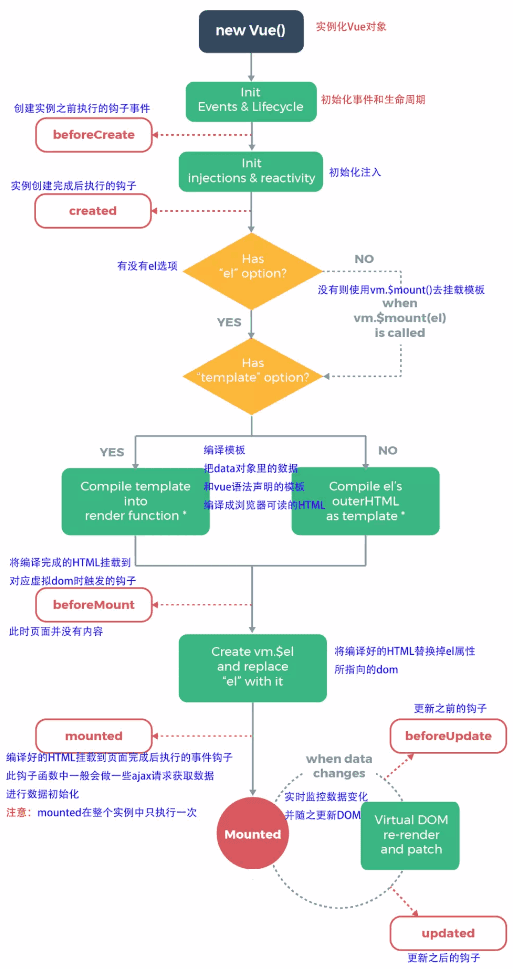
Vue 问题合集



* 1. [你知道vue的模板语法用的是哪个web模板引擎的吗？说说你对这模板引擎的理解](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F561)

印象中是 mustache(胡子)，模板引擎的作用是解决早期的字符模板拼接的问题，也是因这些方面的慢慢的发展壮大。

* 2. [你知道v-model的原理吗？说说看](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F560)

用在表单控件或者组件上创建双向绑定。比如在input 上绑定message，并通过input事件获取当前事件的target.value，并赋值给message。

<input :value="sth" @input="sth = $event.target.value" />



原理：主要是通过Object对象的defineProperty属性，重写data的set和get函数来实现的

* 3. [你有使用过vue开发多语言项目吗？说说你的做法？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F559)

使用插件vue-i18n，支持vue2.x以上

* 4. [在使用计算属性的时，函数名和data数据源中的数据可以同名吗？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F558)

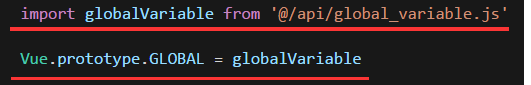
不能同名 因为不管是计算属性还是data还是props 还是methods都会被挂载在vue实例上，因此 这四个都不能同名,同名会报错。

* 5．[vue中data的属性可以和methods中的方法同名吗？为什么？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F557)

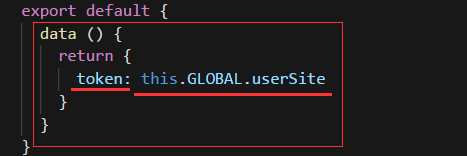
不能同名，Vue会把methods和data的东西，全部挂载到Vue实例中。

* 6. [怎么给vue定义全局的方法？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F556)

变量：

1. 单独新建一个全局变量模块文件，模块中定义一些变量初始状态，用export default 暴露出去。   
2. 在main.js中引入，并通过Vue.prototype挂载到vue实例上面。

供其他模块文件使用（通过this）；

  
3. 或者直接引入到需要的模块文件中使用（先引入，再在data()里面将其挂载到当前组件上）；

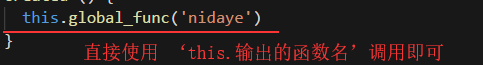
方法：

1. 简单的函数可以直接在main.js里面直接写，即直接挂载在Vue.prototype上

2．**新建一个模块js文件，挂载到main.js上面（全局使用）**







3.局部组件使用，先引入，然后再methods里面将其挂载到当前组件上。

* 7. [vue2.0不再支持v-html中使用过滤器了怎么办？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F555)

Vue2.0的过滤器只能应用在{{ }}和v-bind中，不再支持v-html中使用

1. 使用全局方法，

Vue.prototype. content = function (msg) { return msg.replace("/\n/g", "<br>")};

2.compute属性

computed :{ content：function (msg) { return msg.replace("/\n/g","<br>")}}

3 $options.filters

filters:{ content: function(msg) { return msg.replace(/\n/g, "<br>") ; } }

<div  v-html="$options.filters.content(msg)"></div>

* 8. [怎么解决vue打包后静态资源图片失效的问题？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F554)

解决js，css打包后无法加载

Vue.cli 3.3以上 新建vue.config.js文件并在里面加上

module.exports={

publicPath : ’ ./ ’,

//baseUrl : ‘ ./ ‘, Vue.cli 3.0-3.3

}

Router.js 里面把 mode：‘history’ 注释掉

解决动态img的src打包后无法加载

写绝对路径：https：…..

