框架问题

1、 Vue 的双向数据绑定原理是什么？

vue.js 是采用数据劫持结合发布者-订阅者模式的方式，通过 Object.defineProperty()来劫持各个属性的setter，getter，在数据变动时发布消息给订阅者，触发相应的监听回调。

具体步骤：

第一步：需要observe 的数据对象进行递归遍历，包括子属性对象的属性，都加上 setter 和 getter，这样的

话，给这个对象的某个值赋值，就会触发 setter，那么就能监听到了数据变化。

第二步：compile 解析模板指令，将模板中的变量替换成数据，然后初始化渲染页面视图，并将每个指令对应的节点绑定更新函数，添加监听数据的订阅者，一旦数据有变动，收到通知，更新视图。

第三步：Watcher 订阅者是 Observer 和Compile 之间通信的桥梁，主要做的事情是:1、在自身实例化时往属性订阅器(dep)里面添加自己 2、自身必须有一个 update()方法 3、待属性变动dep.notice()通知时，能调用自身的update()方法，并触发 Compile 中绑定的回调，则功成身退。第四步：MVVM 作为数据绑定的入口，整合 Observer、Compile 和 Watcher 三者，通过 Observer 来监听自己的model 数据变化，通过 Compile 来解析编译模板指令，最终利用 Watcher 搭起Observer 和Compile 之间的通信桥梁，达到数据变化 -> 视图更新；视图交互变化(input) -> 数据model 变更的双向绑定效果。

2、请详细说下你对 vue 生命周期的理解？

总共分为 8 个阶段创建前/后，载入前/后，更新前/后，销毁前/后。创建前/后：

1、在 beforeCreated 阶段，vue 实例的挂载元素$el 和数据对象 data 都为 undefined，还未初始化。

2、在 created 阶段，vue 实例的数据对象 data 有了，$el 还没有。

3、载入前/后：在 beforeMount 阶段，vue 实例的$el 和 data 都初始化了，但还是挂载之前为虚拟的 dom 节点，data.message 还未替换。

4、在 mounted 阶段，vue 实例挂载完成，data.message 成功渲染。

5、更新前/后：当 data 变化时，会触发 beforeUpdate 和 updated 方法。

6、销毁前/后：在执行 destroy 方法后，对 data 的改变不会再触发周期函数，说明此时vue 实例已经解除了事件监听以及和 dom 的绑定，但是 dom 结构依然存在

3、请说出 vue.cli 项目中 src 目录每个文件夹和文件的用法？

assets 文件夹是放静态资源；

components 是放组件；

router 是定义路由相关的配

置；view 视图；

app.vue 是一个应用主组件；

main.js 是入口文件

4、怎么定义 vue-router 的动态路由？怎么获取传过来的动态参数？

在 router 目录下的index.js 文件中，对path 属性加上/:id。 使用router 对象的params.id 5、

vue-router 有哪几种导航钩子？

1、全局导航钩子：router.beforeEach(to,from,next)，作用：跳转前进行判断拦截。
2、组件内的钩子；
3、单独路由独享组件

6、scss 是什么？在 vue.cli 中的安装使用步骤是？有哪几大特性？

css 的预编译。
使用步骤：
第一步：用 npm 下三个 loader（sass-loader、css-loader、node-sass）
第二步：在 build 目录找到 webpack.base.config.js，在那个 extends 属性中加一个拓
展.scss 第三步：还是在同一个文件，配置一个 module 属性
第四步：然后在组件的style 标签加上 lang 属性 ，例如：lang=”scss”
有哪几大特性:1、可以用变量，例如（$变量名称=值）；2、可以用混合器；3、可以嵌套

7、mint-ui 是什么？怎么使用？说出至少三个组件使用方法？

基于vue 的前端组件库。npm 安装，然后 import 样式和 js，
vue.use（mintUi）全局引入。
在单个组件局部引入：import {Toast} from
‘mint-ui’。组件一：Toast(‘登录成功’);
组件二：mint-header;
组件三：mint-swiper

8、请说下封装 vue 组件的过程？

首先，组件可以提升整个项目的开发效率。能够把页面抽象成多个相对独立的模块，解决了我们传统项目开发：效率低、难维护、复用性等问题。

然后，使用Vue.extend 方法创建一个组件，然后使用 Vue.component 方法注册组件。子组件需要数据，可以在 props 中接受定义。而子组件修改好数据后，想把数据传递给父组件。可以采用emit 方法、

9、vue-loader 是什么？使用它的用途有哪些？

解析.vue 文件的一个加载器，跟 template/js/style 转换成 js 模块。
用途：js 可以写es6、style 样式可以 scss 或 less、template 可以加 jade 等

10、vue.cli 中怎样使用自定义的组件？有遇到过哪些问题吗？

第一步：在components 目录新建你的组件文件（smithButton.vue），script 一定要export default 第二步：在需要用的页面（组件）中导入：import smithButton from '../components/smithButton.vue' 第三步：注入到vue 的子组件的components 属性上面,components:{smithButton}
第四步：在 template 视图view 中使用，<smith-button></smith-button>
问题有：smithButton 命名，使用的时候则 smith-button。

11、说下你对 mvvm 的理解？双向绑定的理解？

mvvm 就是vm 框架视图、m 模型就是用来定义驱动的数据、v 经过数据改变后的 html、vm 就是用来实现双向绑定

双向绑定:一个变了另外一个跟着变了，例如：视图一个绑定了模型的节点有变化，模型对应的值会跟着变

12、请说下具体使用 vue 的理解？

1、使用vue 不必担心布局更改和类名重复导致的 js 重写，因为它是靠数据驱动双向绑定，底层是通过Object.defineProperty() 定义的数据 set、get 函数原理实现。
2、组件化开发，让项目的可拓展性、移植性更好，代码重用性更高，就好像农民工建房子，拿起自己的工具包就可以开工。项目经理坐等收楼就好。

3、单页应用的体验零距离接触安卓原生应用，局部组件更新界面，让用户体验更快速省时。

4、js 的代码无形的规范，团队合作开发代码可阅读性更高。

13、你是怎么认识 vuex 的？

1、vuex 可以理解为一种开发模式或框架。比如PHP 有 thinkphp，java 有 spring 等。

2、通过状态（数据源）集中管理驱动组件的变化（好比 spring 的 IOC 容器对 bean 进行集中管理）。
3、应用级的状态集中放在 store 中； 改变状态的方式是提交 mutations，这是个同步的事物； 异步逻辑应该封装在 action 中。

14、vuex 有哪几种属性？

有五种，分别是 State、 Getter、Mutation 、Action、 Module

15、vuex 的 State 特性是？

1、Vuex 就是一个仓库，仓库里面放了很多对象。其中 state 就是数据源存放地，对应于与一般Vue 对象里面的 data。
2、state 里面存放的数据是响应式的，Vue 组件从 store 中读取数据，若是store 中的数据发生改变，依赖这个数据的组件也会发生更新。

3、它通过mapState 把全局的 state 和 getters 映射到当前组件的 computed 计算属性中。

16、vuex 的 Getter 特性是？

1、getters 可以对 State 进行计算操作，它就是Store 的计算属性
2、 虽然在组件内也可以做计算属性，但是 getters 可以在多组件之间复用
3、 如果一个状态只在一个组件内使用，是可以不用 getters

17、vuex 的 Mutation 特性是？

1、Action 类似于 mutation，不同在于：
2、Action 提交的是 mutation，而不是直接变更状态。
3、Action 可以包含任意异步操作

18、<keep-alive>的作用是什么？

<keep-alive></keep-alive> 包裹动态组件时，会缓存不活动的组件实例，主要用于保留组件状态或避免重新渲染。

19、vue 中 ref 的作用是什么？

ref 被用来给 DOM 元素或子组件注册引用信息。引用信息会根据父组件的 $refs 对象进行注册。如果在普通的DOM元素上使用，引用信息就是元素；如果用在子组件上，引用信息就是组件实例
注意：只要想要在Vue 中直接操作 DOM 元素，就必须用ref 属性进行注册

20、vue 中组件直接的通信是如何实现的？

组件关系可分为父子组件通信、兄弟组件通信。
1、父组件向子组件：通
过props 属性来实现
2、子组件向父组件：
子组件用$emit （）来触发事件，父组件用$on（）来监听子组件的事件。
父组件可以直接在子组件的自定义标签上使用 v-on 来监听子组件触发的自定义事件。
3、兄弟之间的通信：
通过实例一个 vue 实例 Bus 作为媒介，要相互通信的兄弟组件之中都引入 Bus，之后通过分别调用Bus事件触发和监听来实现组件之间的通信和参数传递。

