. ES6中的类

## 3.箭头函数与普通函数区别

## 4.如果new一个箭头函数会怎样

## 5.promise的优势

## 6.promise有几种状态

## 7.promise.all和promise.race

## 8. 0.1+0.2 !== 0.3问题，为什么，怎么解决

## 9.原型链，调用一个实例的属性或方法，查找顺序是什么

## 10.vuex的action和mutation区别

## 11.讲项目，讲到首屏优化

## 12.vue中v-for的key值是做什么的，仅仅使用数组下标作为key值有什么缺点（面试官给讲了讲，没太听懂）

## 13. v-if和v-show区别

## 14. js事件循环

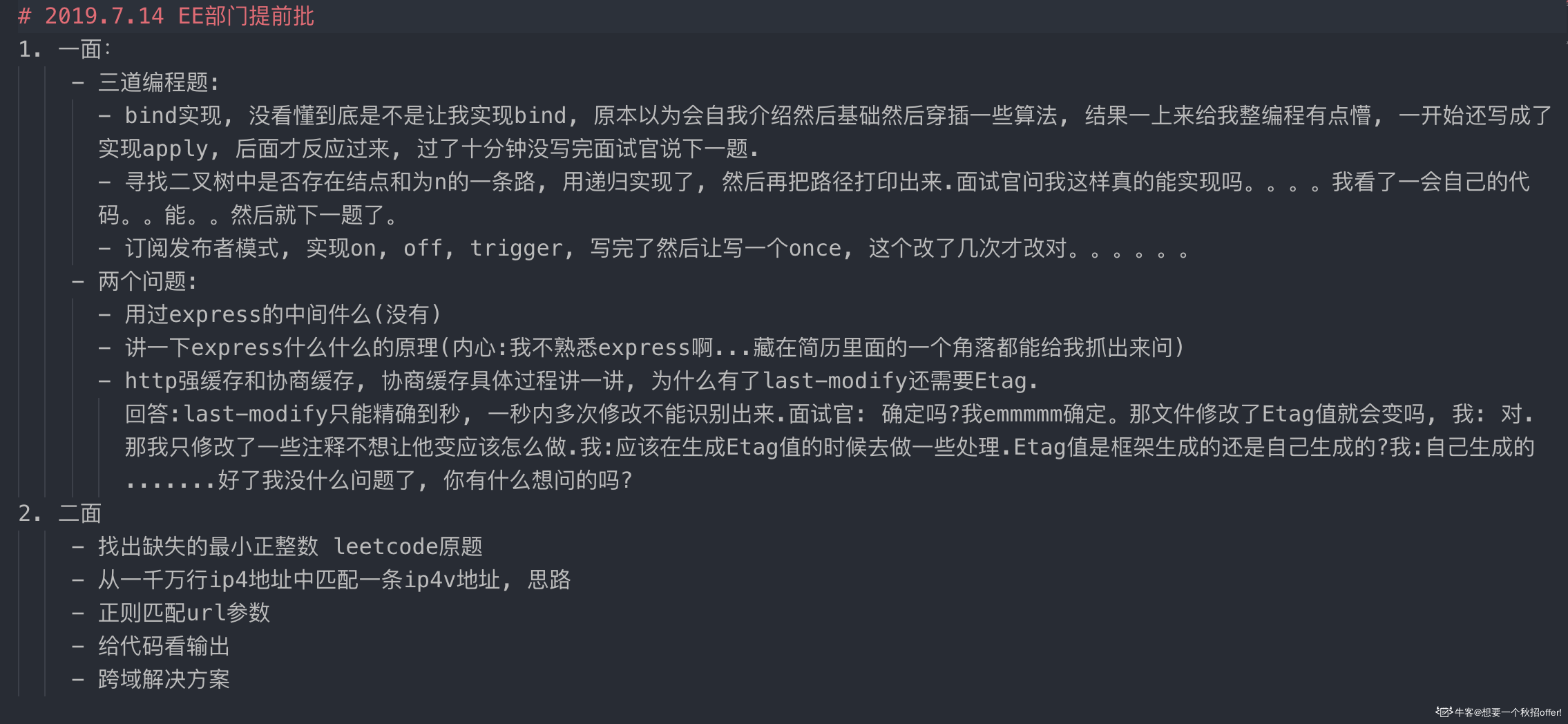
## 15. vue中的computed，什么参数需要写在computed里

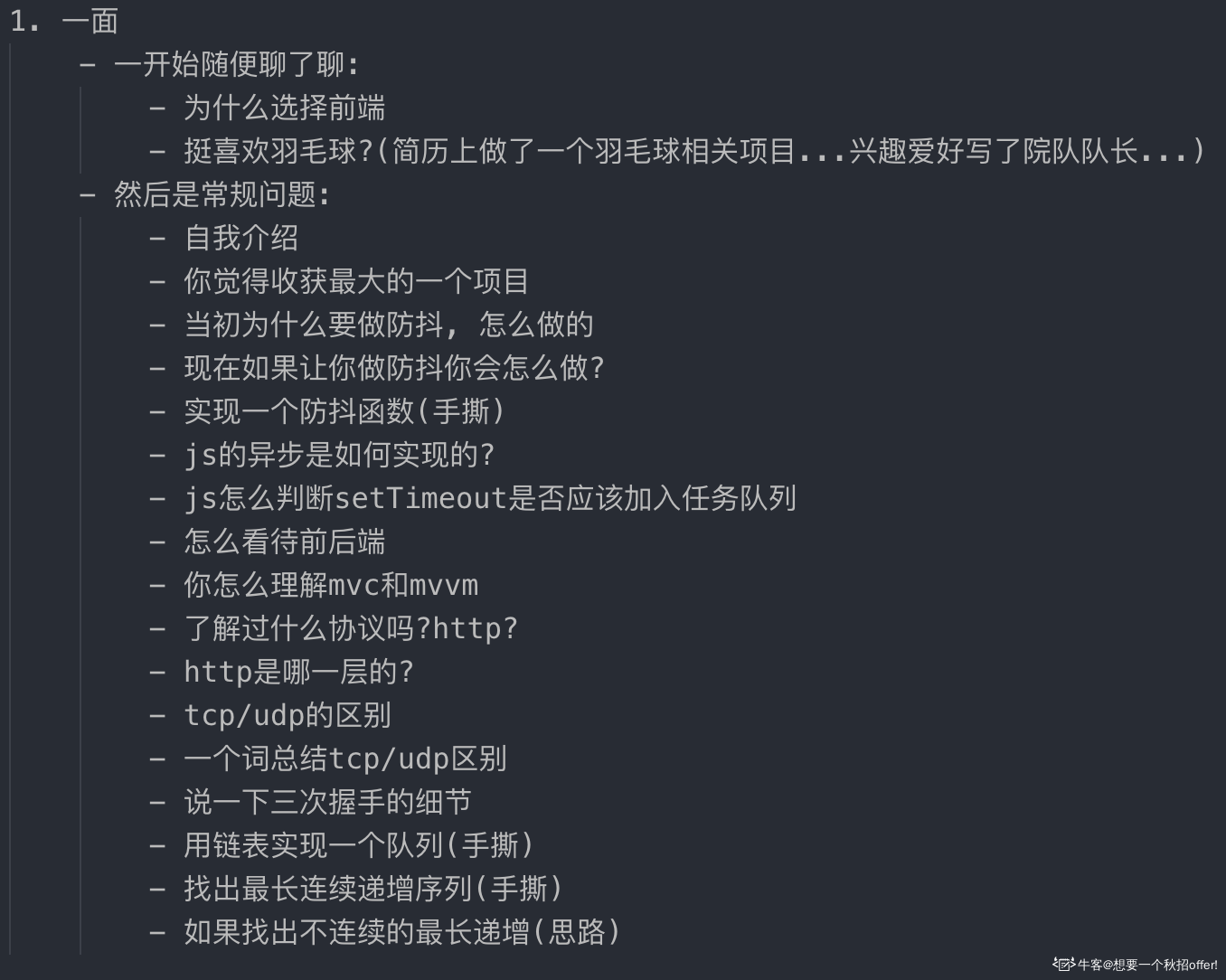
1.问js中this指向的问题，箭头函数可不可以用this来改变。

2.vue双向绑定，数组和对象的操作。

3.webpack热替换实现流程，loader和plugin实现机制。treeShaking这种机制是怎么实现的。

4.koa和express区别等。egg.js使用。





# 手撕算法题

1.栈排序

2.链表实现队列

3.最长连续递增序列

4.最长不连续序列

5.二维数组回行打印

6.无序数组构建一棵二叉排序树

7.一个数组实现两个栈

8.二叉树宽度

9.二叉树是否对称

10.链表m到n反转

11.一个n位数，现在可以删除其中任意k位，使得剩下的数最小

12.实现有符号大数链表加法，靠近头结点位置为高位

13.字符串横向改纵向

14.八皇后的问题

15.找出来数组中每个元素后边第一个比它大的值

16.给你一个二叉树，从上往下看，然后左往右顺序输出你能看到节点，同一个竖直方向上上面的节点把下面的节点遮挡住了

17.链表反转，分别用遍历与递归实现

18.完全二叉树的最大深度与节点个数

