

【web 明教】

快手主站前端面试



本面试题仅供观看，不可商用
如商用后果自负

欢迎大家为本群做推广
加群暗号：我是我



快手一面

- 项目难点介绍
- padding-top 百分比的话是相对于哪个？
- 一道关于层叠上下文的题（z-index、position 都存在时才会创建层叠上下文。● 出了一道题，var 和 let 的应用差别）
- Promise、setTimeout 的顺序输出问题
- async 使用 await 搭配 Promise 的顺序输出问题
- box-sizing 说一下
- opacity: 0; visibility: hidden; display: none;的区别
- display: none;和 visibility: hidden;两者的性能差距？
- margin 垂直重叠问题，并怎么解决

手写:

- 实现函数：将数组扁平化、去重、从小到大排序。
- 实现 Array.prototype.reduce
- 实现 Promise.all

快手二面

- 说一下网络七层模型
- Http、smtp、websocket、tcp、ip 分别在哪一层
- 200 和 304 的区别
- 200、304 响应头的区别（主要说的是 `expire`、`cache-control`、`if-modified`、`etag`） ●

http 1.0/1.1/2.0/3.0 的区别（这一题让说一下 HTTP 版本的发展史，每一个版本解决

了上一个版本的什么问题，为什么会出现这个问题，这一块问的非常细，回答的好菜） ●

TCP、UDP 的区别

- 如何实现一个基于 UDP 的可靠传输（被虐了）

手写：

- 随机生成一个 hex 颜色（`#23f` 或者 `#ffe532`）

```
const fn = () => {  
  let res = '#';
```

```
  const len = Math.random() < 0.5 ? 3 : 6;  
  const hash = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 'a', 'b', 'c', 'd', 'e', 'f'];  
  for (let i = 0; i < len; i++) {  
    res += hash[Math.floor(Math.random() * 16)];  
  }  
  return res;  
}
```

随机生成一个 RGB 颜色

```
const genRGB = () => {  
  const R = Math.floor(Math.random() * 256);  
  const G = Math.floor(Math.random() * 256);  
  const B = Math.floor(Math.random() * 256);  
  return `rgb(${R}, ${G}, ${B})`;  
}
```

写一个 RGB 颜色转化为 Hex 颜色的函数（其中涉及到十进制转换成十六进制，Hex 的缩写情况）

```
const numChange = num => {  
  if (num === 0) return '0';  
  let res = '';  
  const hash = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 'a', 'b', 'c', 'd', 'e', 'f'];  
  while (num !== 0) {  
    res = hash[num % 16] + res;  
    num = Math.floor(num / 16);  
  }  
  return res;  
}  
  
const rgbtoHex = arr => {  
  const ans = [];  
  for (let i = 0; i < arr.length; i++) {  
    ans.push(numChange(arr[i]));  
  }  
}
```

```
let flag = true;  
for (let i = 0; i < ans.length; i++) {  
  if (ans[i].length === 2 && ans[i][0] !== ans[i][1])  
    { flag = false;  
      break;  
    }  
}  
  
return '#' + flag ? ans[0][0] + ans[1][0] + ans[2][0] : ans[0] + ans[1] + ans[2];  
}
```

三面

上来就是一道题，

```
var a = {  
  b: 123,  
  c: '456',  
  e: '789',  
}  
  
var str=`aa{a.b}aaa{a.c}aa {a.d}aaaaaa`;
```

实现函数使得将 **str** 字符串中的{}内的变量替换，如果属性不存在保持原样（比如{a.d}）

基本思路就是提出来{}的内容，再进行替换，第一版写出来之后，让考虑一下深层的情况，比如：{a.d.e}。

最后写了一版，能正常跑通测试例子：

```
const fn1 = (str, obj) => {  
  let res = "";  
  let flag = false;  
  let start;  
  for (let i = 0; i < str.length; i++) {  
    if (str[i] === '{') {  
      flag = true;  
      start = i + 1;  
      continue;  
    }  
    if (!flag) res += str[i];  
    else {  
      if (str[i] === '}') {  
        flag = false;  
        res += match(str.slice(start, i), obj);  
      }  
    }  
  }  
  return res;  
}  
  
const match = (str, obj) => {  
  const keys = str.split('.').slice(1);  
  let index = 0;  
  let o = obj;  
  while (index < keys.length) {  
    const key = keys[index];  
    if (!o[key]) {  
      return `${str}`;  
    } else {  
      o = o[key];  
    }  
  }  
}
```

```

    }

    index++;
}

return o;
}

```

面完结束后探讨了一下这个题，还有另一种思路，先做对象的遍历，然后匹配字符串，面试官问我哪种好，我答的是先做对象遍历那种，结果错了。因为对象的遍历是不确定的，可能会遇到一个深层嵌套很多的情况，而且不容易扩展。

然后是一道定位问题：

```

<span>aaaa</span>
<div>bbbb</div>
<input value='ccc'>
三行还是三列？

<span>aaaa</span>
<div style="position: absolute">bbbb</div>
<input value='ccc'>

```

这样呢？

```

<span>aaaa</div>
<div style="position: relative; top: -20px;">bbbb</div>
<input value='ccc'>

```

这样呢？

hr 面

hr 打电话进行的 hr 面，原本是二三十分钟，最后聊了五十分钟，主要从简历项目、实习的角度，从大一开始到现在的项目、学习问了一遍，还有一些就是常规问题了：

- 为什么不在字节转正？
- 你的优势？
- 有没有遇到一些很苦难的事情？怎么解决的？
- 实习的收获？
- 对于技术团队的期待？
- 对于 base 地点的要求？

最后问了下前端所负责的方向，面的部门是快手主战，移动端主要针对快手 APP 下的 H5，搭载的 toB 端这样子

【web 明教】天下武功，唯快不破

Web 明教