前端面试

# 自我介绍

各位面试官好，我叫郭洋洋，河北邯郸人，2017年毕业于河北经贸大学网络工程专业，本科学历。2016年开始实习，一直至今都在中科软科技股份有限公司工作。我们部门的业务主要是交通运输和医疗行业。我在公司是前端开发，使用的技术是vue和react。我的工作是参与前端框架的搭建，工时的评估，组件的封装，项目的维护。在空闲的时间里，我会在掘金，知乎上学习一些前端相关的知识。我未来的发展定位是前端架构师。

<https://blog.csdn.net/u012194956/category_7410376.html>

<https://segmentfault.com/a/1190000018630871>

<https://segmentfault.com/a/1190000016885832?utm_source=tag-newest>

[https://www.cnblogs.com/onesea/p/13092664.html#\_label0](https://www.cnblogs.com/onesea/p/13092664.html" \l "_label0)

# Js部分

1. JavaScript中的堆、栈、队列分别是什么？

2、var、let、const区别是什么？

3、Const定义的变量是否可修改？

4、JavaScript中的数据类型都有哪些？

答：9种，6种基本数据类型，3种引用类型

基本数据类型：string，number，boolean，null，undefined，symbol

引用类型：对象，数组，函数

1. New出来的对象和直接定义的对象有什么区别？

答：new出来的对象存放在堆内存中，直接定义的对象存放在栈内存中。

6、对this的理解？以下输出的是什么？

指向调用它的对象

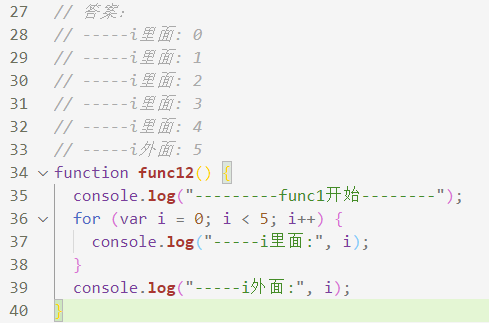
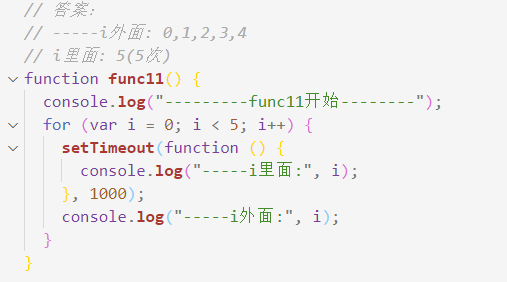
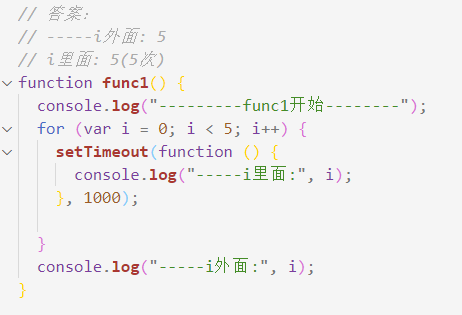
7、对定时器和延时器的理解？以下输出什么？

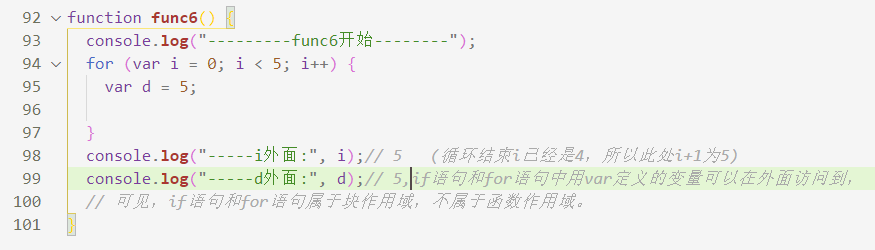
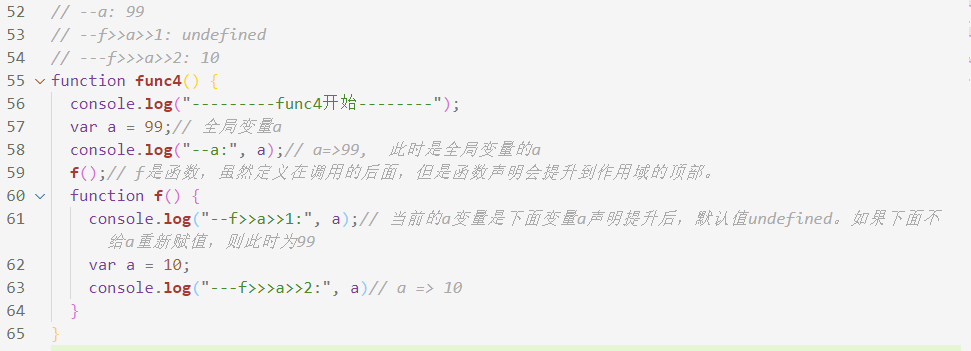
For循环里面输出0、1、2、3、4、5，延时器里面输出6个6