自己配制webPack 解决js、css路径不对

Config/index.js中设置 assetsPublicPath : ‘ ./ ’

解决本地图片引用问题：

Build/utils.js 里面增加 publicPath : ‘ ../../ ’

* 9. [怎么解决vue动态设置img的src不生效的问题？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F553)

<img :src="require('../../../assets/images/xxx.png')" />

因为动态添加src被当做静态资源处理了，没有进行编译，所以要加上require。

* 10. [使用vue后怎么针对搜索引擎做SEO优化？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F552)

可以肯定的是前后端分离不利于SEO

1.搜索引擎的基础爬虫的原理就是抓取你的url，然后获取你的html源代码并解析。 而你的页面通常用了vue等js的数据绑定机制来展示页面数据，爬虫获取到的html是你的模型页面而不是最终数据的渲染页面，所以说用js来渲染数据对seo并不友好。

2.seo 本质是一个服务器向另一个服务器发起请求，解析请求内容。但一般来说搜索引擎是不会去执行请求到的js的。也就是说，如果一个单页应用，html在服务器端还没有渲染部分数据，在浏览器才渲染出数据，而搜索引擎请求到的html是没有渲染数据的。 这样就很不利于内容被搜索引擎搜索到。 所以服务端渲染就是尽量在服务器发送到浏览器前 页面上就是有数据的。

 3.一般的数据逻辑操作是放在后端的。排序这个如果仅仅是几条数据，前后端排序开起来是一样的，如果是有1000条数据，前端要排序就要都请求过来。这样显然是不合理的。

1. 页面预渲染 (prerender 解决SEO,首屏问题)

由于页面较少，且预渲染相对于SSR比较简单，预渲染可以极大的提高网页访问速度。而且配合一些meat插件，完全可以满足SEO需求。Prerender服务需要有NodeJs环境支持, Prerender需要路由使用history模式，使用hash模式会导致失效。

预渲染的本质就是在打包的时候通过插件将js提前执行一遍，然后再将有内容的html放到服务器上，这样爬虫自然可以抓到，首屏的问题也可以得到解决。

**好处：网络传输数据量小（减少了服务器压力）**

**坏处：前端耗时较多，不利于SEO**

1. 服务端渲染 (SSR) **好处：前端耗时少（前端只负责将html进行展示），利于SEO**

**坏处：网络传输数据量大，占用（部分、少部分）服务器运算资源，response 出的数据量会（稍）大点，模板改了前端的交互和样式什么的一样得跟着联动修改**

3. 路由采用h5 history模式

而应用到的技术也是有很多，大体上我总结了四个，也是比较常用的：

第一，ssr, vue官方文档给出的服务器渲染方案，这是一套完整的构建vue服务端渲染应用的指南

第二，vue-meta-info，这个是针对单页面的meta SEO的另一种思路

第三，nuxt 简单易用，

第四，phantomjs 页面预渲染.

* 11. [跟keep-alive有关的生命周期是哪些？描述下这些生命周期](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F551)
* activated：keep-alive 组件激活时调用。 每次进入页面都触发。即页面第一次进入的时候，钩子触发的顺序created->mounted->activated **该钩子在服务器端渲染期间不被调用。**
* deactivated: keep-alive 组件停用时调用。 即页面退出的时候会触发deactivated， **该钩子在服务器端渲染期间不被调用。**



* 12. [如果现在让你从vue/react/angularjs三个中选择一个，你会选哪个？说说你的理由](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F550)

Vue上手容易些,官方文档清晰好看。其他两个框架不怎么了解，各自都有各自的优点。

* 13. [你知道vue2.0兼容IE哪个版本以上吗？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F549)

IE9及以上

* 14. [使用vue开发一个todo小应用，谈下你的思路](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F548)

首先进行初步的原型设计（用 localstory 存储用户登录信息，创建todo-list，展示todo-list），还可以增加一些其他功能（换肤，选择字体，交互等）。

* 15. [你有看过vue推荐的风格指南吗？列举出你知道的几条](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F547)

看vue.js 官方文档

1使用v-for时记得加key，可以快速定位到需要更新的DOM节点，提高效率。

2永远不要把 v-if 和 v-for 同时用在同一个元素上。提高渲染效率

3组件的prop尽可能描述的越详细越好；它们写明了组件的 API，所以很容易看懂组件的用法；在开发环境下，如果向一个组件提供格式不正确的 prop，Vue 将会告警，以帮助你捕获潜在的错误来源。