21、你对 Webpack 的认识？

webpack 是收把项目当作一个整体，通过一个给定的的主文件，webpack 将从这个文件开始找到你的项目的所有依赖文件，使用 loaders 处理它们，最后打包成一个或多个浏览器可识别的 js 文件

22、webpack 中的 entry 是做什么的？

entry: 用来写入口文件，它将是整个依赖关系的根。当我们需要多个入口文件的时候，可以把 entry 写成一个对象。

```
var baseConfig = {
        entry: {
             main: './src/index.js'
```

```
        }
    }
```

23、webpack 中的 output 是做什么的？

output： 即使入口文件有多个，但是只有一个输出配置，如果你定义的入口文件有多个，那么我们需要使用占位符来确保输出文件的唯一性。

```
var baseConfig = {
    entry: {
        main: './src/index.js'
    },
    output: {
        filename: '[name].js',
        path: path.resolve('./build')
    }
}
module.exports = baseConfig
```


24、webpack 中的 Loader 的作用是什么？

1、实现对不同格式的文件的处理，比如说将 scss 转换为 css，或者 typescript 转化为 js
2、转换这些文件，从而使其能够被添加到依赖图中这里
介绍几个常用的loader：

babel-loader： 让下一代的 js 文件转换成现代浏览器能够支持的 JS 文件。

babel 有些复杂，所以大多数都会新建一个.babelrc 进行配置

css-loader,style-loader:两个建议配合使用，用来解析 css 文件，能够解释@import,url()如果需要解析 less 就在后面加一个less-loader

file-loader: 生成的文件名就是文件内容的 MD5 哈希值并会保留所引用资源的原始扩展名

url-loader: 功能类似 file-loader,但是文件大小低于指定的限制时，可以返回一个 DataURL 事实上，在使用

less,scss,stylus 这些的时候，npm 会提示你差什么插件，差什么，你就安上就行了

25、webpack 中的 Plugins 的作用是什么？

loaders 负责的是处理源文件的如 css、jsx，一次处理一个文件。而 plugins 并不是直接操作单个文件，它直接对整个构建过程起作用

1、ExtractTextWebpackPlugin: 它会将入口中引用 css 文件，都打包都独立的 css 文件中，而不是内嵌在 js
打包文件中。
2、HtmlWebpackPlugin:依据一个简单的 index.html 模版，生成一个自动引用你打包后的 js 文件的新 index.html。
3、HotModuleReplacementPlugin: 它允许你在修改组件代码时，自动刷新实时预览修改后的结果注意永远不要在生产环境中使用HMR。

26、webpack 中什么是 bundle,什么是 chunk,什么是 module?

1、bundle 是由 webpack 打包出来的文件，
2、chunk 是指 webpack 在进行模块的依赖分析的时候，代码分割出来的代码块。
3、module 是开发中的单个模块。
4、chunk 打包后变成 bundle.

27、webpack-dev-server 和 http 服务器如 nginx 有什么区别?

webpack-dev-server 使用内存来存储 webpack 开发环境下的打包文件，并且可以使用模块热更新，他比传统的 http 服务对开发更加有效。

28、什么是模块热更新？

模块热更新，是 webpack 的一个功能，他可以使得代码通过修改过后，不用刷新浏览器就可以更新。是高级版的自动刷新浏览器。

29、什么是长缓存？在 webpack 中如何做到长缓存优化？

浏览器在用户访问页面的时候，为了加快加载速度，会对用户访问的静态资源进行存储，但是每一次代码升级或者更新，都需要浏览器去下载新的代码，最方便和最简单的更新方式就是引入新的文件名称。在webpack 中， 可以在 output 给出输出的文件制定 chunkhash，并且分离经常更新的代码和框架代码，通过 NamedModulesPlugin 或者 HashedModulesIdsPlugin 使再次打包文件名不变。

30、简单描述下微信小程序的相关文件类型？

微信小程序项目结构主要有一下几个文件类型,如下

1、WXML （WeiXin Markup Language）是框架设计的一套标签语言，结合基础组件、事件系统，可以构建出页面的结构。内部主要是微信自己定义的一套组件。

2、WXSS (WeiXin Style Sheets)是一套样式语言，用于描述 WXML 的组件样式，

3、js 逻辑处理，网络请求

4、json 小程序设置，如页面注册，页面标题及 tabBar。

5、app.json 必须要有这个文件，如果没有这个文件，项目无法运行，因为微信框架把这个作为配置文件入口，整个小程序的全局配置。包括页面注册，网络设置，以及小程序的 window 背景色，配置导航条样式，配置默认标题。

6、app.js 必须要有这个文件，没有也是会报错！但是这个文件创建一下就行 什么都不需要写以后我们可以在这个文件中监听并处理小程序的生命周期函数、声明全局变量。

31、你是怎么封装微信小程序的数据请求的？

1、将所有的接口放在统一的 js 文件中并导出

2、在 app.js 中创建封装请求数据的方法

3、在子页面中调用封装的方法请求数据

32、小程序有哪些参数传值的方法？

1、给 HTML 元素添加 data-*属性来传递我们需要的值，然后通过 e.currentTarget.dataset 或 onload 的

param 参数获取。但 data-名称不能有大写字母和不可以存放对象

2、设置id 的方法标识来传值通过 e.currentTarget.id 获取设置的 id 的值,然后通过设置全局对象的方式来传递数值

3、在 navigator 中添加参数传值

33、简述微信小程序原理？

1、微信小程序采用 JavaScript、WXML、WXSS 三种技术进行开发，从技术讲和现有的前端开发差不多， 但深入挖掘的话却又有所不同。

2、JavaScript：首先 JavaScript 的代码是运行在微信 App 中的，并不是运行在浏览器中，因此一些 H5 技术的应用，需要微信 App 提供对应的 API 支持，而这限制住了 H5 技术的应用，且其不能称为严格的 H5， 可以称其为伪 H5，同理，微信提供的独有的某些 API，H5 也不支持或支持的不是特别好。

3、WXML：WXML 微信自己基于 XML 语法开发的，因此开发时，只能使用微信提供的现有标签，HTML 的标签是无法使用的。
4、WXSS：WXSS 具有 CSS 的大部分特性，但并不是所有的都支持，而且支持哪些，不支持哪些并没有详细的文档。
5、微信的架构，是数据驱动的架构模式，它的 UI 和数据是分离的，所有的页面更新，都需要通过对数据的更改来实现。
6、小程序分为两个部分 webview 和appService。其中 webview 主要用来展现 UI，appService 有来处理业务逻辑、数据及接口调用。它们在两个进程中运行，通过系统层 JSBridge 实现通信，实现 UI 的渲染、事件的处理。

34、小程序的双向绑定和 vue 哪里不一样？

小程序直接 this.data 的属性是不可以同步到视图的，必须调用：
小程序：

```
Page({
  data: {
    items: []
  },
    onLoad:
  function(options)
  { this.setData({
  items: [1,2,3]
  })
  }})
```


Vue：

```
new
Vue({d
ata: {
  items: []
},
mounted () {
  this.items = [1, 2, 3]
}})
```

35、webview 中的页面怎么跳回小程序中？

<web-view/>网页中可使用 JSSDK 1.3.2 提供的接口返回小程序页面。

```
例如：wx.miniProgram.navigateTo({url:
        '/path/to/page'})
```

36、小程序关联微信公众号如何确定用户的唯一性？

使用wx.getUserInfo 方法 withCredentials 为 true 时可获取encryptedData，里面有 union_id。后端需要进行对称解密。

37、小程序如何实现下拉刷新？

用 view 代替 scroll-view,,设置 onPullDownRefresh 函数实现。

1、在 json 文件中配置enablePullDownRefresh 为 true(app.json 中在window 中设置 enablePullDownRefresh,

此效果作用于全局)。

2、在 js 文件中实现 onPullDownRefresh 方法,在网络请求完成后调用 wx.stopPullDownRefresh()来结束下拉刷新。

38、小程序调用后台接口遇到哪些问题？

1、数据的大小有限制，超过范围会直接导致整个小程序崩溃，除非重启小程序；

2、小程序不可以直接渲染文章内容页这类型的 html 文本内容，若需显示要借住插件，但插件渲染会导致页面加载变慢，所以最好在后台对文章内容的 html 进行过滤，后台直接处理批量替换 p 标签 div 标签为 view 标签，然后其它的标签让插件来做，减轻前端的时间。

39、小程序的 wxss 和 css 有哪些不一样的地方？

1、wxss 的图片引入需使用外链地址。

2、没有Body，样式可直接使用 import 导入。

40、分析下微信小程序的优劣势

优势：

1、无需下载，通过搜索和扫一扫就可以打开。

2、良好的用户体验：打开速度快。

3、开发成本要比App 要低。

4、安卓上可以添加到桌面，与原生 App 差不多。
5、为用户提供良好的安全保障。小程序的发布，微信拥有一套严格的审查流程， 不能通过审查的小程序是无法发布到线上的。