8、在js中，0.1+0.2等于0.3吗？

不等，因为十进制转成二进制时，有精度问题0.30000000000000004

1. 变量





数组

1. 数组常用方法有哪些？
2. unshift：在第一位添加元素，并返回新的长度。直接修改arrayObject，不创建新的数组
3. Push：在末尾添加元素，并返回新的长度。直接修改arrayObject，不创建新的数组
4. Shift：删除第一位元素，并返回第一个元素的值。如果是空的，返回的是undefined。直接修改arrayObject，不创建新的数组
5. Pop：删除最后一位元素。如果是空的，返回的是undefined。直接修改arrayObject，不创建新的数组
6. 深拷贝和浅拷贝
7. 变量提升

# Css部分

1. 背景图如何实现自适应？

background-repeat: no-repeat; background-size: 100% 100%;

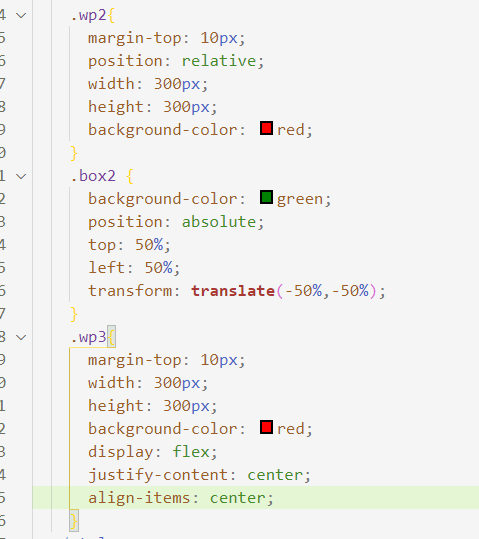
1. 块级元素，已知元素宽高，如何实现水平垂直居中？
2. 绝对定位+top+left+margin-left+margin-right
3. 绝对定位+left+right+top+bottom+margin:auto





3、块级元素，未知元素宽高，如何实现水平垂直居中？

1、flex布局；2、table；3、绝对定位+transform



4、什么是盒子模型？

答：1）每个元素被表示为一个盒子模型，包括四部分：外边距，内边距，边框，内容，它在页面中所占的实际大小是四部分之和

1. 盒模型有两种：标准盒模型和ie盒模型
2. 两种盒模型的区别：标准盒模型大小就是content，ie盒模型是content+padding+border
3. 通过设置box-sizing:content-box;boder-box
4. position有几种？

答：5种，相对定位，绝对定位，fixed，static（默认），inherit（继承父级）

# Html部分

1. 页面在浏览器加载时的过程？
2. 常用的浏览器内核？

答：1）国内绝大部分浏览器ie，360，搜狗：trident内核

1. 国产大部分双核浏览器中的一核，safri（苹果）、谷歌：webkit内核
2. 浏览器内核分成两部分：渲染引擎，js引擎。
3. Html5新特性

答：1）语义化标签，header，nav，fotoer，article

1. 音频，视频标签
2. Canvas
3. 地理定位
4. 缓存

# Webpack部分

1. 是否是否搭建过vue脚手架？用什么配置？vue-cli?

2、Webpack常用的配置有什么？

Entry：入口；Output：打包；Loader；Babel-loader；Css-loader；File-loader；url-loader；plugin----html-webpack-plugin(生成模板)；copy-webpack-plugin（复制文件）；resolve—alias(设置别名)

