

有赞前端面经

1. 自我介绍

XXX

1. 数组的方法

slice(), map(), forEach(), map(),concat(),splice(),filter()

1. map()和forEach()的区别

map有 return,forEach(),没有return,map返回一个新数组(后面的自行脑补)

1. forEach没有return的，如何终止forEach循环呢？

当时想的是没有return，想到的是break，最后查了不行，使用try,catch才行，get到了

1. 对于对象你有什么方法进行循环遍历的？

当时说了for in 和 Object.keys().forEach()这两个，最后问还有吗？没想出来

1. for in 和 for of的区别

前面的遍历的是对象的下标并且会遍历原型链，后面的遍历对象的值，最后一再说一下都是只能遍历含有Symbol.iterator属性的对象

7.问了数据类型

基础数据类型和复杂数据类型（说里面详细的类型）

1. 说一下typeof判断的值

基础数据类型除了null是Object，剩下的都是本身，复杂数据类型除了function是function都是Object，再说一下null为什么是Object

1. typeof NaN是什么

当时心里想得是Number，但是没说出来最后说了不知道（可惜了，答案就是number）

1. 判断这个对象是数组的方法

Array.isArray(), instanceof, Object.prototype.toString().call()

1. apply, call, bind的区别

apply第二个参数是数组，call的第二个参数后是数组的解构，bind返回的是一个function，其他两个返回的是对象

1. ES6的新特性

let const, [...arr]解构, `\${}` , for in for of, Promise, 箭头函数 (有补充的下面说)

1. Promise除了 .then .catch .all .race 还有什么

么比了，考虑了20秒，说不清楚

1. async await 的优势

当时说的是 await 就是 generator 加上 Promise 的语法糖，没有一大堆的Promise.then的链式调用，简洁明了，面试官问还有么，考虑了10秒左右，想不到

1. ES6中把箭头函数和普通函数进行对比

this(这个着重问了)，没有arguments，匿名函数没有new (不全可以补充)

1. css中的grid的属性

这个因为不常用，所以我只记得叫做栅格布局，grid-template-row , grid-template-column和一个fr的单位 (剩下的就不知道了😂)

1. 问学校里有操作系统和数据结构吗

说了有 (这里面就纠结了，到底说有还是没有，有，肯定就有这方面的面试题，没有兴许直接跳过，当时想着就是诚实点说了有，下面的不要喷)

1. 进程与线程

这个我不好说 (就不记录了)

1. 数据结构说一下有什么？

图，树，链表，数组，对象，面试官说等下做一提 (内心慌得一比，真想对自己说一句)

1. 网络安全

xss和csrf这两个了说了一遍

1. 做题 (两道面试题是先写完，再问优化的)

21.1 有一个 乱序的数组nums和一个目标值target,在数组中找出两个等于target值的下标, 例如nums=[1, 8, 2, 5, 7, 9, 6], target = 11

```
let nums=[1, 8, 2, 5, 7, 9, 6]
let target = 11
function getIndex(nums, tagrets) {
  for(let i = 0; i < nums.length; i++) {
    let num = target - nums[i]
    if (nums.indexOf(num) >= 0) {
      return [i, nums.indexOf(num)]
    }
  }
  return null
}
getIndex(nums, target)
```

面试官说可以, 但是又说不用数组原生的方法怎么实现, 就是不用indexOf, 但是想到的是includes (不行, 最好时间快到了, 放弃)

21.2 实现一个深拷贝

这个就不写了, 我使用的是递归, 面试官说不是用递归解决怎么办。但是想到的是JSON.parse(JSON.stringify()),还有吗???

总结: 总体面试还是很舒服的, 全部都是比较基础的, 没有框架, webpack 上的问题偏向基础, 但是自己有一些小角落没有注意到, 该复习了