4组件名应该始终是多个单词的，根组件 App 以及 <transition> <component> 之类的 Vue 内置组件除外。这样做可以避免跟现有的以及未来的 HTML 元素[相冲突](http://w3c.github.io/webcomponents/spec/custom/#valid-custom-element-name)，因为所有的 HTML 元素名称都是单个单词的。

5组件的 data 必须是一个函数

6为组件设置作用域，style 添加scoped属性，使用唯一的class；

7私有属性名用 $\_

* 16. [你是从vue哪个版本开始用的？你知道1.x和2.x有什么区别吗？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F546)
  1. 在每个组件模板，不在支持片段代码，必须有根元素
  2. 生命周期的改变，原来init ，created，beforeCompile，compiled，ready √ -> mounted，beforeDestroy ，destroyed
  3. 可以自定义键盘指令。以前Vue.directive('on').keyCodes.f1=17;

现在Vue.config.keyCodes.ctrl=17

* 1. 之前:系统就自带很多过滤；到了2.0， 内置过滤器，全部删除了
  2. 之前，子组件可以更改父组件信息，可以是同步 async  
     现在，不允许直接给父级的数据做赋值操作
  3. V-for的参数位置调换。之前:v-for="(index,val) in array"；

现在: v-for="(val,index) in array"

* 17. [你知道vue中key的原理吗？说说你对它的理解](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F545)

便于diff算法的更新，key的唯一性，能让算法更快的找到需要更新的dom。

* 18. [你认为vue的核心是什么？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F289)

Vue的核心是双向绑定和虚拟DOM。

* 19. [什么是虚拟DOM和diff算法？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F227)

vdom是树状结构，其节点是虚拟节点，vnode和浏览器DOM中的Node一一对应。它通过JS的Object对象模拟DOM中的节点，然后再通过特定的渲染方法将其渲染成真实的DOM节点。如果有事件发生修改了虚拟DOM；比较两棵虚拟DOM树的差异，得到差异对象；把差异对象应用到真正的DOM树上。

Diff算法用于比较两颗新旧虚拟DOM树的差异。在采取diff算法比较新旧节点的时候，比较只会在同层级进行, 不会跨层级比较。

当数据发生变化时，vue是怎么更新节点的？

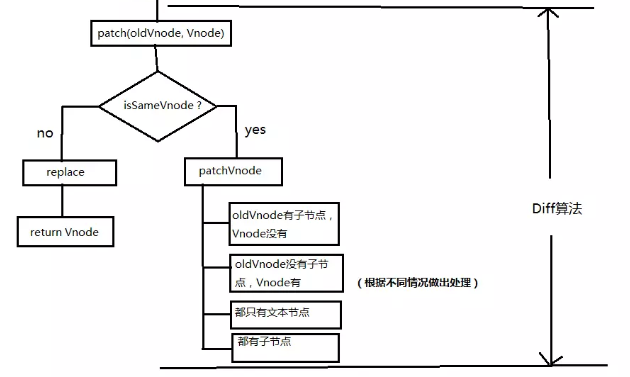
渲染真实DOM的开销是很大的，比如有时候我们修改了某个数据，如果直接渲染到真实dom上会引起整个dom树的重绘和重排。vue先根据真实DOM生成一颗virtual DOM，当virtual DOM某个节点的数据改变后会生成一个新的Vnode，然后Vnode和oldVnode作对比，发现有不一样的地方就直接修改在真实的DOM上，然后使oldVnode的值为Vnode。

diff的过程就是调用名为patch的函数，比较新旧节点，一边比较一边给真实的DOM打补丁。

virtual DOM和真实DOM的区别？

virtual DOM是将真实的DOM的数据抽取出来，以对象的形式模拟树形结构

diff比较过程



相比较新旧两个节点。如果两个节点不一样那就说明Vnode完全被改变了，就可以直接替换oldVnode。如果两个节点都是一样的，那么就深入检查他们的子节点。

判断Vnode和oldVnode是否指向同一个对象，如果是，则不改变真实dom

如果oldVnode有子节点而Vnode没有，则删除真实DOM对应的子节点

如果oldVnode没有子节点而Vnode有，则将Vnode的子节点真实化之后添加到真实dom对应位置上

如果他们都有文本节点并且不相等，那么将真实dom的文本节点设置为Vnode的文本节点。

如果两者都有子节点，则比较子节点，比较方法如下：

将Vnode的子节点Vch和oldVnode的子节点oldCh提取出来

oldCh和vCh各有两个头尾的变量StartIdx和EndIdx，它们的2个变量相互比较，一共有4种比较方式。在比较的过程中，如果是oldS和E匹配上了，那么真实dom中的第一个节点会移到最后；如果是oldE和S匹配上了，那么真实dom中的最后一个节点会移到最前，匹配上的两个指针向中间移动；其它两种匹配上了不用动。然后变量会往中间移动，一旦StartIdx>EndIdx表明oldCh和vCh至少有一个已经遍历完了，就会结束比较。若vCh还有节点，则将剩余的vCh节点根据自己的的index插入到真实dom中去。

