

美团四轮面试面经

一面:

1. 实现一个深拷贝，（考虑类型，循环调用优化，公共优化引用等问题）

ini 复制代码

```
function clone(target, map = new WeakMap()) {

    // 克隆原始类型
    if (!isObject(target)) {
        return target;
    }

    // 初始化
    const type = getType(target);
    let cloneTarget;
    if (deepTag.includes(type)) {
        cloneTarget = getInit(target, type);
    }
    // 防止循环引用
    if (map.get(target)) {
        return map.get(target);
    }
    map.set(target, cloneTarget);

    // 克隆set
    if (type === setTag) {
        target.forEach(value => {
            cloneTarget.add(clone(value, map));
        });
        return cloneTarget;
    }

    // 克隆map
    if (type === mapTag) {
        target.forEach((value, key) => {
            cloneTarget.set(key, clone(value, map));
        });
        return cloneTarget;
    }
}
```

```
// 克隆对象和数组
const keys = type === arrayTag ? undefined : Object.keys(target);
forEach(keys || target, (value, key) => {
  if (keys) {
    key = value;
  }
  cloneTarget[key] = clone(target[key], map);
});

return cloneTarget;
}
function getInit(target) {
  const Ctor = target.constructor;
  return new Ctor();
}
```

1.1 js的数据存储（堆，栈）

1.2 深拷贝和浅拷贝

1.3 如何判断一个数组，typeof为什么不行

1.4 循环引用是如何解决的，map和weakMap的区别

1.5 为什么用到foreach？for in和for of的区别？

2. URL后发生了什么？

讲之前问一下面试官要详细还是概述

3. [JavaScript]为什么会阻塞渲染？css会不会阻塞渲染

js在加载之前，浏览器不能识别js具体的操作，js是可以操作DOM的，如果在修改元素属性的同时渲染界面，会出现不可预期的结果，导致渲染混乱，因此浏览器设置了GUI渲染现成和js线程互斥，在js执行的时候，GUI线程会被挂起，等空闲的时候再执行。

dom解析和css解析是两个并行的线程，所以css加载不会阻塞dom的解析

但是render tree是依赖dom tree和cssom tree的，因此，css的加载会阻塞dom的渲染

4. 浏览器的事件循环机制讲一下，node了解么，跟node循环机制有什么区别

5. [项目]中用node处理了哪些工作，大量的日志工作你是如何处理的

大数据（缓冲队列）？

6. 小程序的架构方案解决了什么问题，是怎么与native通信的？

jsBridge，追问jsBridge是实现方式，如何保持时序？

7. 开发中有遇到哪些安全问题，怎么解决的？

8. [项目]中的跨域问题是如何解决的？

9. https是怎么保证安全的？开发时如何抓包，抓包的原理是什么

中间人攻击 ~

10. js的垃圾回收中，内部函数返回给上层的对象是如何管理的？

11, meta viewport的原理是什么

meta viewport 标签的作用是让当前 viewport 的宽度等于设备的宽度，同时不允许用户进行手动缩放

viewport的原理：移动端浏览器通常都会在一个比移动端屏幕更宽的虚拟窗口中渲染页面，这个虚拟窗口就是 viewport; 目的是正常展示没有做移动端适配的网页，让他们完整的展示给用户；

解析：

Viewport：字面意思为视图窗口，在移动 web 开发中使用。表示将设备浏览器宽度虚拟成一个特定的值（或计算得出），这样利于移动 web 站点跨设备显示效果基本一致。移动版的 Safari 浏览器最新引进了 viewport 这个 meta tag，让网页开发者来控制 viewport 的大小和缩放，其他手机浏览器也基本支持。

在移动端浏览器当中，存在着两种视口，一种是可见视口（也就是我们说的设备大小），另一种是视窗视口（网页的宽度是多少）。

举个例子：如果我们的屏幕是 320 像素 * 480 像素的大小（iPhone4），假设在浏览器中，320 像素的屏幕宽度能够展示 980 像素宽度的内容。那么 320 像素的宽度就是可见视口的宽度，而能够显示的 980 像素的宽度就是视窗视口的宽度。

为了显示更多的内容，大多数的浏览器会把自己的视窗视口扩大，简易的理解，就是让原本 320 像素的屏幕宽度能够容下 980 像素甚至更宽的内容（将网页等比例缩小）。

Viewport 属性值

- width 设置 layout viewport 的宽度，为一个正整数，或字符串"width=device"
- initial-scale 设置页面的初始缩放值，为一个数字，可以带小数
- minimum-scale 允许用户的最小缩放值，为一个数字，可以带小数

- maximum-scale 允许用户的最大缩放值，为一个数字，可以带小数
- height 设置 layout viewport 的高度，这个属性对我们并不重要，很少使用
- user-scalable 是否允许用户进行缩放，值为"no"或"yes", no 代表不允许，yes 代表允许这些属性可以同时使用，也可以单独使用或混合使用，多个属性同时使用时用逗号隔开就行了。

12. flex布局, flex属性

13. vue的响应式

14. vue3.0

二面:

1. 项目简述

可视化搭建

目标用户

nocode

2. 项目难点, 怎么解决的?

