

迟来的前端面经

最近也是在换工作，小公司和大厂（虾皮、腾讯）都有面试。几次面试收获还是比较大的，了解许多自己的短板，当然也拿到了合适的offer。本文主要整理下面面试遇到的问题和知识点，希望对准备找工作的掘友有所帮助。

基础部分

面试的时候，前端基础知识，还是仔细准备，这一块可以问的东西很多，不管哪一面，都将会问到，或者通过面试题来考察。

1. 你对css3的动画属性了解多少？

这个问题，是在介绍项目的时候，面试官提出来的。我介绍的项目是一个利用贝塞尔曲线公式，来控制小车做曲线运动移动到对应的位置，抽取奖品。我自己是用js套用公式做的，当时面试官，有提到具体的公式是怎样的（有点慌，确实是百度来的。。。），车的坐标怎么控制，小车是否可以连续运动（项目中采用的是把所有动画函数放入一个数组，采用类似express中间件 next这样的方式来实现）。然后面试就提了上面的问题，我的回答（常规操作），animation开头的后面的属性，能记住的都说了，最后面试官问我知道 css3也可以设置贝塞尔曲线么，还有 `animationend` 事件，然后项目的新的思路就出来了，利用css3 `cubic-bezier` 结合动画结束事件 `animationend` ,就可以实现小车的曲线运动了。嗯，服气！

2. session和cookie的区别是啥？

这个网上答案有很多，建议回答的时候，就是解释一下基本的概念，然后结合场景说明下用途就ok啦。。。。

3. 聊聊js的原型链吧？

这个考的方式有很多。比如直接说出一个 `function Person`，让你画图的（如果头脑清晰，可以把函数也是对象的那部分画出来，这是一个加分项）;还有就是出一个题，让你说出 `p1.__proto__`，`Person.prototype` 的关系或是各种 `__proto__` 的指向问题，建议用纸笔仔细捋捋。

4. 防抖和节流的实现原理，和使用场景？这个也是一个面试高频问题。防抖类似于，我们排队上地铁，一个人上去后，后面的人才能接着上（one by one），一般用在 搜索展示下拉

框 这样的场景；节流，是理解为是 水龙头按节奏的滴水，一般是用在有 `onresize` `onscroll` 这样的场景。

5. http的缓存机制？协商缓存(etag,last-modify)和强制缓存（cache-control）。分别解释了各种概念后，面试官追问，既然有了last-modify为什么还要有etag：1、文件内容没有变的情况，但是last-modify会变，比如，打开文件修改，当时内容没有变化。2、就是last-modify的时间精度是秒级，但后台可能存在一秒对文件多次操作的可能。

6. 聊聊http2的特性？

概念性问题，背背书。。。。

7. addEventListener第三个参数作用

项目场景问题

1. 项目中长列表问题怎么优化？这个主要是虚拟列表，vue和react 都要对应的解决方案。具体的原理，推荐一篇文章：[虚拟滚动轮子](#)

2. 聊聊vue总key的作用？

在vue中主要是通过 `tag`元素标签 和 `key` 来判断元素是否为相同元素。如果不是同一个元素，就按照新的虚拟节点生成dom，删除老节点。如果项目在进行diff比较，重点是子元素的比较（然后子元素又会重新，按照 `tag` 和 `key` 来判断是否同一元素）这是一个递归的过程。具体的实现原理，大家多看看相关的文章，我也在摸索中，哈哈。。。

3. 平常项目中怎么做性能和异常监控？

这个主要是 `window.performance` 来监控页面的白屏时间、tcp连接时间、资源加载。
[performance](#)。异常监控，[window.error](#)

4. 如果有100万的pv访问，前端有什么方案去配合后端处理压力？

这个问题，我当然的就是骨架屏，资源懒加载这样的体验上面的解决方案；还有一些静态数据缓存。有清晰的大佬，可以帮忙解决下，感谢！

编程题

1、形似于数组var arr = [1, [2, [3, [4, 5, [6, 7], 8, [9, 10, [11]]], 12]]]扁平? 这个是之前看过牙羽大佬的 [javascript专题](#) 系列文章有提到,上代码

javascript 复制代码

```
var flat = function (arr) {  
  
  while (arr.some(i => Array.isArray(i))) {  
    arr = [].concat(...arr)  
  }  
  return arr  
}
```

2. 写出Promise.all的pollify? 上代码:

ini 复制代码

```
Promise.all = function (arr) {  
  const fns = arr || [];  
  let result = [];  
  const count = 0;  
  return new Promise((resolve, reject) => {  
    for (let i = 0, len = fns.length; i < len; i++) {  
      count++;  
      fn.then((data) => {  
        result[i] = data;  
        if (count == len) {  
          resolve(result)  
        }  
      }, (reson) => {  
        // fail  
        result[i] = reson  
        reject(result);  
      })  
    }  
  })  
}
```

3. 实现下compose函数?

这个最简单的方法是用reduceRight函数来实现, 但当时在编码的时候, 有点卡壳, 采用的while循环来实现, 请各位大佬指正哈

ini 复制代码

```
function compose(...fns) {  
  let isFist = true;  
  return function () {  
    let result = Array.from(arguments);  
    let fn;
```

```

while (fns.length) {
  fn = fns.pop();

  if (isFist) {
    result = fn.apply(null, result)
  } else {
    result = fn.call(null, result)
  }

  isFist = false
}
return result
}
}

```

测试用例：

ini 复制代码

```

const add2 = num => num + 2;
const fn1 = compose(add2);
console.log(fn1(3)); //执行后，打印 5
const sum = (a, b) => a + b;
const fn2 = compose(add2, sum); console.log(fn2(3, 2)); //执行后，打印 7

```

4. 统计一个超导文本（只有a-z）里面 每个字母出现的个数？

这个问题就部贴代码了，我也不知道自己的实现有么有问题。我简单说下我的思路：采用分片的思想统计，首先我们需要实现一个函数 `doStatic` 接受首尾索引（start, end），这个函数用来统计一小段文本（切片），一般是用对象key-value中形势来统计。然后，我们可以采用 `setTimeout`每次从文本截取一段传到函数 `dostatic` 中，把放回的key-valu对象放到一个数组 `result` 中。如果文本还有剩余，继续截取，统计。当文本全部截取完后，我们就可以通过回调或者其他方法把数组result拿出来。再这个数组result进行处理。这是我当时的思路，如果各位掘友有更好的实现和思路，欢迎留言大家一起学习哈！

小结

面试也是检验自己一种方式，只要我们自己总结就一定会有收获offer或者其他技能方面的，一起加油吧！