如果四种匹配没有一对是成功的，分为两种情况

如果新旧子节点都存在key，那么会根据oldChild的key生成一张hash表，用S的key与hash表做匹配，匹配成功就判断S和匹配节点是否为sameNode，如果是，就在真实dom中将成功的节点移到最前面，否则，将S生成对应的节点插入到dom中对应的oldS位置，S指针向中间移动。

如果没有key,则直接将S生成新的节点在真实DOM的尾部插入（ps：这下可以解释为什么v-for的时候需要设置key了，如果没有key那么就只会做四种匹配，就算指针中间有可复用的节点都不能被复用了）

* 20. [vue中怎么重置data？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F544)

使用Object.assign(this.$data, this.$options.data())，vm.$data可以获取当前状态下的data，vm.$options.data可以获取到组件初始化状态下的data。

* 21. [vue渲染模板时怎么保留模板中的HTML注释呢？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F508)

<template comments> //添加comments选项来保留template的注释;官方强调comments //选项只在完整构建版本中的浏览器内编译时可用

...

</template>



* 22. [Vue.observable你有了解过吗？说说看](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F507)

详情见vue官网

vue2.6发布一个新的API, 它让一个对象可响应。Vue 内部会用它来处理 data 函数返回的对象。返回的对象可以直接用于[渲染函数](https://cn.vuejs.org/v2/guide/render-function.html)和[计算属性](https://cn.vuejs.org/v2/guide/computed.html)内，并且会在发生改变时触发相应的更新。也可以作为最小化的跨组件状态存储器，用于简单的场景。简单来说就是类轻量级vuex，用作状态管理。

* 23. [你知道style加scoped属性的用途和原理吗？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F506)

用途：实现组件样式的私有化，防止全局同名CSS污染  
原理： 给HTML的DOM节点加一个不重复的data-v-469af010属性来标志它的唯一性

然后在对应的每个样式选择器后添加一个与当前组件相同的data-v-469af010属性选择器来私有化样式

如果组件内部还有组件，只会给最外层的组件里的标签加上唯一属性字段，不影响组件内部引用的组件（ 注意 ）

* 24. [你期待vue3.0有什么功能或者改进的地方？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F489)

ts支持, 更易用, 速度提升，双向数据绑定用的proxy,插槽也换成了函数的方式

* 25. [vue边界情况有哪些？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F478)

边界情况指即一些需要对 Vue 的规则做一些小调整的特殊情况。

[访问根实例](https://cn.vuejs.org/v2/guide/components-edge-cases.html#%E8%AE%BF%E9%97%AE%E6%A0%B9%E5%AE%9E%E4%BE%8B)：在每个 new Vue 实例的子组件中，其根实例可以通过this. $root 属性进行访问; 所有的子组件都可以将这个实例作为一个全局的store来访问或者使用; 小型的小项目其实可以直接这样共享数据，大型的项目还是要使用Vuex来管理状态。

[访问父级组件实例](https://cn.vuejs.org/v2/guide/components-edge-cases.html#%E8%AE%BF%E9%97%AE%E7%88%B6%E7%BA%A7%E7%BB%84%E4%BB%B6%E5%AE%9E%E4%BE%8B)：$parent 属性可以用来从一个子组件访问父组件的实例。可以替代将数据以 prop 的方式传入子组件的方式。

访[问子组件实例或子元素](https://cn.vuejs.org/v2/guide/components-edge-cases.html#%E8%AE%BF%E9%97%AE%E5%AD%90%E7%BB%84%E4%BB%B6%E5%AE%9E%E4%BE%8B%E6%88%96%E5%AD%90%E5%85%83%E7%B4%A0)：通过 ref 特性为这个子组件赋予一个 ID 引用。然后在父组件里通过this.$refs.usernameInput访问子组件。$refs 只会在组件渲染完成之后生效，并且它们不是响应式的。应该避免在模板或计算属性中访问 $refs。

依赖注入：通过provide 选项指定我们想要提供给后代组件的数据/方法。然后在任何后代组件里，可以使用 inject 选项来接收指定的我们想要添加在这个实例上的属性inject: ['getMap']。缺点：它将你应用程序中的组件与它们当前的组织方式耦合起来，使重构变得更加困难。同时所提供的属性是非响应式的。