难点1, 最初的设计方案, 技术选型, 维护性

难点2, 跨小程序端处理

难点3, 持续的性能优化, 从3.5s优化到秒开

3. 性能优化做了哪些工作

- 1, 工程规范 + 开发规范 (将一些优化策略通过规范约束)
- 2, 工具分析
- 3, 图片优化, 长列表优化, 预渲染, 骨架屏, 小程序特性, 网络层优化 (数据并行请求, dns预解析, 缓存策略, http2.0)

4. 性能指标有哪些

遵循谷歌的 RAIL 模型

FP, FCP, 白屏时间, 首屏加载时间

5. 预渲染是如何实现的？

6. 缓存策略？后续的[项目]规划

7. 算法题

统计字符串中次数最多的字母

ini 复制代码

```
function findMaxDuplicateChar(str) {  
  if(str.length == 1) {  
    return str;  
  }
```

```
}
var charObj = {};
for(var i = 0; i < str.length; i++) {
    if(!charObj[str.charAt(i)]) {
        charObj[str.charAt(i)] = 1;
    } else {
        charObj[str.charAt(i)] += 1;
    }
}
var maxChar = '',
    maxValue = 1;
for(var k in charObj) {
    if(charObj[k] >= maxValue) {
        maxChar = k;
        maxValue = charObj[k];
    }
}
return maxChar + ': ' + maxValue;
}
```

三面:

1. 项目简述

可视化搭建

2. 项目中的难点是如何解决的?

3. 跟竞品相比, 有哪些做的好的地方, 哪些需要提升的地方

4. 如何设计一个监控系统

5. 项目管理, 需求, 质量

6.算法题：顺时针打印矩阵

输入一个矩阵，按照从外向里以顺时针的顺序依次打印出每一个数字，例如，如果输入如下4 X 4矩阵： 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 则依次打印出数字 1,2,3,4,8,12,16,15,14,13,9,5,6,7,11,10.

思路

- 1.选坐标为(0,0),(1,1)...的点记为(start,start)，作为开始坐标,下一圈开始坐标为(start+1,start+1);
- 2.判断是否进入下一圈(即是否打印完成)的条件是rows>start2 && cols>start2;
- 3.打印一圈的左上角坐标为(start,start)，右下角的坐标为(cols-start-1,rows-start-1)
- 4.根据一圈左上角和右下角坐标判断“从左到右”，“从上到下”，“从右到左”，“从下到上”需要打印的点。

ini 复制代码

```
function printMatrix(matrix) {
    if (matrix == null || matrix.length == 0) {
        return;
    }
    var rows = matrix.length;
    var cols = matrix[0].length;
    var start = 0;
    var result = [];

    while (cols > start * 2 && rows > start * 2) {
        var endX = cols - 1 - start;
        var endY = rows - 1 - start;
        //从左到右打印一行
        for (var i = start; i <= endX; i++) {
            result.push(matrix[start][i]);
        }
        //从上到下打印一行
        if (start < endY) {
            for (var i = start + 1; i <= endY; i++) {
                result.push(matrix[i][endX]);
            }
        }
    }
}
```



```

//从右到左打印一行
if (start < endX && start < endY) {
    for (var i = endX - 1; i >= start; i--) {
        result.push(matrix[endY][i]);
    }
}
//从下到上打印一行
if (start < endX && start < endY - 1) {
    for (var i = endY - 1; i >= start + 1; i--) {
        result.push(matrix[i][start]);
    }
}
start++;
}
return result
}

```

四面:

1. [项目]简述，你在[项目]中的指责，你的亮点工作是什么

可视化搭建

2. 性能指标，白屏时间是怎么计算的，业内的统计方式有哪些

如何做的性能分析，分析指标是什么，业内的性能指标了解过么

3. js的垃圾回收讲一下

4. [项目]中的内存泄露有遇到过么？

5. 用到了哪些性能优化策略

6. ssr具体讲一下

为什么要打两个包，ssr存在的问题是什么，怎么去规避的？

7. http2.0的特性，为什么目前http2.0还没有完全推广，你觉的还有哪些问题

8. 你刚才说的[项目]中，小程序的同构方案能详细讲一下么

9. CDN的实现原理是什么，如何优化？