1.vueAPP适配是怎么写的，用的什么单位？

方法一：rem 布局

方法二：lib-flexible 插件实现

2.get和post有什么区别？  
3.计算属性和监听器有什么区别？  
4.v-for渲染列表是key是用来做什么的？  
5.数据请求在生命周期哪一个阶段？  
6.水平居中有哪几种方法？  
7.flex弹性盒模型？  
8.父盒子中子盒子浮动有什么后果？  
9.清除浮动有哪些方法？  
10.定位有哪些属性？相对定位是相对什么定位？  
11.行没元素和行内块元素有什么区别？行内块元素在同一行显示时有默认空隙，如何解决？  
12.给DOM元素绑定事件有哪些方法？  
13.数组里面有哪些遍历方法？es6

**「 CSS篇 」**

1. CSS 盒子模型，绝对定位和相对定位

答：CSS中，盒子模型也叫框模型，它规定了元素框处理元素内容、内边距、边框 和 外边距 的方式。在HTML文档中，每个元素都有盒子模型。

1. 清除浮动，什么时候需要清除浮动，清除浮动都有哪些方法

答：清除(clear)也有4个可能值。最常用的是 both，清楚左右两边的浮动。left 和 right 只能清楚一个方向的浮动。none 是默认值。

**方法一**：添加新的元素 应用 clear：both；

**方法二**：父级div定义 overflow: auto（注意：是父级div也就是这里的 div.outer）

**方法三**： 据说是最高大上的方法 :after 方法：（注意：作用于浮动元素的父亲

**原理**：使用overflow属性来清除浮动有一点需要注意，overflow属性共有三个属性值：hidden,auto,visible。我们可以使用hiddent和auto值来清除浮动，但切记不能使用visible值，如果使用这个值将无法达到清除浮动效果，其他两个值都可以

**浮动的特点:**   
1.浮动的元素，讲向左或者向右浮动，浮动到包围元素的边上，或者上一个浮动元素的边上为止。

2.浮动的元素，不再占用空间，且浮动元素的层级要高于普通元素。

3.浮动的元素，一定是块元素，不管之前是什么元素。

4.如果浮动的元素没有指定宽度的话，浮动后会尽可能变窄，因此浮动元素要指定宽和高。

5.一行的多个元素，要浮动大家一起浮动。

3. 如何保持浮层水平垂直居中

4. position 和 display 的取值和各自的意思和用法

答：position

    1、position属性取值：static(默认)、relative、absolute、fixed、inherit。

    2、postision：static；始终处于文档流给予的位置。看起来好像没有用，但它可以快速取消定位，让top，right，bottom，left的值失效。在切换的时候可以尝试这个方法。

    3、除了static值，在其他三个值的设置下，z-index才会起作用。（确切地说z-index只在定位元素上有效）

    4、position：relative和absolute都可以用于定位，区别在于前者的div还属于正常的文档流，后者已经是脱离了正常文档流，不占据空间位置，不会将父类撑开。定位原点relative是相对于它在正常流中的默认位置偏移，它原本占据的空间任然保留；absolute相对于第一个position属性值不为static的父类。所以设置了position：absolute，其父类的该属性值要注意，而且overflow：hidden也不能乱设置，因为不属于正常文档流，不会占据父类的高度，也就不会有滚动条。

    5、position：fixed 旧版本IE不支持，定位原点相对于浏览器窗口，而且不能变。常用于header，footer，或者一些固定的悬浮div，随滚动条滚动又稳定又流畅，比JS好多了。fixed可以有很多创造性的布局和作用，兼容性是问题。

    6、position：inherit。规定从父类继承position属性的值。但是任何版本的IE都不支持该属性值。

    display

    1、display属性取值：none、inline、inline-block、block、flex、inherit。

    2、display属性规定元素应该生成的框的类型。文档内任何元素都是框，块框或行内框。

    3、display：none和visiability：hidden都可以隐藏div，区别有点像absolute和relative，前者不占据文档的空间，后者还是占据文档的位置。

    4、display：inline和block，又叫行内元素和块级元素。表现出来的区别就是block独占一行，在浏览器中通常垂直布局，可以用margin来控制块级元素之间的间距；而inline以水平方式布局，垂直方向的margin和padding都是无效的，大小跟内容一样，且无法设置宽高。inline就像塑料袋，内容怎么样，就长得怎么样；block就像盒子，有固定的宽和高。

    5、inline-block就介于两者之间。

    6、display: flex 意为"弹性盒布局模型"，用来为盒状模型提供最大的灵活性。任何一个容器都可以指定为flex布局。设为Flex布局以后，子元素的float、clear和vertical-align属性将失效。采用flex布局的元素，称为flex容器。它的所有子元素自动成为容器成员，称为flex项目（flex item）。