劣势：
1、限制较多。页面大小不能超过 1M。不能打开超过 5 个层级的页面。
2、样式单一。小程序的部分组件已经是成型的了，样式不可以修改。例如：幻灯片、导航。
3、推广面窄，不能分享朋友圈，只能通过分享给朋友，附近小程序推广。其中附近小程序也受到微信的限制。
4、依托于微信，无法开发后台管理功能。

41、Angular 中是怎么实现数据双向绑定的？

Angular 实现了双向绑定机制。所谓的双向绑定，无非是从界面的操作能实时反映到数据，数据的变更能实时展现到界面。即数据模型（Module）和视图（View）之间的双向绑定。
脏检查机制。双向数据绑定是 AngularJS 的核心机制之一。当 view 中有任何数据变化时，会更新到 model ，当 model 中数据有变化时，view 也会同步更新，显然，这需要一个监控。
原理就是，Angular 在 scope 模型上设置了一个 监听队列，用来监听数据变化并更新 view 。每次绑定一个东西到 view 上时 AngularJS 就会往 $watch 队列里插入一条 $watch，用来检测它监视的 model 里是否有变化的东西。当浏览器接收到可以被 angular context 处理的事件时，$digest 循环就会触发，遍历所有的 $watch，最后更新 dom。

42、Angular 中 ng-show/ng-hide 与 ng-if 可以用来作什么？有什么区别？

angularJS 中的 ng-show、ng-hide、ng-if 指令都可以用来控制 dom 元素的显示或隐藏。ng-show 和 ng-hide 根据所给表达式的值来显示或隐藏 HTML 元素。当赋值给 ng-show 指令的值为 false 时元素会被隐藏，值为true 时元素会显示。ng-hide 功能类似，使用方式相反。元素的显示或隐藏是通过改变 CSS 的display 属性值来实现的。
ng-if 指令可以根据表达式的值在 DOM 中生成或移除一个元素。如果赋值给 ng-if 的表达式的值是 false，那对应的元素将会从 DOM 中移除，否则生成一个新的元素插入 DOM 中。ng-if 同 ng-show 和ng-hide 指令最本质的区别是，它不是通过 CSS 显示或隐藏 DOM 节点，而是删除或者新增结点。

43、angular 中$rootScope 和 scope 有什么区别？

scope 是 html 和单个 controller 之间的桥梁，数据绑定就靠他了。rootscope 是各个 controller
中 scope 的桥梁。用rootscope 定义的值，可以在各个 controller 中使用。

44、angular 中应用是怎么启动的，有哪几种方式，且怎么实现的，请写出实现代码？

① angular 启动的主要步骤：

（1） 匿名自执行函数，保证 angular.js 加载完后，立即执行其中的代码。

（2） 通过 window.angular.bootstrap 检测是否 angular 被多次启动/angular.js 被多次加载。多次加载，耗时耗力，不值得提倡。在window 对象上绑定一个属性，这就是个全局属性，全局的嘛，就能用来判断是否多次加 载了（自己写 lib 也可以好好利用window 属性哦）。

（3） 绑定 jQuery，即 bindJQuery()：如果用户导入了 jQuery，就用这个导入的外部 jQuery 。否则用angular 内 置的 jQLite。看来 jQuery 已经成为不可或缺的神物。

（4） 发布 ng 的 API，即 publishExternalAPI()。这样我们才能用 angular.module()之类的方

法。

（5） 查找 ng-app，即 angularInit()。ng 的边界就是 ng-app。

② 启动方式有自启动和手启动

自启：angular.moudle(“myApp”).run(function($rootScope){ console.log($rootScope)});

手启：angular.moudle(“myApp”). run(function($rootScope){console.log($rootScope)});

angular.bootstrap().bootstrap(DOM 对象，[‘myApp’]);

注：手动启动的时候，不需要在页面中指定 ng-app 指令；一般使用自启动。对于一个页面中存在多个 ng-app 的情况，第一个 ng-app 会自动启动，其余的需要手动启动才可以，否则，不生效而且会报错，可以忽略这种方法。

45、angular 自定义指令中 restrict 可以怎么样设置，分别是什么意思？Scope 中@，=，& 有什么区别？怎么实现与父级别作用进行交互？

restrict 属性，来决定这个指令是作为标签（E）、属性（A）、属性值（C）、还是注释（M）。

Scope

1 false（默认值）：直接使用父 scope。比较“危险”。

2 true：继承父 scope

3 {} 可以理解成指令内部并没有一个新的 scope，它和指令以外的代码共享同一个 scope。

@：单向绑定，外部 scope 能够影响内部 scope，但反过来不成立

=：双向绑定，外部 scope 和内部 scope 的 model 能够相互改变

&：把内部 scope 的函数的返回值和外部 scope 的任何属性绑定起来

46、不同模块之间可以使用哪些方式实现交互？可以怎么实现优化 angular 性能？以及你对在 angular 中使用的 jquery 的个人看法

不同模块间的通信

1 采用 SharedService, 利用共享的公共服务来实现数据交换。AngularJS 中的 Service 被设计成单例的，这为这一方案，提供来底层的实现可行性，这一方案也是被广泛采用的。

2 利用 AngularJS 提供的事件机制，$rootScope.$broadcast/ $scope.$emit 配合 $on 方法实现。该方案的实现具备一定的局限性，比如：$emit 方法只能向上传递事件，而不能实现向下的事件传播。但是进行合理的搭配组合已经基本够用了。

3 利用浏览器的 SessionStorage 或者 LocalStorage 进行数据交换。由于浏览器提供的这两类本地存储方案都提供了相应的storage 事件，所以我们也可以使用该方案进行数据交换。使用该方案是应该注意及时清理无关数据。

优化性能：

1 官方提倡的,关闭 debug,$compileProvider

```
myApp.config(function ($compileProvider)
{
        $compileProvider.debugInfoEnabled(false);
  });
```

2 使用一次绑定表达式即{{::yourModel}}

3 减少watcher 数量

4 在无限滚动加载中避免使用 ng-repeat,关于解决方法可以参考这篇文章

5 使用性能测试的小工具去挖掘你的 angular 性能问题，我们可以使用简单的 console.time()也可以借助开发者工具以及 Batarang

```
console.time("TimerName");
codeconsole.timeEnd("TimerName");
```


angularJS 的用法更像是面向对象的编程模式。它会要求你定义一个 view model，然后所有的页面变化，是通过改变这个 view model 来实现的。一旦搭建好了这个框框之后，页面的操作会非常爽， 完全不用考虑具体页面怎么变化，怎么去操作 dom 的问题，浏览器兼容性的问题 angularJS 也一并帮你解决了。

而 jquery 的用法，更像是面向过程的编程模式。你在用 jquery 写具体代码的时候，你需要先考虑到你要实现的场景是怎么样的，然后把具体的实现过程用代码表示出来，而且这种实现往往是单向的。也就是说下次你要再进行页面改变的时候，又得用代码实现一边（而 angularJS 则只要吧view model 的数据还原下就可以了）。

47、关于 angular.js 中，ng-repeat 迭代数组的时候，如果数组中有相同值，会有什么问题， 如何解决；

会提示 Duplicates in a repeater are not allowed. 加 track by $index 可解决。当然，也可以 trace by
任何一个普通的值，只要能唯一性标识数组中的每一项即可（建立 dom 和数据之间的关联）。

48、开发 one page 页面，需要有哪些注意事项；

1事件执行效率

大量实现给未来 dom 节点绑定事件，这种绑定的很多封装实现，是利用往 document 绑定事件，然后冒泡到未来节点来的原理。当页面切换后，这些事件并未清理，页面会给新页面再执行类似的事件绑定。当 document 上点击后，执行的冒泡判断会越来越多，不仅影响到冒泡到未来 dom 节点模式的事件响应速度（因为冒泡判断），还会影响到精确绑定到 dom 节点事件的响应速度（因为执行完事件后，如果没有返回 false，会继续冒泡到 document），导致页面的点击响应越来越慢。

解决这类问题，可以往未来 dom 节点中已存在的父节点绑定事件实现冒泡，或将未来 dom 节点的精确绑定事件触发延迟执行。

2第三方插件重复渲染报错

页面切换，将原理的 uploadify flash 从 dom 中清除，后面再次进入需要显示 uploadify 的页面时，因为没有刷新页面，uploadify 再次渲染会报 id 冲突错误。

类似问题可以在页面切换前或新页面渲染前销毁插件对象。

3会话状态变化

单页应用不刷新页面，很多交互操作只是发出 ajax 请求。如果同一浏览器打开两个窗口，在一个窗口进行了帐号退出操作或切换了帐号，另一窗口继续进行操作，因为不会刷新页面，而且基本都是发 ajax 请求获取数据，这样就会出现页面显示的是切换前的帐号数据，而 ajax 获取到的是切换后的帐号的数据。

