

「面经」你可能需要的三轮网易面经

前言

最近一个星期面了几家公司，最后收获了心仪的 网易offer，于是推掉了其他的面试，至于一些其他大厂，并没有投简历，由于种种原因（就是菜），准备目前先踏实的学吧。

希望大家秋招顺利，成为offer收割机。

最有意思的就是 网易有道第三轮技术面试，因为这个没有具体的答案，有兴趣的可以看看，我就先把这个第三轮面试场景题拿出来吧，其他的一些基础的话，自己可以过一遍。

网易有道

三面

第三轮面试的是一个小哥哥，面试全程大概70分钟，本来是远程面试的，但是因为声音的问题，所以选择的就电话面试了，小哥哥也挺好的，给我感觉就是很厉害的感觉，确实，接下来的问题，我就知道不简单了。

第一个场景问题

比如直播的场景，你应该知道吧，你需要实现一个这样子的场景，比如某个老师点击某个地方，比如U盘，你这个时候需要展示U盘的动画效果，比如这个时候，老师点击这个电脑屏幕，你需要展示一个小电脑的动画效果，向上述这样子，**需要在特定的时间点，完成特定的动画效果。**

- 嗯，这个问题，我的想法是，动画是例外加上去的，如果说是直接后期处理的话，那应该跟我们前端的关系不大了，所以我们接下来的问题，就是如何去处理，时间同步的问题，怎么在具体的时间点，开始展示动画呢
- 第二个问题，假设我们可以获取到某个时间点的动画，那么接下来，小哥哥，给我们提出了一个新的问题，就是当你的网络拥塞时，比如有延迟的时候，这个时候，出现卡顿的效果，原本需要5秒播放完的，可能需要7秒，那么你是如何去解决动画同步的？

嗯，我没有做过这种类似的问题，所以回答起来感觉很吃力，有了解的小伙伴可以评论留下你们的答案，我虚心学习。

第二个场景问题

有一个场景，在一个输入框输入内容，怎么更加高效的去提示用户你输入的信息，举个例子，你输入天猫，那么对应的提示信息是天猫商城，天猫集团，这个信息如何最快的获取，有没有不需要发请求的方式来实现？

- 比如数据请求的时候，尽量发的请求少，那么可以做防抖和节流处理
- 接下来的小哥哥，给了新的场景，当你的服务器挂了，数据取不到，那如何设计一个小型的本地数据库
- 接下来问题就是如何设计一个本地的数据，优化的点，就是尽可能的快，每次查询数据尽可能的快
- 我的第一个思路，是key-value这样子去设计，但是随后就被小哥哥给质疑出问题了，举个例子，如果按照你的想法，如果关键字为 天， 天猫，这样子设计数据的话，就会存在被数据重复，这样子显然是不合理的。
- 想了很久，这个时候，既然有 前缀 重合的情况，那么是不是有一种数据结构可以解决这个问题呢？👉
- 字典树，我们可以在本地建立一个预读的字典树，这样子的话，根据用户输入的内容，查字典树，这个时间复杂度大概就是 $O(m+n)$ ，所以很大程度上加快了查找效率。

当然了，有更好的解决办法的话，可以留下你们的答案，看看你们是如何解决问题的。

第三个场景问题

Git版本工具你使用过吧，那你能不能实现一个这样子的效果，完成Git Diff算法，比较两个文件的不同，然后说一说具体的思路，这个过程怎么去比较的？两个文件不同的位置如何标注出来，如何找出这个不同，具体说一说思路。

- 一开始我想的是diff算法，比如是vue中的虚拟dom算法，但是感觉不对啊，diff是做了优化的，这里很明显不合理，于是这个方法就不合理了。
- 那么两个文件，该如何快速的找到对应的两者文件的差别呢？这个问题想了好久，嗯，当时自己好像是口胡了一些思路，比如去逐行逐行的比较，这样子的话，其实也不是很合理的，仔细想一想不行
- 小哥哥提示了我，我们是不是要去找 最长的公共子串，这个是时候，我应该想起来这个是两个串的 LCS，应该就是经典的 动态规划 问题，最后一个问题，确实没有想到这个，可能就是很久没有接触这类动态规划问题了。