19.两个栈实现队列

20.两个有序数组交集、并集

21.给定一个有序存在重复的值链表，使得每个元素只出现一次

22.leetcode 200

23.二叉搜索树转有序双向链表

24.字符串全排列 ，可能有重复的，要去重

25.二叉搜索树第k个节点，不用中序遍历

26.有序数组查找重复元素个数

27.定长数组实现队列

28.用二分法对一个数字开根号

29.判断一颗树是不是二叉搜索树

30.Excel表的列字母转换，输入第几列，输出列字母组合

31.链表第k-1个节点

32.手撕快排

33.二分查找

34.一个无序有正有负数组，求乘积最大的三个数的乘积

35.求二叉树的深度，不使用递归

36.实现链表，无序链表，对链表值奇偶分离并排序，空间复杂度O(1)

37.单调不递减数组，给一个target，找出大于等于target的下标index

38.单调不递减链表，删除掉重复值

39.无序数组构建一棵二叉排序树

40.行和列都是有序的二维矩阵找一个target值

41.是否是回文链表

42.打印出根节点到叶子节点的最长路径

43.双链表按照奇偶顺序分成两个链表，要求不要复制链表

44.不严格递增数组，要求删除出现次数大于k的数字，要求不要新建存储空间

45.链表相邻元素交换

46.二叉树的最小公共祖先

47.字符串形式自定义进制大数相加

48.链表每隔k个反转

49.输出根节点到叶子节点路径之和为target的路径列表

50.一些数，任意排列求可形成最小的值

51.LeetCode 1038.

52.数组题，任意一个整型数组，判断是否可以将数组分为三个区间，每个区间中数值的和相同

53.已排序的整数数组去重

54.错位的全排列（第一位不能是1，第二位不能是2）

55.k路链表归并

56.非降序数组，找与target最相近的数的下标

57.二叉树逆时针打印最外层节点

58.输入一个数字n，构建一个完全二叉树并输出

59.输入一个矩阵，起始点和目标点，判断是否存在可达路径

60.无向图最短路径

61.第K层叶子节点个数