可以在每次 ajax 请求，都缓存会话 id，然后 ajax 请求前，比较缓存的会话 id 和cookie 中的会话id 是否一致，如果不一致，则表示已经切换帐号，可以刷新当前页面。

49、Angular 和 vue 的优缺点，你是怎么看待的

① Vue：

优点：
1. 简单：官方文档很清晰，比 Angular 简单易学。
2. 快速：异步批处理方式更新 DOM。
3. 组合：用解耦的、可复用的组件组合你的应用程序。
4. 紧凑：~18kb min+gzip，且无依赖。
5. 强大：表达式 & 无需声明依赖的可推导属性 (computed properties)。
6. 对模块友好：可以通过 NPMBower 或 Duo 安装，不强迫你所有的代码都遵循 Angular

的各种 规定，使用场景更加灵活。

缺点： 1. 新生儿：Vue.js 是一个新的项目，没有 angular 那么成熟。

2. 影响度不是很大：google 了一下，有关于 Vue.js 多样性或者说丰富性少于其他一些有名库。

3. 不支持 IE8：

②angularJS：

优点：1. 模板功能强大丰富，自带了极其丰富的angular 指令。

2. 是一个比较完善的前端框架，包含服务，模板，数据双向绑定，模块化，路由， 过滤器， 依赖注入等所有功能；

3. 自定义指令，自定义指令后可以在项目中多次使用。

4. ng 模块化比较大胆的引入了Java 的一些东西（依赖注入），能够很容易的写出可复用的 代码，对于敏捷开发的团队来说非常有帮助。

5. angularjs 是互联网巨人谷歌开发，这也意味着他有一个坚实的基础和社区支持。缺点：1. angular 入门很容易 但深入后概念很多，学习中较难理解.

2. 文档例子非常少，官方的文档基本只写了 api，一个例子都没有， 很多时候具体怎么用都是 google 来的，或直接问misko,angular 的作者.

3. 对 IE6/7 兼容不算特别好，就是可以用 jQuery 自己手写代码解决一些.

4. 指令的应用的最佳实践教程少，angular 其实很灵活，如果不看一些作者的使用原则，很容易 写出 四不像的代码，例如js 中还是像jQuery 的思想有很多dom 操作.

5. DI 依赖注入 如果代码压缩需要显示声明.

。

50、RequireJS 和 seaJs 的区别

相同之处

RequireJS 和 SeaJS 都是模块加载器，倡导的是一种模块化开发理念，核心价值是让 JavaScript 的模块化开发变得更简单自然。

不同之处

两者的区别如下：

② 定位有差异。RequireJS 想成为浏览器端的模块加载器，同时也想成为 Rhino / Node 等环境的模块加载器。SeaJS 则专注于 Web 浏览器端，同时通过 Node 扩展的方式可以很方便跑在
Node 服务器端。

③ 遵循的规范不同。RequireJS 遵循的是 AMD（异步模块定义）规范，SeaJS 遵循的是 CMD （通用模块定义规范。规范的不同，导致了两者 API 的不同。SeaJS 更简洁优雅，更贴近 CommonJS Modules/1.1 和 Node Modules 规范。

④ 社区理念有差异。RequireJS 在尝试让第三方类库修改自身来支持 RequireJS，目前只有少数社区采纳。SeaJS 不强推，采用自主封装的方式来“海纳百川”，目前已有较成熟的封装策略。

⑤ 代码质量有差异。RequireJS 是没有明显的 bug，SeaJS 是明显没有 bug。

⑥ 对调试等的支持有差异。SeaJS 通过插件，可以实现 Fiddler 中自动映射的功能，还可以实现自动 combo 等功能，非常方便。RequireJS 无这方面的支持。

⑦ 插件机制不同。RequireJS 采取的是在源码中预留接口的形式，源码中留有为插件而写的代码。

SeaJS 采取的插件机制则与 javascript 语言以及 Node 的方式一致：开放自身，让插件开发者可直接访问或修改，从而非常灵活，可以实现各种类型的插件。

51、Angular 的架构思想

1、MVVM

指 Model View Controller

2、模块化和依赖注入

模块用于单独的逻辑表示服务，控制器，应用程序等，并保持代码的整洁。我们在单独的 js 文件中定义的模块，并将其命名为按照module.js 文件形式

模块化的好处

1增加了模块的可重用性

2 通过定义模块，实现加载顺序的自定义

3 在单元测试中，不必加载所有的内容

依赖注入是一种软件设计模式,用于处理如何让程序获得其依赖(对象的)引用

3、双向数据绑定

一但建立双向绑定，使用者输入，会由 Angular 自动传到一个变量中，再自动读到所有绑到它的内容，更新它，效果上就是立即的资料同步， 在程式码中修改变量，也会直接反应到呈现的外观上。

4、指令

指令是DOM 元素上的标记，使元素拥有特定的行为。举例来说，静态的 HTML 不知道如何来创建和展现一个日期选择器控件。让 HTML 能识别这个语法，我们需要使用指令。指令通过某种方法来创建一个能够支持日期选择的元素。我们会循序渐进地介绍这是如何实现的。 如果你写过 AngularJS 的应用，那么你一定已经使用过指令，不管你有没有意识到。你肯定已经用过简单的指令，比如 ng-model, ng-repeat, ng-show 等。这些指令都赋予 DOM 元素特定的行为。例如，ng-repeat 重复特定的元素，ng-show 有条件地显示一个元素。如果你想让一个元素支持拖拽，你也需要创建一个指令来实现它。指令背后基本的想法很简单。它通过对元素绑定事件监听或者改变 DOM 而使 HTML 拥有真实的交互性。

52、谈谈模块化的理解

所谓的模块化开发就是封装细节，提供使用接口，彼此之间互不影响，每个模块都是实现某一特定的功能。模块化开发的基础就是函数

53、angular 中 service 服务三种方式是什么？区别是什么？

Factory ：把 service 的方法和数据放在一个对象里，并返回这个对象。

Service：通过构造函数方式创建 service，返回一个实例化对象。

Provider：创建一个可通过 config 配置的 service，$get 中返回的，就是用 factory 创建 service 的

内容

从底层实现上来看，service 调用了 factory，返回其实例；factory 调用了 provider，返回其

$get 中定义的内容。factory 和 service 功能类似，只不过 factory 是普通 function，可以返回任何东西（return 的都可以被访问，所以那些私有变量怎么写，你懂的）；service 是构造器，可以不返回（绑定到 this 的都可以被访问）；provider 是加强版 factory，返回一个可配置的 factory。

54、列举 React 的生命周期

1. 实例化：

getDefaultPro
ps
getInitialSta
te
componentWillMount
render
componentDidMount

2)更新

componentWillReceivePro
ps
shouldComponentUpdate
componentWillUpdate
render
componentDidUpdate

3) 销毁&清理期

componentWillUnmount

55、react 中 state 和 prop 的区别，改变 state 将对页面有什么影响

props 放初始化数据，是一个父组件传递给子组件的数据流，这个数据流可以一直传递到子孙组件， 组件本身不能修改自己的 props。而 state 代表的是一个组件内部自身的状态，改变一个组件自身状态，从语义上来说，就是这个组件内部已经发生变化，有可能需要对此组件以及组件所包含的子孙组件进行重渲染

56、在 react 中如何获取真实 dom 节点

React.findDOMNode(component)，在已经装载的组件中调用获取真实节点，在 render 中调用会报错。

57、props.children 的作用是什么，如何使用？

this.props.children 属性，它表示组件的所有子节点可以使用以下方法遍历：

使用方法：

```
    React.Children.map(this.props.children,
    function (child) { return <li>{child}</li>;
})
```

其他方法

```
    this.props.children.forEach(func
    tion (child) { return
    <li>{child}</li>
})
```

58、如何为 react 组件设置样式

可以使用style 或 className

```
style={{border:"1px solid #333333", background:"red"}}
className="input"
```


59、实现简单的 react 组件

```
  var ReactCmponent =
  React.createClass({ handleClick:
  function() {
 alert(this.refs.input.value);
  },
    render:
    function()
    { return (
      <div>
        <input type="text" ref="input" />
<input type="button" value="Get Input Value" onClick={this.handleClick}
style={{border:"1px solid #333333", background:"red"}} className="input"
/>
      </div>
    );
  }
});
```

```
ReactDOM.render(
  <ReactCmponent />,
  document.getElementById('container')
);
```