[程序化的事件侦听器](https://cn.vuejs.org/v2/guide/components-edge-cases.html#%E7%A8%8B%E5%BA%8F%E5%8C%96%E7%9A%84%E4%BA%8B%E4%BB%B6%E4%BE%A6%E5%90%AC%E5%99%A8)：用在一个组件实例上手动侦听事件时。

通过 $on(eventName, eventHandler) 侦听一个事件

通过 $once(eventName, eventHandler) 一次性侦听一个事件

通过 $off(eventName, eventHandler) 停止侦听一个事件

循环引用：

组件通过 name 选项可以在它们自己的模板中调用自身的。

[内联模板](https://cn.vuejs.org/v2/guide/components-edge-cases.html#%E5%86%85%E8%81%94%E6%A8%A1%E6%9D%BF)

当 inline-template出现在一个子组件上时，这个组件将会使用其里面的内容作为模板，<my-component inline-template>…</my-component>。

在一个 <script> 元素中，并为其带上 text/x-template 的类型，然后通过一个 id 将模板引用过去。



强制更新：使用[vm.$forceUpdate()](https://cn.vuejs.org/v2/api/#vm-forceUpdate) 注意它仅仅影响实例本身和插入插槽内容的子组件，而不是所有子组件。

[通过 v-once 创建低开销的静态组件](https://cn.vuejs.org/v2/guide/components-edge-cases.html#%E9%80%9A%E8%BF%87-v-once-%E5%88%9B%E5%BB%BA%E4%BD%8E%E5%BC%80%E9%94%80%E7%9A%84%E9%9D%99%E6%80%81%E7%BB%84%E4%BB%B6)

有的时候你可能有一个组件，这个组件包含了大量静态内容。在这种情况下，你可以在根元素上添加 v-once 特性以确保这些内容只计算一次然后缓存起来。

* 26. [如何在子组件中访问父组件的实例？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F477)

$parent 属性可以用来从一个子组件访问父组件的实例。可以替代将数据以 prop 的方式传入子组件的方式。

* 27. [watch的属性用箭头函数定义结果会怎么样？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F476)

 this 是undefind；因为箭头函数默绑定父级作用域的上下文，this 不会是如你所预期的 Vue 实例。不要在选项属性（compute,method,created等）或回调上使用箭头函数，不应该使用箭头函数来定义 method 函数。生命周期钩子的 this 上下文指向调用它的 Vue 实例。

* 28. [在vue项目中如果methods的方法用箭头函数定义结果会怎么样？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F475)

 this 是undefind；因为箭头函数默绑定父级作用域的上下文，this 不会是如你所预期的 Vue 实例。不要在选项属性或回调上使用箭头函数。生命周期钩子的 this 上下文指向调用它的 Vue 实例。

* 29. [在vue项目中如何配置favicon？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F474)

用vue.cli3.x以上配置的项目自动有个public文件夹，里面有favicon.icon.

在webpack里面的htmlWebpackPlugin添加favicon属性；有两处修改，分别是开发环境和生成环境。

* 30. [你有使用过babel-polyfill模块吗？主要是用来做什么的？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F473)

babel默认只转换新的JavaScript句法,而不转换新的API（比如Iterator、Generator、Set、Maps、Proxy、Reflect、Symbol、Promise等全局对象）,如需使用新的API,还需要使用对应的转换插件或者polyfill去模拟这些新特性。

* 31[说说你对vue的错误处理的了解？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F472)

有errorHandler、errorCaptured两个函数。

errorCaptured是组件内部钩子，可捕捉子孙组件抛出的错误，此钩子会收到三个参数：错误对象、发生错误的组件实例以及一个包含错误来源信息的字符串。return false后可以阻止错误继续向上传播。  
errorHandler为全局钩子，使用Vue.config.errorHandler配置，此钩子会收到三个参数：错误对象、vue实例以及一个包含错误来源信息的字符串。2.6后可捕捉v-on与promise链的错误，可用于统一错误处理与错误兜底。

* 32. [在vue事件中传入$event，使用e.target和e.currentTarget有什么区别？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F471)

event.currentTarget指向事件所绑定的元素，而event.target始终指向事件触发时的元素。

* 33. [在.vue文件中style是必须的吗？那script是必须的吗？为什么？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F470)

都不是必须的。如果只有一个<template>,说明是普通组件，是一个静态html。但最好加上<script>部分。

* 34．[vue怎么实现强制刷新组件？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F469)

使用[vm.$forceUpdate()](https://cn.vuejs.org/v2/api/#vm-forceUpdate) 强制刷新。注意它仅仅影响实例本身和插入插槽内容的子组件，而不是所有子组件。

* 35. [vue自定义事件中父组件怎么接收子组件的多个参数？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F468)

这种情况发生在子组件向父组件传值时，父组件可以使用arguments参数接收。Arguments是一个数组。父组件还可以自定义参数。

* 36. [实际工作中，你总结的有哪些？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F467)