- 核心应该是动态规划解决LCS，以及后续的处理，可以去看看有些文章，写的很不错，我这里就不张开了。

[Git是怎样生成diff的：Myers算法](#)

嗯,三面的话，考察的是你思考问题，以及结合问题是如何分析，可能也考了算法吧，嗯，害，挺难的。

一面

面试流程50分钟，基本上自我介绍，我花了一分钟介绍完自己在校经历，接下来就是提问环节。

全程下来小哥哥耐心指点，非常温柔，这就是我现象中的面试官应该有的样子，还会耐心去提示你，有问题的话，也就帮助你，引导你怎么去回答。

H5新特性

简历上面写了这个内容，所以被问到了，害，当时脑子一蒙，都完全没有答好，这里在好好的梳理一遍？

1. 本地存储特性
2. 设备兼容特性 HTML5提供了前所未有的数据与应用接入开放接口
3. 连接特性 WebSockets
4. 网页多媒体特性 支持Audio Video SVG Canvas WebGL CSS3
5. CSS3特性

增加 [拖放API](#)、[地理定位](#)、[SVG绘图](#)、[canvas绘图](#)、[Web Worker](#)、[WebSocket](#)

然后我答了本地存储，接下来就问我这方便的问题啦？

localstorge sessionstoge 区别 应用场景

vue组件间通信

好几次面试都问了这个问题，这个问题我是这么看待的，取决于你平时项目中经常使用的方式是哪些，所以我每次都会答三种方式，反而网上8种组件间通信的方式，我记不住，也觉得了解一下即可，跟面试官交流一下，你在项目种是如何使用的即可。

遍历对象方法

- for in 遍历的也可以，不过对于非继承的属性名称也会获取到，通过hasOwnProperty判断
- Object.keys() 可枚举属性和方法名
- Object.getOwnPropertyNames() 可以获取数组内包括自身拥有的枚举或不可枚举属性名称字符串，如果是数组的话，还有可能获取到 `length` 属性

编程题👉

数组去重

常规题，讲清楚思路，最后小哥哥提示能不能使用O(n)，我给出了两者方案

- Set
- 用对象特性，我觉得他就是想考这个，给你们贴一个代码吧？

js 复制代码

```
let unique = arr => {
  let obj = {}
  return arr.filter((ele) => {
    return obj.hasOwnProperty(typeof ele + ele) ? false : (obj[typeof ele + ele] = true)
  })
}
```

数组的扁平化

js 复制代码

```
let flatArr = (arr) => {
  return arr.reduce((res, ele) => {
    if(Object.prototype.toString.call(ele).slice(8,-1) === 'Array') {
      return [...res, ...flatArr(ele)]
    }else{
      return [...res, ele]
    }
  },[])
}
let arr = [1,2,3,[2,3,4,5]];
console.log(flatArr(arr))
```

当然了，实现的方式有很多种，看你自己怎么去实现它了，最简单的就是去递归对象。

深度遍历

```

const tree = {
  name: 'root',
  children: [
    {
      name: 'c1',
      children: [
        {
          name: 'c11',
          children: []
        },
        {
          name: 'c12',
          children: []
        }
      ]
    },
    {
      name: 'c2',
      children: [
        {
          name: 'c21',
          children: []
        },
        {
          name: 'c22',
          children: []
        }
      ]
    }
  ]
}

// 深度优先的方式遍历 打印 name
// ['root', 'c1', 'c11', 'c12', 'c2', 'c21', 'c22']

```

这题，我一开始想到的就是递归的写法，写完之后，然后小哥哥问了我递归的缺点，以及如何去优化，不用递归的方法该怎么去实现？

面试的时候，没有写出来，太紧张了，不在状态，复盘的时候，写了一下用栈的实现方式？

```

function solve(root) {
  let stack = [],
      result = [];
  if(!root) return [];
  stack.push(root)
  while(stack.length) {

```

```

    let node = stack.pop()
    if(node == null ) continue
    result.push(node.name)
    for(let i = node.children.length-1; i >= 0; i--) {
        // 这里就是面试的重点,应该从后面的节点压入栈中
        stack.push(node.children[i])
    }
}
return result
}

```

链表的相加问题?