60、react 中 value 和 defaultValue 属性的区别是什么

defaultValue 在react 为form 组件里设置的初始值，组件成为非受控组件，设置 defaultValue，可以在页面上自动改变控件的值。
如果设置了value 值，组件变成受控组件，当在页面上改变控件的值的时候，并不会生效，要响应用户的输入值，需为其绑定 onChange 事件，改变组件对应的 state，让页面重新渲染。

```
    getInitialState: function() {
            return {value: 'Hello!'};
    },
    handleChange: function(event) {
            this.setState({value: event.target.value});
    },
    render: function() {
            return (<input type="text" value={this.state.value} onChange={this.handleChange}
     />
     );
    }
```


61、Bootstrap 有什么好处，为什么要用 bootstrap，什么情况用比较合适

Bootstrap 是一个用于快速开发 Web 应用程序和网站的前端框架。Bootstrap 是基于 HTML,CSS、JAVASCRIPT 的。Bootstrap 是由 Twitter 的 Mark Otto 和 Jacob Thornton 开发的。Bootstrap 是
2011 年八月在 GitHub 上发布的开源产品。

为什么使用 Bootstrap?
移动设备优先：自 Bootstrap 3 起，框架包含了贯穿于整个库的移动设备优先的样式。
浏览器支持：所有的主流浏览器都支持 Bootstrap。
容易上手：只要您具备 HTML 和 CSS 的基础知识，您就可以开始学习 Bootstrap。
响应式设计：Bootstrap 的响应式 CSS 能够自适应于台式机、平板电脑和手机。它为开发人员创建接口提供了一个简洁统一的解决方案。
它包含了功能强大的内置组件，易于定制。
它还提供了基于 Web 的定制。
它是开源的。

61、什么是 Bootstrap 网格系统（Grid System）？

Bootstrap 包含了一个响应式的、移动设备优先的、不固定的网格系统，可以随着设备或视口大小的增加而适当地扩展到 12 列。它包含了用于简单的布局选项的预定义类，也包含了用于生成更多语义布局的功能强大的混合类。
响应式网格系统随着屏幕或视口（viewport）尺寸的增加，系统会自动分为最多 12 列。

63、Bootstrap 网格系统（Grid System）的工作原理？

1、行必须放置在 .container class 内，以便获得适当的对齐（alignment）和内边距（padding）。
2、使用行来创建列的水平组。
3、内容应该放置在列内，且唯有列可以是行的直接子元素。
4、预定义的网格类，比如 .row 和.col-xs-4，可用于快速创建网格布局。LESS 混合类可用于更多语义布局。
5、列通过内边距（padding）来创建列内容之间的间隙。该内边距是通过 .rows 上的外边距（margin） 取负，表示第一列和最后一列的行偏移。
6、网格系统是通过指定您想要横跨的十二个可用的列来创建的。例如，要创建三个相等的列，则使用三个 .col-xs-4。

64、Bootstrap 网格系统列与列之间的间隙宽度是多少？

间隙宽度为 30px（一个列的每边分别是 15px）。

65、使用 Bootstrap 创建垂直表单的基本步骤？

1、向父<form>元素添加 role="form"；
2、把标签和控件放在一个带有 class="form-group"的 中，这是获取最佳间距所必需的；
3、向所有的文本元素<input>、<textarea>、<select>添加 class="form-control"。

66、使用 Bootstrap 创建水平表单的基本步骤？

1、向父<form>元素添加 class="form-horizontal"；

2、把标签和控件放在一个带有 class="form-group"的 中；
3、向标签添加 class="control-label"。

67、对于各类尺寸的设备，Bootstrap 设置的 class 前缀分别是什么？

超小设备手机（<768px）：.col-xs-
小型设备平板电脑
（>=768px）:.col-
sm- 中型设备台式
电 脑
（>=992px）:.col-
md- 大型设备台式
电 脑
（>=1200px）:.col
-lg-

68、为什么使用 node？

总结起来 node 有以下几个特点:简单强大，轻量可扩展. 简单体现在 node 使用的是 javascript,json 来进行编码，人人都会；强大体现在非阻塞 IO,可以适应分块传输数据，较慢的网络环境，尤其擅长高并发访问；轻量体现在 node 本身既是代码，又是服务器，前后端使用统一语言;可扩展体现在可以轻松应对多实例，多服务器架构，同时有海量的第三方应用组件.

69、Node 有哪些全局对象？

答：process, console, Buffer 和 exports

70、Node.js 的适用场景？

1) 、实时应用：如在线聊天，实时通知推送等等（如socket.io）
2) 、分布式应用：通过高效的并行 I/O 使用已有的数据
3) 、工具类应用：海量的工具，小到前端压缩部署（如grunt），大到桌面图形界面应用程序
4) 、游戏类应用：游戏领域对实时和并发有很高的要求（如网易的 pomelo 框架）
5) 、利用稳定接口提升Web 渲染能力
6) 、前后端编程语言环境统一：前端开发人员可以非常快速地切入到服务器端的开发（如著名的纯

Javascript 全栈式MEAN 架构）

71. 用过哪些移动端框架和类库

zepto，轻量级的jquery
swiper 滑动，滑屏等效果
fullpage.js 全屏插件
fastclick 加快移动端点击响应时间
animate.css CSS3动画效果库
masonry 瀑布流框架

72. 你项目中vue是如何搭配其他技术使用的？

jQuery，vuex，elementUI，iView，vue-router
1. 主入口页面 index.html 中用 script 标签绝对路径引入，全局可用；
2. webpack中配置plugin
3. npm安装，main.js中引入，Vue.use()

73. computed & watch 的区别

都可以监听页面数据的变化
computed基于它的依赖进行缓存，当依赖变化时才会重新计算，必须有返回值，主要应用于计算商品总价等
watch是侦听一个特定的值，当值变化时来进行一些函数，例如分页组件中监听页码变化，获取对应数据，更换显示内容，还有监听$route变化，解决created生命周期钩子内函数只能执行一次导致的页面数据不刷新的问题。

74. 生命周期钩子有哪些？你在项目中用过哪些？

beforeCreate/created/beforeMount/mounted/beforeUpdate/updated/activated/deactivated/beforeDestroy/destroyed/errorCaptured

75. 虚拟dom怎么实现的?

将DOM做一层映射关系v-DOM，本该对DOM的操作映射到v-DOM上，v-DOM完全用js实现，与宿主浏览器无关，增删改查执行速度很快；
vue中，用js对象表示一个虚拟dom树，用这个树构建一个真实dom树，插入文档，

每次需要更新视图时，会重新构建一个虚拟dom树，并将新树与老树对比，记录差异，把差异应用到第一步构建的真实DOM树上，视图就更新了。

76. Vue.js 定义组件模板的除了单文件还有什么方法吗?

1. 字符串和模板字符串

```
Vue.component('my-checkbox', {
  template: '<div></div>',
});
```

2. X-template

```
Vue.component('my-checkbox', {
  template: '#idName',
});
```

3. 内联模板

html中给标签添加inline-template属性来告诉vue标签中的内容是模板：

```
<my-check inline-template></my-check>
```

4. 渲染函数

77. render函数使用过吗？怎么用的？

通过调用一个方法来生成template模板，通过render函数的参数来传递这个方法，这个方法有三个参数，标签名，属性，内容

78. vue-router 路由组件传参，接收方式？

声明式导航`<router-link :to="/"></router-link>`

编程式导航 `this.$router.push({ path: 'register', query: { plan: 'private' }}) ///register?plan=private`

79. 为什么要用vuex?

当多个状态（数据）分散在不同组件的各个角落，有大量状态（数据）需要相互间传递，如果都放在后台，统一去后台获取，则网络开销比较大，多层嵌套组件间就需要vuex这样的解决方案，公共数据托管在state里，不同组件都可以使用，vuex有一个轻量型替代方案，bus.js。