    容器的属性：以下6个属性设置在容器上。

        flex-direction    决定主轴的方向（项目的排列方向）。

            row（默认值）：主轴为水平方向，起点在左端。

            row-reverse：主轴为水平方向，起点在右端。

            column：主轴为垂直方向，起点在上沿。

            column-reverse：主轴为垂直方向，起点在下沿。

        flex-wrap   默认情况下，项目都排在一条线上。flex-wrap属性定义，如果一条轴线排不下，如何换行。

            nowrap（默认）：不换行。

            wrap：换行，第一行在上方。

            wrap-reverse：换行，第一行在下方。

        flex-flow   flex-flow属性是flex-direction属性和flex-wrap属性的简写形式，默认值为row nowrap。

        justify-content     定义了项目在主轴上的对齐方式。假设主轴为从左到右。

            flex-start（默认值）：左对齐

            flex-end：右对齐

            center： 居中

            space-between：两端对齐，项目之间的间隔都相等。

            space-around：每个项目两侧的间隔相等。所以，项目之间的间隔比项目与边框的间隔大一倍。

        align-items     定义项目在交叉轴上如何对齐。假设交叉轴从上到下。

            flex-start：交叉轴的起点对齐。

            flex-end：交叉轴的终点对齐。

            center：交叉轴的中点对齐。

            baseline: 项目的第一行文字的基线对齐。

            stretch（默认值）：如果项目未设置高度或设为auto，将占满整个容器的高度。

        align-content   义了多根轴线的对齐方式。如果项目只有一根轴线，该属性不起作用。

            flex-start：与交叉轴的起点对齐。

            flex-end：与交叉轴的终点对齐。

            center：与交叉轴的中点对齐。

            space-between：与交叉轴两端对齐，轴线之间的间隔平均分布。

            space-around：每根轴线两侧的间隔都相等。所以，轴线之间的间隔比轴线与边框的间隔大一倍。

            stretch（默认值）：轴线占满整个交叉轴。

    项目的属性：以下6个属性设置在项目上。

        order           定义项目的排列顺序。数值越小，排列越靠前，默认为0。

        flex-grow       定义项目的放大比例，默认为0，即如果存在剩余空间，也不放大。如果所有项目的flex-grow属性都为1，则它们将等分剩余空间（如果有的话）。如果一个项目的flex-grow属性为2，其他项目都为1，则前者占据的剩余空间将比其他项多一倍。

        flex-shrink     定义了项目的缩小比例，默认为1，即如果空间不足，该项目将缩小。

        flex-basis      定义了在分配多余空间之前，项目占据的主轴空间。浏览器根据这个属性，计算主轴是否有多余空间。它的默认值为auto，即项目的本来大小。可以设为跟width或height属性一样的值（比如350px），则项目将占据固定空间。

        flex        flex属性是flex-grow, flex-shrink 和 flex-basis的简写，默认值为0 1 auto。后两个属性可选。该属性有两个快捷值：auto (1 1 auto) 和 none (0 0 auto)。建议优先使用这个属性，而不是单独写三个分离的属性

        align-self      align-self属性允许单个项目有与其他项目不一样的对齐方式，可覆盖align-items属性。默认值为auto，表示继承父元素的align-items属性，如果没有父元素，则等同于stretch。该属性可能取6个值，除了auto，其他都与align-items属性完全一致

5. 样式的层级关系，选择器优先级，样式冲突，以及抽离样式模块怎么写，说出思路，有无实践经验

答：1、样式的层级关系：一个是权重，另一个就是共用样式和私用样式了，比如说两个ul，它们的子元素除了背景色之外都一样，那可以直接用li {}来定义相同的公用样式，用 .ul\_1 li {} ， .ul\_2 li {} 来定义不相同的样式。可以根据元素之间的差别来选择用哪种方法。推荐用多层级的方式书写css选择器。

2、选择器优先级：（!important>）id选择器>class选择器（属性选择器/伪类选择器）>标签选择器（伪元素选择器） 同类选择符条件下层级越多的优先级越高。优先级就近原则，同权重情况下样式定义最近者为准。载入样式以最后载入的定位为准。