项目目录结构清晰

第三方库按需加载

Api的模块化管理，比如将请求后台操作全都封装在一个js文件里

vue组件中写选项的顺序

vuex模块拆分管理

* 37. [vue给组件绑定自定义事件无效怎么解决？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F466)

加.native修饰符，可以在一个组件的根元素上直接监听一个原生事件。但监听一个类似 <input> 的非常特定的元素时，会失效，可以使用$listeners 属性解决这个问题，它是一个对象，里面包含了作用在这个组件上的所有监听器。

* 38. [vue的属性名称与method的方法名称一样时会发生什么问题？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F465)

会报错。因为不管是计算属性还是data还是props 还是methods都会被挂载在vue实例上，因此 这四个都不能同名。优先级prop>data>methods

* 39. [vue变量名如果以\_、$开头的属性会发生什么问题？怎么访问到它们的值？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F464)

以 \_ 或 $ 开头的属性 不会 被 Vue 实例代理，因为它们可能和 Vue 内置的属性、API 方法冲突。可以使用例如 vm.$data.\_property 的方式访问这些属性。

* 40. [vue使用v-for遍历对象时，是按什么顺序遍历的？如何保证顺序？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F463)

在遍历对象时，会按 Object.keys() 的结果遍历，但是不能保证它的结果在不同的 JavaScript 引擎下都一致。

* 41. [vue如果想扩展某个现有的组件时，怎么做呢？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F462)

使用mixins混入

当组件和混入对象含有同名选项时，则以组件数据优先。同名钩子函数将合并为一个数组，因此都将被调用。另外，混入对象的钩子将在组件自身钩子之前调用。

使用Vue.extend扩展

* 42. [说下$attrs和$listeners的使用场景](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F461)

在创建高级别的组件时，用于组件传值，尤其是祖孙组件有跨度的传值。

$attrs--继承所有的父组件属性。子辈可以将未在自己组件内注册的祖辈传递下来的属性（除了prop传递的属性、class 和 style ）通过 v-bind="$attrs" 传递给孙辈。

$listeners包含了父作用域中的 (不含 .native 修饰器的) v-on 事件监听器。它可以通过 v-on="$listeners" 传入内部组件

inheritAttrs：默认值true,父组件传递给子组件的属性，子组件内未被注册的父组件属性（除props的特定绑定）将作为普通的HTML特性应用在子组件的根元素上，如果你不希望组件的根元素继承特性设置inheritAttrs: false,但是class和style会继承。

* 43. [分析下vue项目本地开发完成后部署到服务器后报404是什么原因呢？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F460)

路径不对

* 44. [v-once的使用场景有哪些？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F459)

v-once指只渲染元素和组件一次。随后的重新渲染，元素或组件及其所有的子节点将被视为静态内容并跳过。

场景：[通过 v-once 创建低开销的静态组件](https://cn.vuejs.org/v2/guide/components-edge-cases.html#%E9%80%9A%E8%BF%87-v-once-%E5%88%9B%E5%BB%BA%E4%BD%8E%E5%BC%80%E9%94%80%E7%9A%84%E9%9D%99%E6%80%81%E7%BB%84%E4%BB%B6)

表单提交。可防止用户在请求未及时响应时，多次提交。

* 45. [说说你对vue的表单修饰符.lazy的理解](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F458)

表单三大修饰符 .lazy .number .trim

在默认情况下，v-model 在每次 input 事件触发后将输入框的值与数据进行同步。添加 lazy 修饰符，则变为使用 change 事件进行同步（即失去焦点时触发事件）。

.number 修饰符 可以将返回结果自动转化为 Number 类型，否则就是 String 类型

.trim 修饰符 过滤首尾空格符

* 46. [vue为什么要求组件模板只能有一个根元素？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F457)

一个单文件组件，是一个vue实例，都需要一个根入口，通过这个‘根节点’，来递归遍历整个vue‘树’下的所有节点，并处理为vdom，最后再渲染成真正的HTML，插入在正确的位置。

为什么说一个vue组件就是一个vue实例

vue 组件本身是一个 vue 实例，同时任何 vue 实例都可以注册为 vue 组件。你只是把一个 vue 实例注册成了另一个 vue 实例的组件而已。就像对象的属性也可以是对象一样，迭代关系。

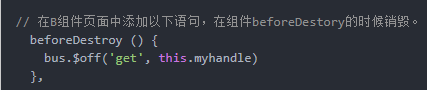
* 47. [EventBus注册在全局上时，路由切换时会重复触发事件，如何解决呢？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F456)