# Vue部分

1. Vue使用过程中的坑有哪些？
2. Vue和什么ui组件组合开发？
3. Vue搭配使用什么字体图标库？

答：font-awesom

1. Vue的核心是什么？vue和普通html的加载方式有什么区别？
2. Vue父子组件怎么传值、调用方法？

Props(接收时，需要定义吗？还是直接使用？)，$emit

1. 兄弟组件怎么传值、调用方法？

中间组件然后使用props；中央事件总线

1. Vue全局变量怎么存储的？
2. Vue-store再刷新时会有什么问题码？
3. Vue的ref什么时候使用？
4. Vue的$nextclick什么时候使用？
5. Vue路由hash和history有什么区别？
6. Vue路由怎么配置加载？
7. 单页面和多页面的区别
8. 列表页调到详情页，再跳转到列表页，列表页的查询条件怎么缓存？keep-alive
9. 移动端，字体怎么处理。自适应不同屏幕的手机，ipad
10. 是否使用过，ant design vue
11. 传表单和不同json的区别？
12. [https://www.cnblogs.com/onesea/p/13092664.html#\_label0](https://www.cnblogs.com/onesea/p/13092664.html" \l "_label0)
13. Vue实现数据双向绑定的原理？

答：采用数据劫持和发布者、订阅者模式的方式。通过object.definePorperty()来劫持各个属性的getter、setter，在数据变动时通过发布消息给订阅者，触发相应监听回调。

1. Vue.js的两个核心是什么？

答：1）数据驱动：viewModal保证数据和视图的一致性

2）组件系统：应用类ui可以看做都是由组件树构成的

1. Vue中key值的作用是什么？

答：key的作用是为了高效的更新虚拟DOM。

1. Vue的生命周期？

答：创建前（beforeCreate），创建后（created），载入前（beforeMounte），载入后（mounted），更新前，更新后，销毁前，销毁后

1. 如何让css只在当前组件使用？

答：在组件的style前面加上scoped

1. $nextclick的用法？

答：当修改了data的值，然后马上获取这个dom的值，是不能获取到更新后的值。需要使用$nextclick回调，让修改后的data值渲染更新到dom元素之后在获取，才能成功。

1. <https://zhuanlan.zhihu.com/p/92407628>

与后台接口交互部分

1. 前端如何解决跨域？
2. 前端如何封装ajax、aiox

部署

1. 是否使用过nginx部署前端？
2. 使用nginx需要配置什么？
3. 发布后前端包后，浏览器端有缓存怎么解决？

其他

1. 是否做过小程序
2. 是否使用过react
3. 图表用什么？canvas？地图？app?
4. 参与开发什么类型的项目？

最长的开发周期是多少？

# Http部分

1、一次完整的http事务流程

答：1）DNS解析域名

1. 发起TCP三次握手
2. 建立TCP连接后发起http请求
3. 服务器响应http请求，浏览器得到html代码
4. 浏览器解析html代码，并请求html代码中的资源
5. 浏览器对资源进行渲染，呈现给用户
6. TCP连接结束
7. 浏览器解析渲染过程

答：1）浏览器解析html源码，然后创建一个dom树

1. 浏览器解析css代码，然后创建一个csssom树
2. 根据dom树和cssom树，来构建一个渲染树
3. 当渲染树创建完成之后，浏览器就可以根据渲染树直接把页面绘制到屏幕上
4. 渲染树和dom树的区别？

答：1）在dom树构建的同时，浏览器也会构建渲染树。（为了提高用户体验，并不会等，浏览器会先把收到的局部内容渲染出来）

1. dom树完全和html标签一一对应。而渲染树会忽略一些不需要渲染的元素（如：ddisplay：none）
2. 渲染树中每一个节点都存储着对应的css属性。
3. 为什么要先引入css文件，再引入js文件？

答：1）js的下载时阻塞下载，不可以和其他代码并行下载和解析；css的下载不会阻塞dom树的解析

1. 页面加载时，是从上到下，从左右到的顺序加载的。如果将js放到前面，会立即执行。阻塞后续资源的下载，从而导致浏览器“假死”的状态。
2. 浏览器会等header中的css和js加载完后才可以进行绘制。当js放到body后，不需要等待，避免阻塞，是静态页面快速的呈现出来。
3. 部分js是依赖于前面的样式。
4. Js一般是处理功能，所以不需要提前加载。先给用户观感，再给用户上手体验。
5. 为了避免用户长时间看到白屏，应尽可能的提高css的加载速度，具体方法？

答：1）使用cdn加速器

1. 压缩css代码
2. 合理的使用缓存
3. 减少http的访问次数，将多个css合并。
4. 页面性能如何优化？

答：1）减少请求的数量：合并脚本和样式表，css sprites(css 精灵)

1. 缓存：cdn，get请求

3）页面结构：将样式表放在头部，js代码放在底部。

1. 减少重排和重绘（不逐个修改dom样式，预设css的class。不使用table。使用display：none。）