3、样式冲突：   如果同个元素有两个或以上冲突的CSS规则，浏览器有一些基本的规则来决定哪一个非常特殊而胜出。  
        选择器一样的情况下后面的会覆盖前面的属性。  
        使用嵌套选择器时：  
        　　一组嵌套选择器的实际特性可以计算出来。基本的，使用ID选择器的值是100，使用class选择器的值是10，每个html选择器的值是1。它们加起来就可以计算出特性的值。  
        　　p的特性是1（一个html选择器）  
        　　div p的特性是2（两个html选择器）  
        　　.tree的特性是10（1个class选择器）  
        　　div p.tree的特性是1+1+10=12，（两个html选择器，一个class选择器）  
        　　#baobab的特性是100（1个ID选择器）  
        　　body #content .alternative p的特性是112（两个html选择器，一个ID选择器，一个类选择器）

            基本上，一个选择器越多特性，样式冲突的时候将显示它的样式。

4、抽离样式模块  
    因为浏览器的兼容问题，不同浏览器对有些标签的默认值是不同的，如果没对CSS初始化往往会出现浏览器之间的页面显示差异。当然，初始化样式会对SEO有一定的影响，但鱼和熊掌不可兼得，但力求影响最小的情况下初始化。  
        最简单的初始化方法： \* {padding: 0; margin: 0;} （强烈不建议）  
  
  
        body, h1, h2, h3, h4, h5, h6, hr, p, blockquote, dl, dt, dd, ul, ol, li, pre, form, fieldset, legend, button, input, textarea, th, td { margin:0; padding:0; }  
        body, button, input, select, textarea { font:12px/1.5tahoma, arial, \5b8b\4f53; }  
        h1, h2, h3, h4, h5, h6{ font-size:100%; }  
        address, cite, dfn, em, var { font-style:normal; }  
        code, kbd, pre, samp { font-family:couriernew, courier, monospace; }  
        small{ font-size:12px; }  
        ul, ol { list-style:none; }  
        a { text-decoration:none; }  
        a:hover { text-decoration:underline; }  
        sup { vertical-align:text-top; }  
        sub{ vertical-align:text-bottom; }  
        legend { color:#000; }  
        fieldset, img { border:0; }  
        button, input, select, textarea { font-size:100%; }  
        table { border-collapse:collapse; border-spacing:0; }

6. css3动画效果属性，canvas、svg的区别，CSS3中新增伪类举例

①、canvas时h5提供的新的绘图方法

svg已经有了十多年的历史

②、vanvas画图基于像素点，是位图，如果进行放大或缩小会失真

svg基于图形，用html标签描绘形状，放大缩小不会失真

③、canvas需要在js中绘制

svg在html正绘制

④、canvas支持颜色较svg多

⑤、canvas无法对已经绘制的图像进行修改、操作

svg可以获取到标签进行操作

7. px和em和rem的区别，CSS中link 和@import的区别是?

答：em是相对长度单位。相对于当前对象内文本的字体尺寸。如当前对行内文本的字体尺寸未被人为设置，则相对于浏览器的默认字体尺寸。px像素（Pixel）。相对长度单位。像素px是相对于显示器屏幕分辨率而言的。rem是CSS3新增的一个相对单位（root em，根em），这个单位引起了广泛关注。这个单位与em有什么区别呢？区别在于使用rem为元素设定字体大小时，仍然是相对大小，但相对的只是HTML根元素。这个单位可谓集相对大小和绝对大小的优点于一身，通过它既可以做到只修改根元素就成比例地调整所有字体大小，又可以避免字体大小逐层复合的连锁反应。

5. 了解过flex吗?

答: flex是一种典型的弹性布局；它存在的意义就是为盒状模型提供灵活性；在flex中无法使用float，clear，vertical-align这些属性在flex中不起作用。Flex技术的三大组成部分：UI、数据、服务器技术介绍。。从根本上说，Flex技术是表现层解决方案，像所有其他类似技术一样，表现层技术要解决三个基本问题：表现层界面展示和人机交互，客户端数据操作及服务器端数据交互和整合。

为何要使用flex？

1.开发者更习惯使用flex。而设计者更喜欢使用flesh，为了满足更多的程序员的习惯；Flex应运而生，提供了非常简单的mxml界面描述语言给jsp/asp/php程序人员使用，而且mxml更加规范化、标准化。

2.flex可以给开发人员一个标准。

**「 JavaScript 篇 」**

**JavaScript 基础**

1. JavaScript 里有哪些数据类型，解释清楚 null 和 undefined，解释清楚原始数据类型和引用数据类型。比如讲一下 1 和 Number(1)的区别

答：6种数据类型 。null undefined number boolean string object   
null 表示变量指向一个空指针，是有值的这个值为null。  
undefined 表示该变量没有被定义,该变量不存在

2. 将一下 prototype 是什么东西，原型链的理解，什么时候用 prototype

3. 函数里的this什么含义，什么情况下，怎么用。

4. apply和 call 什么含义，什么区别?什么时候用。(我有篇文章 重点分析过)

5. 数组和对象有哪些原生方法，列举一下，分别是什么含义，比如连接两个数组用哪个方法，删除数组的指定项和重新组装数组(操作数据的重点)。

6. 怎样避免全局变量污染?ES5严格模式的作用，ES6箭头函数和ES5普通函数一样吗?