这个是LeetCode上面的题目，我好像还写过，面试的最后一题的时候，我以及蒙了，完全不知道自己在干嘛，其实**链表题都是套路**，我连套路都没有掌握，

这个我写了一个专题，把题目刷完之后，应该遇到链表问题，可以轻松解决了。

[「算法与数据结构」链表的9个基本操作](#)

二面

大概的时间上的安排，算了一下，大致上是花了50分钟吧，是个小姐姐，小姐姐好温柔，我印象中小姐姐很nice，我记得我笔试做Promise的时候，我做错了，她还特意问了我一遍，当时我大概知道错了，不过呢，这个过程小姐姐是微笑的，缓解了尴尬，而且还耐心的去指导我，给她点赞呀。

ES6语法，Promise了解吗

javascript 复制代码

```

const promise = new Promise((resolve, reject) => {
    console.log(1);
    resolve();
    reject()
    console.log(2);
})

promise.then(() => {
    console.log(3);
}, () => {
    console.log("失败的状态")
})

```

```
console.log(4);
```

我看到以后，就直接说答案了，这点不好，因为一般而言，面试官出的题目肯定有点小坑，下次要注意了，最后面试官小姐姐还是微笑的告诉我，应该这么去做，然后怎么怎么样。

聊一聊map和set

这个我是跟她说了用法，以及它们之间的区别，也就是它们经常使用的场景是哪些。

顺便的话，就聊了一下Weakmap，然后这里的难点也不是很多，就是你的明白它们两者数据结构的区别是啥，举个例子说明情况即可。

前端性能优化

这个问题太大了，而且对于一个实际开发经验为0的而言，这个问题就很置命，所以呢，我就准备了从URL到页面渲染这个一块去说，里面的优化点挺多的，可以自行去了解。

- 构建请求行
- 查缓存（重点说一说）
- dns解析（如何优化）
- tcp http（比如减少请求次数，嗯，应该还有其他优化吧，cdn？）
- 浏览器渲染过程（这里面就有优化了，比如常见的如何避免回流和重绘）
- 防抖和节流处理
- webpack打包优化也可以说一说，前提你得有自信

其他问题

这场面试的话，给我的感觉就是，并不是跟面经一样，问一些标准的答案，反而更多是跟你交流技术上的问题，比如，你遇到的问题，是如何去解决的。小姐姐还提到了，如果你需要做技术上的分享，你觉得你有哪些技术上的分享是可以跟团队分享的。

好尴尬，我一个实习生，我感觉我最近研究的是webpack打包上面的问题，以及会的都是写基础的内容，所以多多少少的话，我也把我的观点表达清楚了，表示我愿意去学习，愿意去分享这个技术。

其他面经

这里面就是其他一些公司的问题了，比如有赞，涂鸦等，问题很基础，所以我带过了，主要是觉得简单，所以就掉过啦。

你说你最近在研究webpack，说一说

这个问题，我介绍的时候，就直接说了，最近在写博客以及研究webpack，讲一讲webpack一些配置，比如loader，plugins，常见的loader，自己配过loader的话，答起来就很流畅。

然后顺便叫我说一说原理，这我暂时就不清楚了。

ES6了解吗，说一说

嗯，就按照你平常的来说，比如箭头函数，展开运算符，Promise，然后好像就说了这三个....太紧张了

其实还有很多都用过，这里记录一下

- ES6类 Class
- for...of 和 for...in
- 对象的解构
- rest操作符 / Spread操作符
- 模板字符串
- const let

闭包

嗯，这个问题，老生常谈的问题了，就过吧，不同的人，对这个有不一样的理解。

某知音科技

面试长达80分钟，我觉得我都快要被问倒了，真的，这个过程太长了呀，不过呢，这个小哥哥也非常nice，过程中有非常认真听我讲，嗯，听我一个人在那么巴拉巴拉半天。

- 原型
- 闭包

- 作用域
- 输入url过程整个过程
- https区别，TLS握手
- 浏览器缓存
- https如何保证安全，TLS握手的过程聊一聊
- vue通信方式
- vue数据响应式的原理，数组是怎么重写的

以上的答案，就不梳理了，我之前的写的博客都有涉及了，所以，好好准备的话，其实是没有多大的问题的。

可以看看我之前梳理的，基本上真的全部覆盖了👉