EventBus指Vue中的 **事件总线。** 是所有组件共用相同的事件中心，可以向该中心注册发送事件或接收事件，所以组件都可以上下平行地通知其他组件。EventBus 可以较好的实现兄弟组件之间的数据通讯。<https://www.jianshu.com/p/4fa3bf211785>。

第一次路由跳转未触发的情况：在A组件的beforeDestoryed钩子函数时调用$emit；因为这是B组件的created钩子函数已经执行。<https://www.jianshu.com/p/fde85549e3b0>



重复触发情况：**$on事件是不会自动清除销毁的，需要我们手动来销毁。可以在B组件的**beforeDestoryed（）钩子函数里调用$off（）函数。



* 48. [怎么修改vue打包后生成文件路径？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F455)

Vue.cli 3.3以上 新建vue.config.js文件并在里面加上

module.exports={

publicPath : ’ ./ ’,

//baseUrl : ‘ ./ ‘, Vue.cli 3.0-3.3

}

Router.js 里面把 mode：‘history’ 注释掉

解决动态img的src打包后无法加载

写绝对路径：https：…..

自己配制webPack 解决js、css路径不对

Config/index.js中设置 assetsPublicPath : ‘ ./ ’

解决本地图片引用问题：

Build/utils.js 里面增加 publicPath : ‘ ../../ ’

解决动态添加的img失效问题

<img :src="require('../../../assets/images/xxx.png')" />

因为动态添加src被当做静态资源处理了，没有进行编译，所以要加上require。

* 49. [你有使用做过vue与原生app交互吗？说说vue与ap交互的方法](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F454)

app定义一个方法传给我们，根据方法调用。

JsBridge第三方库的使用。建立连接，然后相互调用。

* 50. [使用vue写一个tab切换](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F453)

v-for循环list，根据索引值设置active的样式和显示内容

* 51. [vue中什么是递归组件？举个例子说明下？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F452)

组件中需要组件调用自身，则为递归组件。递归组件需要声明name，并在组件里自身调用。并且需要使用props进行传参，其还需要一个停止条件，否则会进入死循环。应用：折叠列表，按级划分的选项。

* 52. [怎么访问到子组件的实例或者子元素？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F451)

首先给子组件上添加一个ref属性，然后使用this.$refs访问。注意是非响应的。与v-for一起使用时，返回的是组件实例的数组。

This.$children获取当前实例的直接子组件。是一个数组。需要注意 $children 并不保证顺序，也不是响应式的。

* 53. [在子组件中怎么访问到父组件的实例？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F450)

this.$parent

* 54. [在组件中怎么访问到根实例？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F449)

this.$root

* 55. [说说你对Object.defineProperty的理解](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F448)

Object.defineProperty定义新属性或修改原有的属性；  
vue的数据双向绑定的原理就是用的Object.defineProperty这个方法，里面定义了setter和getter方法，通过观察者模式（发布订阅模式）来监听数据的变化，从而做相应的逻辑处理。

* 56. [vue组件里写的原生addEventListeners监听事件，要手动去销毁吗？为什么？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F447)

（Destoryed钩子函数执行时，组件销毁只会销毁与实例相关的事件，如果将事件绑在文档或文档中其他与实例不相关的节点上是必须取消监听的，因为该事件与实例无关，因此不会自动销毁。）必须手动销毁，不然整个组件的实例都无法被垃圾回收，导致内存泄漏。在beforeDestroy或者destroy钩子函数里面清除监听。

* 57. [vue组件里的定时器要怎么销毁？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F446)

定时器是要销毁的，在beforeDestroy或者destroy钩子函数里面清除定时器。

通过$once来监听定时器，在beforeDestroy钩子函数里可以被清除。 this.$once('hook:beforeDestroy', () => { clearInterval(timer); })。

* 58. [vue组件会在什么时候下被销毁？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F445)

在beforeDestroy之后，在destroy之前。

* 59. [使用vue渲染大量数据时应该怎么优化？说下你的思路！](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F444)

使用 [Object.freeze](https://link.juejin.im/?target=https%3A%2F%2Fdeveloper.mozilla.org%2Fzh-CN%2Fdocs%2FWeb%2FJavaScript%2FReference%2FGlobal_Objects%2FObject%2Ffreeze) 方法，它可以冻结一个对象。使用了 Object.freeze 之后，不仅可以减少 observer 的开销，还能减少不少内存开销。使用方式：this.item = Object.freeze(Object.assign({}, this.item))。缺点是：无法响应。

建议是用“分页”的思路，不一定是几个按钮实际的分页。比如你可以渲染前100条，滚动到离底部一定距离的时候，再渲染100条，正式进入新的100条之后，把第一个100条销毁掉。如果向上滚动到距离顶部一定距离，再逆向一遍。因为实际上你所有的数据都是一次性发过来的，所以这个分页并没有ajax这样比较重的操作，用户的角度来说是感受不到这个渲染-销毁的过程的。