常用场景：SPA中音乐播放，购物车数据，登录状态，配合localstorage使用

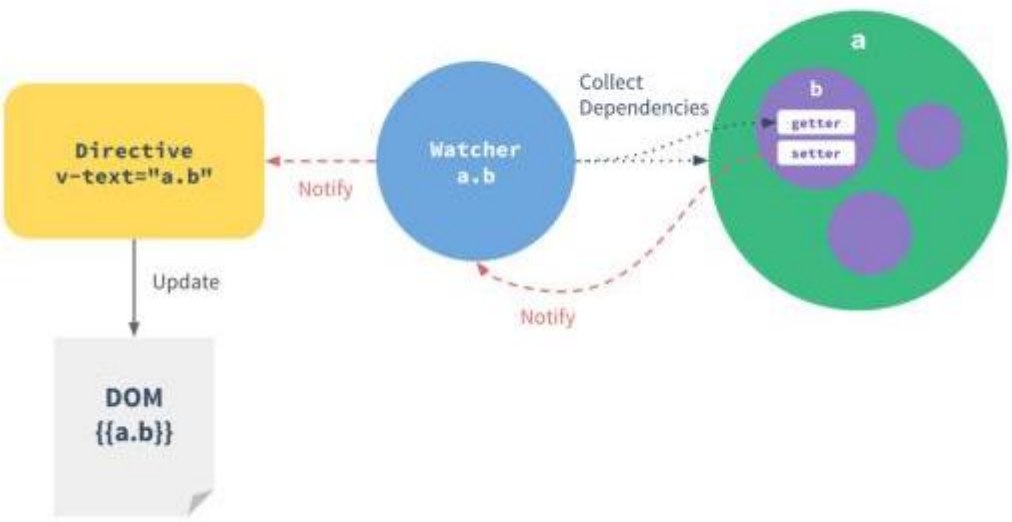
80. 你用过那些webpack插件？

html-webpack-plugin 自动生成html插件，它会在dist目录下自动生成一个

index.html
babel-loader 对ES6语法进行转码
css-loader 对css文件打包
style-loader 将样式添加进dom中
url-loader 图片自动转成base64编码
UglifyJsPlugin //压缩混淆插件
DefinePlugin //决定打成dev包还是production包

81. 数据双向绑定的原理？（好好看看）或者看看129题

三种方式：
发布者-订阅者模式（backbone.js，比较low），脏值检测（angular.js），数据劫持（vue.js）
vue采用数据劫持配合“发布者-订阅者”模式来实现数据双向绑定。
数据变化更新视图，视图变化更新数据，通过数据劫持监听来实现。
1. 设置监听器对view层数据进行劫持监听，发生变化时，通知订阅者们(watcher)更新数据
2. 订阅者收到变化通知并执行相应函数，对数据重新赋值，改变model层
3. 指令解析器对元素节点进行扫描和解析，根据指令模板替换数据



82. 父子组件传值/非父子组件传值

父组件向子组件传值：

1）子组件在props中创建一个属性，用来接收父组件传过来的值；

2）在父组件中注册子组件；

3）在子组件标签中添加子组件props中创建的属性；

4）把需要传给子组件的值赋给该属性

子组件向父组件传值：

1）子组件中需要以某种方式（如点击事件）的方法来触发一个自定义的事件；

2）将需要传的值作为$emit的第二个参数，该值将作为实参传给响应事件的方法；

3）在父组件中注册子组件并在子组件标签上绑定自定义事件的监听。

83. vue指令

v-if，v-show，v-for，v-on，v-bind，v-html，v-model，v-once

84. 做过哪些浏览器兼容性处理

1. 块标签float后，横向设置margin时，IE6margin比设置的大
   解决：将块标签设为行内块
2. 部分浏览器默认图片之间有间距
   解决：横排时使用float给img布局
3. min-height本身是个不兼容的属性
   解决：

```
{
min-height:200px;
height:auto!important
height:200px;
overflow:visible
}
```

4. css hack
   使用hacker时，可以将浏览器分为三类：IE6，IE7和遨游，其他
   IE6认识_和*
   IE7和遨游认识*

85. SPA的优缺点

优点：
分离前后端关注点
用户体验好，不需要加载页面
充分利用缓存，减少服务器压力

第一次请求中加载所有资源，不需要重新加载

缺点：
SEO无法抓取（没有html页面，而且使用路由，不发送请求）
初始时加载比较慢
用户操作前进后退，需要写逻辑来完成，不能用浏览器的返回按钮
页面复杂度，难度提高

86. 为什么要用Babel

1. 可以把ES6转为ES5
2. 可以在VUE中使用JSX语法
3. 可以使用大括号按需加载

99. VUE有哪些好处？
1. 低耦合
2.可重用性
3. 独立开发，前后端关注点分离
4. 数据驱动界面，优化性能

87. VUE自定义组件的步骤

第一步：在components目录新建你的组件文件（indexPage.vue），script一定要export default {}
第二步：在需要用的页面（组件）中导入：import indexPage from '@/components/indexPage.vue'
第三步：注入到vue的子组件的components属性上面,components:{indexPage}
第四步：在template视图view中使用
例如有indexPage命名，使用的时候则index-page

88. VUEX属性

State、 Getter、Mutation 、Action、 Module
vuex的State特性
A、Vuex就是一个仓库，仓库里面放了很多对象。其中state就是数据源存放地，对应于一般Vue对象里面的data
B、state里面存放的数据是响应式的，Vue组件从store中读取数据，若是store中的数据发生改变，依赖这个数据的组件也会发生更新
C、它通过mapState把全局的 state 和 getters 映射到当前组件的 computed 计算属性中
vuex的Getter特性
A、getters 可以对State进行计算操作，它就是Store的计算属性
B、 虽然在组件内也可以做计算属性，但是getters 可以在多组件之间复用

C、 如果一个状态只在一个组件内使用，是可以不用getters
vuex的Mutation特性
Action 类似于 mutation，不同在于：Action 提交的是 mutation，而不是直接变更状态；Action 可以包含任意异步操作。
应用级的状态集中放在store中； 改变状态的方式是提交mutations，这是个同步的； 异步逻辑应该封装在action中。

89. 不使用VUEX会有哪些问题

可维护性下降，修改一个数据要维护多个地方
可读性下降，一个组件里的数据看不出从哪里来的
增加了耦合性，大量的上传派发

90. 你常用生命周期在哪些场景？

答：生命周期钩子的一些使用方法：

- beforecreate：可以在这加个loading事件，在加载实例时触发
- created：初始化完成时的事件写在这里，如在这结束loading事件，异步请求也适宜在这里调用
- mounted：挂载元素，获取到DOM节点
- updated：如果对数据统一处理，在这里写上相应函数
- beforeDestroy：可以做一个确认停止事件的确认框
- nextTick：更新数据后立即操作dom

91. 说一下封装vue组件的过程

使用Vue.extend方法创建一个组件，然后使用Vue.component方法注册组件。子组件需要数据，可以在props中接受定义。而子组件修改好数据后，想把数据传递给父组件。可以采用emit方法。

92. 对VUE的template编译的理解

compile编译器将template转化为AST（抽象语法树），合并option，AST经过generate得到render函数，返回值是VNODE虚拟DOM节点（包括标签名、子节点、文本等）。

93. AJAX状态值

0：(未初始化)还没有调用send()方法。
1：(载入)已经调用send()方法，正在派发请求。
2：(载入完成)send()已经执行完成，已经接收到全部的响应内容。
3：(交互)正在解析响应内容。
4：(完成)响应内容已经解析完成，用户可以调用。

94.常见的HTTP状态码：

200——OK，请求成功
301——Moved Permanently，资源（网页等）被永久转移到其他URL
302——Found，307——Temporary Redirect，临时重定向，请求的文档被临时移动到别处
304——Not Modified，未修改，表示客户端缓存的版本是最近的
401——Unauthorized，请求要求用户的身份认证
403——Forbidden，禁止，服务器理解客户端请求，但是拒绝处理此请求，通常是权限设置所致
404——Not Found，请求的资源（网页等）不存在
500——Internal Server Error——内部服务器错误
502——Bad Gateway，充当网关或代理的服务器从远端服务器接收到了一个无效的请求
504——Gateway Time-out，充当网关或代理的服务器，未及时从远端服务器获取请求

95. 数组去重的所有方式

方法1：双层循环，外层循环元素，内层循环作比较
方法2：利用对象属性不能相同的特点去重
方法3：数组递归去重
方法4:使用ES6的set,一句话搞定:Array.from(new Set([1,1,2,3,3,4,2])) //[1,2,3,4]

96. 自己写过组件么？描述一下（自己总结）

参照答案87和130

97. cookie和session的关系，禁用cookie还能使用session么？

session的id存在cookie中，但session不一定依赖cookie，禁用cookie时，session的id可以通过url传递，或者通过隐藏的表单来传递到服务器

98. 输入url之后发生的事情

1. DNS解析域名对应的IP地址
2. 浏览器发送请求
3. 服务器接收请求，返回html（可能经过压缩的）
4. 浏览器接收，解压缩，开始页面渲染

渲染过程：

1. 解析HTML，构建DOM树
2. 构建View树，将css样式应用的DOM树的节点上（此时关于大小的样式还在内存里，没有应用）
3. 布局计算，递归遍历子节点，根据窗口的实际大小，计算每个View树节点的大小和位置
4. 绘制，顺序：背景色-背景图片-边框-子View树节点-轮廓

99、mvvm和mvc区别？它和其它框架（jquery）的区别是什么？哪些场景适合？

mvc和mvvm其实区别并不大。都是一种设计思想。主要就是mvc中Controller演变成mvvm中的viewModel。mvvm主要解决了mvc中大量的DOM 操作使页面渲染性能降低，加载速度变慢，影响用户体验。