**JavaScript 的面向对象**

1. JS 模块包装格式都用过哪些，CommonJS、AMD、CMD。定义一个JS 模块代码，最精简的格式是怎样。

2. JS 怎么实现一个类。怎么实例化这个类。

3. 理解闭包吗?请讲一讲闭包在实际开发中的作用;闭包建议频繁使用吗?

4. 说一下了解的js 设计模式，解释一下单例、工厂、观察者。

答：单例：任意对象都会单例，不需要特别处理；（单例模式中的单件确保类有且仅有一个）；

工厂：同样形式参数返回不同的实例；

代理：新建类调用老类的接口，包一下；

观察者：就是事件模式比如按钮事件；

发布者，订阅者：发布一个对象/字符串到所有订阅者；

观察者模式就是处理对象间一对多的依赖关系的，当一个对象发生变化，其它依赖他的对象都要得到通知并更新。

抽象工厂的目的是要提供一个创建一系列相关或相互依赖对象的接口，而不需要指定它们具体的类。这种模式可以汽车制造厂所使用的金属冲压设备中找到。这种冲压设备可以制造汽车车身部件。同样的机械用于冲压不同的车型的右边车门、左边车门、右前挡泥板、左前挡泥板和引擎罩等等。通过使用转轮来改变冲压盘，这个机械产生的具体类可以在三分钟内改变。

装饰模式

工厂模式：把实现同一件事情的相同代码放到一个函数中，之后如果用到这个功能就不用再重新写代码，直接调用，也叫函数的封装；减少页面的冗余代码，提高代码的重复利用率；达到开发软件时的低耦合高内聚。

定义：

MVVM（Model-View-ViewModel），是一种基于前端开发的架构模式。

核心是提供对View 和 ViewModel 的双向数据绑定，View和Model之间并没有直接的联系，而是通过ViewModel进行交互，View的变动，自动反映在ViewModel上，反之亦然，这样就保证视图和数据的一致性。

来源：

现在前端，不仅仅是简单的数据展示了，它不仅要管理复杂的数据状态，还要处理移动设备上各种操作行为等等。因此，前端也需要工程化，也需要一个类似于MVC 的框架来管理这些复杂的逻辑，使开发更加高效：

View UI布局，展示数据

Model 管理数据

Controller 响应用户操作，并将 Model 更新到 View 上

前端应用的复杂程度已不同往日，今非昔比。这时前端开发就暴露出了三个痛点问题：

开发者在代码中大量调用相同的DOM API, 处理繁琐，操作冗余，使得代码难以维护。

大量的DOM 操作使页面渲染性能降低，加载速度变慢，影响用户体验。

当 Model 频繁发生变化，开发者需要主动更新到View ；当用户的操作导致Model发生变化，开发者同样需要将变化的数据同步到Model中， 这样的工作不仅繁琐，而且很难维护复杂多变的数据状态。

早期 jQuery 的出现就是为了前端能更简洁的操作DOM 而设计的，但它只解决了第一个问题，另外两个问题始终伴随着前端一直存在。

MVVM 的出现，完美解决了以上三个问题。

MVVM 的三部分：

Model 层代表数据模型，也可以在Model中定义数据修改和操作的业务逻辑；

View 代表UI 组件，它负责将数据模型转化成UI 展现出来，

ViewModel 是一个同步View 和 Model的对象。

View 和 Model 之间并没有直接的联系，而是通过ViewModel进行交互。

ViewModel 通过双向数据绑定把 View 层和 Model 层连接了起来，而View 和 Model 之间的同步工作完全是自动的，无需人为干涉，因此开发者只需关注业务逻辑，不需要手动操作DOM, 不需要关注数据状态的同步问题，复杂的数据状态维护完全由 MVVM 来统一管理。

5. ajax 跨域有哪些方法，jsonp 的原理是什么，如果页面编码和被请求的资源编码不一致如何处理?

