

技术岗一面试题

参加字节跳动的面试，笔试一个小时，面试一个多小时，中午在公司食堂吃了顿饭，然后完毕，邮件通知我挂掉了。

先笔试，然后拿着卷子，面试官会根据笔试题从后往前问，整个过程感觉学习到超级多东西

先说笔试题～来画重点来

1.块级元素和行内元素有什么区别，举例常用的块级和行内元素，行内元素有 padding / margin 吗？

2.call,apply,bind 的区别，并举例使用的场景

3.画出一个正方形，并且自适应，列出的方法越多越好

这个题，注意 padding: 30%，
这里问到很多题

```
<!-- 窗口大小 500*500，img 大小 1000*800-->
```

```
<p>
```

```
<a><img></img></a>
```

```
<p>
```

首先问 a，p 宽高

然后给 img 加了绝对定位，a，p 的宽高

然后给 p 加 relative，img 加 margin-top: 30%，margin-left: 30%，

变成 top: 30%，left: 30%

这个题真是能问的多种多样，多方向的考你 css 基础知识，a 标签还有 img 我感

觉还能考关于基线的问题。

.....这个题之后面试结束了。

当时到这个题的时候 12 点多了，然后面试官带我去吃饭，说这个是他们公司的习惯，都会带候选人来吃顿食堂的饭，有红烧肉，吃的好爽，期间，面试官吐槽了下百度。。。。。。问了我下想换工作原因，公司的情况，和我实际的工作经验等等等。

4.父级元素下面无固定宽高的块元素，实现水平垂直居中

使用 flex，问了一会 flex。面试官要是想问应该还能问很多，感觉知识很充分啊

5.argument 是数组吗？不是的话，怎么变成数组

不是，类数组对象，

```
var a=[].slice.call(arguments)
```

5.写出下面会输出的值

```
if([]==false){console.log(1)};if({}==false){console.log(2)};if([]){console.log(3)}if([1]==[1]){console.log(4)}
```

```
// 只输出 1,3
```

这个是隐式转换，if([])直接调用 boolean()方法，==号的转换套路要知道

6.修改错误，可以使用 es6

```
for (var i = 0; i < 5; i++) {  
  
    setTimeout(function() {  
  
        console.log(i);  
  
    }, 1000 * i);  
  
} // 使得其输出为 0, 1, 2, 3, 4
```

这个把 `var` 变成 `let` 就可以，也可以使用一个闭包，不过要彻底理解这个题，很有可能还能问出来 `js` 事件循环。

7. 写出节流函数, 并说明在什么场景下使用

这个题，我当时随手写了一个防抖，结果被问了很多防抖的话更多运用在输入框搜索，每次 `keyup` 事件走一个接口

```
//其实是防抖 var debounce = function(delay, fn){  
  
  var last  
  
  return function(){  
  
    var ctx = this, args = arguments  
  
    clearTimeout(last)  
  
    last = setTimeout(function(){  
  
      fn.apply(ctx, args)  
  
    }, delay)  
  
  }  
  
}
```

由于写了防抖，于是面试官给我讲解了下节流，大约是每一个 `delay` 时间段内走一次 `fn`，感觉如果答出来继续会问如果是滚动条，想一开始触发就走 `fn`。`underscore.js` 里面关于节流写的挺完美的，可以参照。

8. 从一个无序，不相等的数组中，选取 **N** 个数，使其和为 **M** 实现算法

这个其实是一个 01 背包的简单版，然而我并不会，乱七八糟说了一个思路，然后面试官直接说，如果数组里面有负数，一上来面试就直奔算法题，紧张。

面试官会引导你的思路，也没有为难，感觉我不会，就把问题简单话。

如：选取 4 个数和，可以先把可以组成的四个数的情况全部列出来，然后筛选和可以为 M 的。

面试官很棒有没有，我说用穷举法。好了，怎么穷举，我当时满脑子已经是这是笛卡尔啊笛卡尔乘积啊啊啊啊。

然后面试官继续引导我，说可以使用归纳法先思考，如组成 2 个数为 M 的和
答：两个 `for` 循环嵌套呗，多个呢？递归啊，写出来试试。。。。。。然后没有然后，绵绵飒飒我当时觉着自己要恶补算法了。

整个面试过程更像是一次学习，面试过程中遇到不会的会引导你的思路，最后没有写出来也会告诉你应该怎么写，也会等很久等你思考，非常的有耐心，也没有架子。

问了下别人，一面过了，二面会问关于设计模式，`bind` 的实现方法，还会看下你的 `github`