区别：vue数据驱动，通过数据来显示视图层而不是节点操作。

场景：数据操作比较多的场景，更加便捷

100、vue的优点是什么？

- 低耦合。视图（View）可以独立于Model变化和修改，一个ViewModel可以绑定到不同的"View"上，当View变化的时候Model可以不变，当Model变化的时候View也可以不变。

- 可重用性。你可以把一些视图逻辑放在一个ViewModel里面，让很多view重用这段视图逻辑。

- 独立开发。开发人员可以专注于业务逻辑和数据的开发(ViewModel)，设计人员可以专注于页面设计。

- 可测试。界面素来是比较难于测试的，而现在测试可以针对ViewModel来写。

99.vuex是什么？怎么使用？哪种功能场景使用它？

vue框架中状态管理。在main.js引入store，注入。新建一个目录store，…..

export 。场景有：单页应用中，组件之间的状态。音乐播放、登录状态、加入购物车

100.vuex有哪几种属性？

有五种，分别是 State、 Getter、Mutation 、Action、 Module

· vuex的State特性

A、Vuex就是一个仓库，仓库里面放了很多对象。其中state就是数据源存放地，对应于一般Vue对象里面的data

B、state里面存放的数据是响应式的，Vue组件从store中读取数据，若是store中的数据发生改变，依赖这个数据的组件也会发生更新

C、它通过mapState把全局的 state 和 getters 映射到当前组件的 computed 计算属性中

· vuex的Getter特性

A、getters 可以对State进行计算操作，它就是Store的计算属性

B、 虽然在组件内也可以做计算属性，但是getters 可以在多组件之间复用

C、 如果一个状态只在一个组件内使用，是可以不用getters

· vuex的Mutation特性

Action 类似于 mutation，不同在于：Action 提交的是 mutation，而不是直接变更状态；Action 可以包含任意异步操作。

（3）不用Vuex会带来什么问题？

· 可维护性会下降，想修改数据要维护三个地方；

· 可读性会下降，因为一个组件里的数据，根本就看不出来是从哪来的；

· 增加耦合，大量的上传派发，会让耦合性大大增加，本来Vue用Component就是为了减少耦合，现在这么用，和组件化的初衷相背。

101、 v-show和v-if指令的共同点和不同点

- v-show指令是通过修改元素的display的CSS属性让其显示或者隐藏
- v-if指令是直接销毁和重建DOM达到让元素显示和隐藏的效果

102、如何让CSS只在当前组件中起作用

将当前组件的<style>修改为<style scoped>

103、Vue中引入组件的步骤?

1）采用ES6的import ... from ...语法或CommonJS的require()方法引入组件
2）对组件进行注册,代码如下

```
// 注册
Vue.component('my-component', {
    template: '<div>A custom component!</div>'
})
```


3）使用组件<my-component></my-component>

104、指令v-el的作用是什么?

提供一个在页面上已存在的 DOM 元素作为 Vue 实例的挂载目标.可以是 CSS 选择器，也可以是一个 HTMLElement 实例

105、在Vue中使用插件的步骤

- 采用ES6的import ... from ...语法或CommonJSd的require()方法引入插件
- 使用全局方法Vue.use(plugin)使用插件,可以传入一个选项对象Vue.use(MyPlugin, { someOption: true })

如使用懒加载插件：

```
Vue.use(VueLazyload, {
  loading: require('common/image/default.png')
})
```

106、请列举出3个Vue中常用的生命周期钩子函数

· created: 实例已经创建完成之后调用,在这一步,实例已经完成数据观测, 属性和方法的运算, watch/event事件回调. 然而, 挂载阶段还没有开始, $el属性目前还不可见

· mounted: el被新创建的 vm.$el 替换，并挂载到实例上去之后调用该钩子。如果 root 实例挂载了一个文档内元素，当 mounted 被调用时 vm.$el 也在文档内。

· activated: keep-alive组件激活时调用

107、active-class是哪个组件的属性？

vue-router模块的router-link组件。

108、怎么定义vue-router的动态路由以及如何获取传过来的动态参数？

在router目录下的index.js文件中，对path属性加上/:id。
使用router对象的params.id。

109.vue-router有哪几种导航钩子？

三种，一种是全局导航钩子：router.beforeEach(to,from,next)，作用：跳转前进行判断拦截。
第二种：组件内的钩子；
第三种：单独路由独享组件

110、什么是vue生命周期

答： Vue 实例从创建到销毁的过程，就是生命周期。也就是从开始创建、初始化数据、编译模板、挂载Dom→渲染、更新→渲染、卸载等一系列过程，我们称这是 Vue 的生命周期。

111、vue生命周期的作用是什么

答：它的生命周期中有多个事件钩子，让我们在控制整个Vue实例的过程时更容易形成好的逻辑。

112、第一次页面加载会触发哪几个钩子

答：第一次页面加载时会触发 beforeCreate, created, beforeMount, mounted 这几个钩子

113、DOM 渲染在哪个周期中就已经完成

答：DOM 渲染在 mounted 中就已经完成了。

114、说出至少4种vue当中的指令和它的用法？

v-if：判断是否隐藏；v-for：数据循环；v-bind:class：绑定一个属性；v-model：实现双向绑定

115、vue-loader是什么？使用它的用途有哪些？

解析.vue文件的一个加载器。（深入理解见https://www.jb51.net/article/115480.htm）
用途：js可以写es6、style样式可以scss或less、template可以加jade等

根据官网的定义，vue-loader 是 webpack 的一个 loader，用于处理 .vue 文件。其次，使用vue-cli脚手架，作者已经配置好了基本的配置，开箱及用，你需要做的就是npm install 安装下依赖，然后就可以开发业务代码了。当然，如果你想进阶，最好熟悉下vue-loader的具体配置，而不要依赖脚手架

116、scss是什么？在vue.cli中的安装使用步骤是？有哪几大特性？

答：css的预编译。（scss是sass的一个升级版本，完全兼容sass之前的功能，又有了些新增能力，最主要的就是sass是靠缩进表示嵌套关系，scss是花括号）
使用步骤：

第一步：先装css-loader、node-loader、sass-loader等加载器模块
第二步：在build目录找到webpack.base.config.js，在那个extends属性中加一个拓展.scss
第三步：在同一个文件，配置一个module属性
第四步：然后在组件的style标签加上lang属性 ，例如：lang=”scss”

特性:

· 可以用变量，例如（$变量名称=值）；

· 可以用混合器，混入@mixin 可以传变量

· 可以嵌套

继承@extend 不可以传变量，相同样式直接继承，不会造成代码冗余；基类未被继承时，也会被编译成css代码

117、为什么使用key？

当有相同标签名的元素切换时，需要通过 key 特性设置唯一的值来标记以让 Vue 区分它们，否则 Vue 为了效率只会替换相同标签内部的内容。

118、为什么避免 v-if 和 v-for 用在一起

当 Vue 处理指令时，v-for 比 v-if 具有更高的优先级，这意味着 v-if 将分别重复运行于每个 v-for 循环中。通过v-if 移动到容器元素，不会再重复遍历列表中的每个值。取而代之的是，我们只检查它一次，且不会在 v-if 为否的时候运算 v-for。

119、active-class是哪个组件的属性？嵌套路由怎么定义？

答：vue-router模块的router-link组件。
嵌套路由顾名思义就是路由的多层嵌套。 一级路由里面使用children数组配置子路由，就是嵌套路由。

120、v-model是什么？怎么使用？ vue中标签怎么绑定事件？

答：可以实现双向绑定，指令（v-class、v-for、v-if、v-show、v-on）。vue的model层的data属性。绑定事件：<input @click=doLog() />

121、axios是什么？怎么使用？描述使用它实现登录功能的流程？

答：请求后台资源的模块。npm install axios -S装好，然后发送的是跨域，需在配置文件中config/index.js进行设置。后台如果是Tp5则定义一个资源路由。js中使用import进来，然后.get或.post。返回在.then函数中如果成功，失败则是在.catch函数中

Vue.js 1.0 我们常使用 vue-resource (官方ajax库), Vue 2.0 发布后作者宣告不再对 vue-resource 进行更新， 推荐我们使用 axios (基于 Promise 的 HTTP 请求客户端，可同时在浏览器和 node.js 中使用)

122、axios+tp5进阶中，调用axios.post(‘api/user’)是进行的什么操作？axios.put(‘api/user/8')呢？

答：跨域，添加用户操作，更新操作。

123、什么是RESTful API？怎么使用?