**「 开源工具 」**

1)是否了解开源的架构工具 bower、npm、yeoman、gulp、webpack，有无用过，有无写过，一个 npm 的包里的 package.json 具备的必要的字段都有哪些(名称、版本号，依赖)

2)github常用不常用，关注过哪些项目

3)会不会用 ps 扣图，png、jpg、gif 这些图片格式解释一下，分别什么时候用。如何优化图像、图像格式的区别

4)说一下你常用的命令行工具

5)会不会用git，说上来几个命令，说一下git和svn的区别，有没有用git解决过冲突

**「 计算机网络基础 」**

1)说一下HTTP 协议头字段说上来几个，是否尽可能详细的掌握HTTP协议。一次完整的HTTP事务是怎样的一个过程?

2)cookies 是干嘛的，服务器和浏览器之间的 cookies 是怎么传的，httponly 的 cookies 和可读写的 cookie 有什么区别，有无长度限制

请描述一下cookies，sessionStorage和localStorage的区别

3)从敲入 URL 到渲染完成的整个过程，包括 DOM 构建的过程，说的约详细越好。

4)是否了解Web注入攻击，说下原理，最常见的两种攻击(XSS 和 CSRF)了解到什么程度。

5)是否了解公钥加密和私钥加密。如何确保表单提交里的密码字段不被泄露。验证码是干嘛的，是为了解决什么安全问题。

6)编码常识:文件编码、URL 编码、Unicode编码 什么含义。一个gbk编码的页面如何正确引

**「 前端框架 」**

1. 对 MVC、MVVM的理解

答：MVC（Model View Controller 模型-视图-控制器）是一种Web架构的模式。

特点：把业务逻辑、模型数据、用户界面分离开来，让开发者将数据与表现解耦。

**MVC三要素：**

Model（数据模型）。是对客观事物的抽象。

比如知乎Live，Live就是一个模型，可以用Live类来表示。而一个模型通常还带有很多的和业务相关的逻辑，比如添加，更新，获取Live主讲人信息等等，这些组成了模型的方法。对于开发者模型的表示方法非常易懂和清晰，可以通过非常便捷的代码来做CURD操作而无需写一条又一条的SQL语句。

View（视图）。呈现给用户的效果，呈现的内容是基于Model，它也是收集用户输入的地方。

比如看到一篇Live，数据是一个Live.get(live\_id).to\_dict()的结果，效果是通过对应的模板和样式把这个数据展示出来。

Contorller（控制器）。是Model和View之间的沟通者。

因为View中不会对Model作任何操作，Model不会输出任何用于表现的东西，如HTML代码或者某种效果等，它就是点数据。而Contorller用于决定使用哪些Model，对Model执行什么操作，为视图准备哪些数据，是MVC中沟通的桥梁。

MVC的通信是单向的：

浏览器发送请求

Contorller和Model交互获取数据

Contorller调用View

View渲染数据返回

更简单的表达式：V -> C-> M -> C -> V

2) vue、angularjs等 相对于 jQuery在开发上有什么优点?

3)前后分离的思想了解吗?

4)你上一个项目都用到了那些方法优化js的性能?

5)angular的生命周期?

6)说一下你对vue和vuex的使用方法，vue的组件复用机制

**考察学习能力和方法**

1)你每天必须登录的网站(前端技术相关)是什么?

2)前端技术方面看过哪些书，有无笔记，都有哪些收获。

3)收藏了哪些代码片段?有想过开源自己的代码嘛?

4)怎么理解前端技术的大趋势?自己再做哪方面的知识储备?

5)是否了解或精通其他(后端)的编程语言?

6)做项目有没有遇到哪些印象深刻的技术攻关，具体遇到什么问题，怎么找答案的，最后怎么解的。

7)对以后自己的前端职业路线，怎么规划?

**开放性问题(重要)**

这些问题往往决定你是否最终被录用或者等到终轮面试，技术点回答错了不要紧，人脑不是机器，是可以恶补的。

但如果你没有思想和独到的思路，基础挖的再深，可能也打动不了面试官，因为比你基础好的一大堆，但每个人的个性思想却是不同的。

1. 如果需要你加班，你会加吗，抵触吗?

其实你肯定抵触，但你肯定要回答如果项目需要肯定会加。

2. 一个小项目让你自己负责搭建底层一些架构，你能胜任吗?

回答例如:我肯定愿意尝试，并做到最优的选择方案出来。

3. 如果项目拖太久，你情绪低落或者厌烦了怎么调节?

回答就是， 你结合自身挑着好听的说就行，就像聊天。

4. 你建议自己造轮子，还是利用开源的轮子?

回答:根据实际情况而定，如果开源完全满足 可以自己二次开发就好，大大缩短开发周期，

如果实在没有契合度很高的，可以花费几个工作日尝试造轮。