答：是一个api的标准，无状态请求。请求的路由地址是固定的，如果是tp5则先路由配置中把资源路由配置好。标准有：.post .put .delete

124、vuex是什么？怎么使用？哪种功能场景使用它？

答：vue框架中状态管理。在main.js引入store，注入。新建了一个目录store，…..export 。场景有：单页应用中，组件之间的状态。音乐播放、登录状态、加入购物车

125、mvvm框架是什么？它和其它框架（jquery）的区别是什么？哪些场景适合？

答：一个model+view+viewModel框架，数据模型model，viewModel连接两个

区别：vue数据驱动，通过数据来显示视图层而不是节点操作。

场景：数据操作比较多的场景，更加便捷

126、自定义指令（v-check、v-focus）的方法有哪些？它有哪些钩子函数？还有哪些钩子函数参数？

答：全局定义指令：在vue对象的directive方法里面有两个参数，一个是指令名称，另外一个是函数。组件内定义指令：directives

钩子函数：bind（绑定事件触发）、inserted(节点插入的时候触发)、update（组件内相关更新）

钩子函数参数：el、binding

127、vue-router是什么？它有哪些组件？

答：vue用来写路由一个插件。router-link、router-view

128、导航钩子有哪些？它们有哪些参数？

答：导航钩子有：a/全局钩子和组件内独享的钩子。b/beforeRouteEnter、afterEnter、beforeRouterUpdate、beforeRouteLeave

参数：有to（去的那个路由）、from（离开的路由）、next（一定要用这个函数才能去到下一个路由，如果不用就拦截）最常用就这几种

129、Vue的双向数据绑定原理是什么？

答：vue.js 是采用数据劫持结合发布者-订阅者模式的方式，通过Object.defineProperty()来劫持各个属性的setter，getter，在数据变动时发布消息给订阅者，触发相应的监听回调。

具体步骤：

第一步：需要observe的数据对象进行递归遍历，包括子属性对象的属性，都加上setter和getter
这样的话，给这个对象的某个值赋值，就会触发setter，那么就能监听到了数据

变化

第二步：compile解析模板指令，将模板中的变量替换成数据，然后初始化渲染页面视图，并将每个指令对应的节点绑定更新函数，添加监听数据的订阅者，一旦数据有变动，收到通知，更新视图

第三步：Watcher订阅者是Observer和Compile之间通信的桥梁，主要做的事情是:
1、在自身实例化时往属性订阅器(dep)里面添加自己
2、自身必须有一个update()方法
3、待属性变动dep.notice()通知时，能调用自身的update()方法，并触发Compile中绑定的回调，则功成身退。

第四步：MVVM作为数据绑定的入口，整合Observer、Compile和Watcher三者，通过Observer来监听自己的model数据变化，通过Compile来解析编译模板指令，最终利用Watcher搭起Observer和Compile之间的通信桥梁，达到数据变化 -> 视图更新；视图交互变化(input) -> 数据model变更的双向绑定效果。

ps：1答案同样适合”vue data是怎么实现的？”此面试题。

130、请说下封装 vue 组件的过程？

答：首先，组件可以提升整个项目的开发效率。能够把页面抽象成多个相对独立的模块，解决了我们传统项目开发：效率低、难维护、复用性等问题。

然后，使用Vue.extend方法创建一个组件，然后使用Vue.component方法注册组件。子组件需要数据，可以在props中接受定义。而子组件修改好数据后，想把数据传递给父组件。可以采用emit方法。

131、你是怎么认识vuex的？

答：vuex可以理解为一种开发模式或框架。比如PHP有thinkphp，java有spring等。通过状态（数据源）集中管理驱动组件的变化（好比spring的IOC容器对bean进行集中管理）。

应用级的状态集中放在store中； 改变状态的方式是提交mutations，这是个同步的事物； 异步逻辑应该封装在action中。
132、请说出vue.cli项目中src目录每个文件夹和文件的用法？

答：assets文件夹是放静态资源；components是放组件；router是定义路由相关的配置;view视图； app.vue是一个应用主组件； main.js是入口文件

132、聊聊你对Vue.js的template编译的理解？

https://www.jianshu.com/p/e1669afa30b8

答：简而言之，就是先转化成AST树，再得到的render函数返回VNode（Vue的虚拟DOM节点）

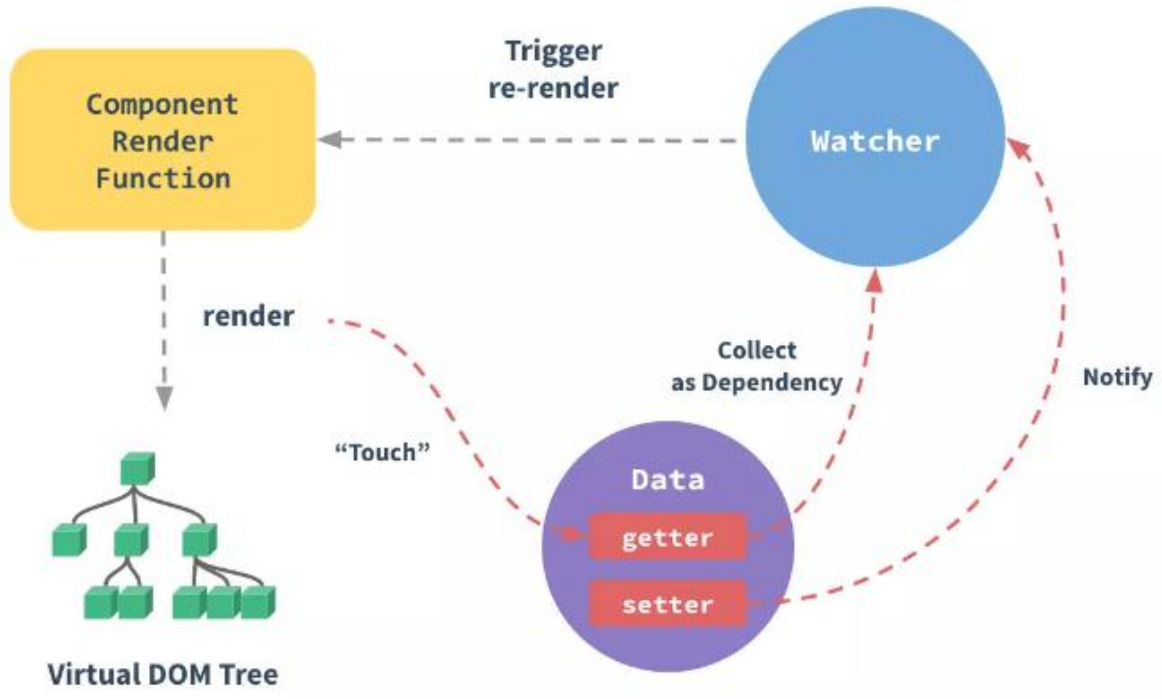
详情步骤：

首先，通过compile编译器把template编译成AST语法树（abstract syntax tree 抽象语法树 即 源代码的抽象语法结构的树状表现形式），compile是createCompiler的返回值，createCompiler是用以创建编译器的。另外compile还负责合并option。

然后，AST会经过generate（将AST语法树转化成render funtion字符串的过程）得到render函数，render的返回值是VNode，VNode是Vue的虚拟DOM节点，里面有（标签名、子节点、文本等等）

133.vue响应式原理？

https://blog.csdn.net/GitChat/article/details/78516752
看这篇https://blog.csdn.net/aaa333qwe/article/details/80093810



有一个数据a.b,在vue对象实例化过程中，会给a,b通过ES5的defineProperty()方法，添加getter和setter方法，同时vue.js会对模板做编译，解析生成一个指令对象，比如v-text指令，每个指令对象都会关联一个watcher，当对指令对象求值时，就会触发getter，并将依赖收集到watcher中；当再次改变a.b值时，就会触发setter方法，会通知到对应关联的watcher,watcher则再次对a.b求值，计算对比新旧值，当值改变时，watcher会通知到指令，调用指令的update方法，由于指令是对dom的封装，所以会调用原生dom的方法，去更新视图。

134、vue-router实现原理？

https://www.cnblogs.com/yanze/p/7644631.html
https://segmentfault.com/a/1190000015123061

135、为什么要选vue？与其它框架对比的优势和劣势？

(1)vue.js更轻量，gzip后大小只有20K+,React gzip后大小为44k，Angular gzip后大小有56k，所以对于移动端来说，vue.js更适合；
(2)vue.js**更易上手，学习曲线平稳，而Angular入门较难，概念较多(比如依赖注入)，它使用java写的**，很多思想沿用了后台的技术，react需学习较多东西，附带react全家桶，
(3)吸收两家之长，借用了angular的指令(比如v-show,v-hide对应angular的ng-show,ng-hide)和react的组件化(将一个页面抽成一个组件，组件具有完整的生命周期)
(4)vue还有自己的特点，比如计算属性

136、vue如何实现父子组件通信，以及非父子组件通信？

http://www.cnblogs.com/sichaoyun/p/6690322.html#4021894