* 60. [在vue中使用this应该注意哪些问题？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F443)

在vue中使用axios做网络请求的时候，会遇到this不指向vue，而为undefined。

1可以使用箭头函数”=>”来解决，

2. bind(this)来改变匿名函数的this指向   
3. hack写法 var \_this= this;：

不要在选项属性或回调上使用箭头函数。this 是undefind

生命周期函数里面的this都是指向vue实例。

* 61. [你有使用过JSX吗？说说你对JSX的理解](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F442)

JSX就是Javascript和XML结合的一种格式。例如const element = <h1>Hello, world!</h1>;它将XML语法直接加入到JavaScript代码中，让你能够高效的通过代码而不是模板来定义界面。之后JSX通过翻译器转换到纯JavaScript再由浏览器执行。在实际开发中，JSX在产品打包阶段都已经编译成纯JavaScript，JSX的语法不会带来任何性能影响。JavaScript的关键字class, for等不能出现在XML中，而使用className, htmlFor代替。

* 62. [说说组件的命名规范](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F441)

[单文件组件](https://cn.vuejs.org/v2/guide/single-file-components.html)的文件名应该要么始终是单词大写开头 (PascalCase)，要么始终是横线连接 (kebab-case)。

**应用特定样式和约定的基础组件 (也就是展示类的、无逻辑的或无状态的组件) 应该全部以一个特定的前缀开头，比如 Base、App 或 V。**

**只应该拥有单个活跃实例的组件应该以 The 前缀命名，以示其唯一性。**这不意味着组件只可用于一个单页面，而是每个页面只使用一次。这些组件永远不接受任何 prop，因为它们是为你的应用定制的，而不是它们在你的应用中的上下文。如果你发现有必要添加 prop，那就表明这实际上是一个可复用的组件，只是目前在每个页面里只使用一次。

**和父组件紧密耦合的子组件应该以父组件名作为前缀命名。**

**组件名应该以高级别的 (通常是一般化描述的) 单词开头，以描述性的修饰词结尾。例如**ClearSearchButton.vue -> SearchButtonClear.vue

**组件名应该倾向于完整单词而不是缩写。**

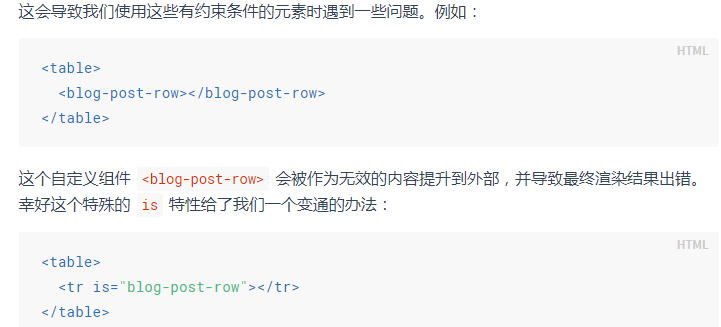
* 63. [怎么配置使vue2.0+支持TypeScript写法？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F440)
* 64. [<template></template>有什么用？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F439)

作为一个字符串模板供 Vue 实例的标识使用。模板将会 **替换** 挂载的元素。挂载元素的内容都将被忽略，除非模板的内容有分发插槽。

当做一个不可见的包裹元素，减少不必要的DOM元素，整个结构会更加清晰。

* 65. [vue的is这个特性你有用过吗？主要用在哪些方面？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F438)

有些 HTML 元素，对于哪些元素可以出现在其内部是有严格限制的，如 <ul>、<ol>、<table> 和 <select>。而有些元素，诸如 <li>、<tr> 和 <option>，只能出现在其它某些特定的元素内部。





vue中is的属性引入是为了解决dom结构中对放入html的元素有限制的问题

用于动态组件，当你多个组件需要通过 v-if 切换时，可以使用 is 来简化代码。

* 66. [vue的:class和:style有几种表示方式？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F437)

这两个都可以与普通class属性共存

1绑定变量

2对象绑定方式 <div v-bind:class="{ active: isActive }"></div>

3绑定数组 数组里面是变量。



4三元表达式

：style有对象绑定和数组绑定

* 67. [你了解什么是函数式组件吗？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F436)

函数式组件是一个不渲染任何html文本的组件。相反，它只管理状态和行为，给父组件或者消费组件暴露一个作用域插槽，以便它们能自己控制该渲染的内容。函数式组件能够准确的渲染你给它传入的内容，无需任何其它元素。

特点：

组件自身是没有状态的

组件自身没有实例，也就是没有this

也没有生命周期方法。

应用场景：

你打算构建一个库，并且希望用户可以自定义组件的外观

在你的项目中有很多功能相似但布局不一样的组件

代码示例：<template function></template>

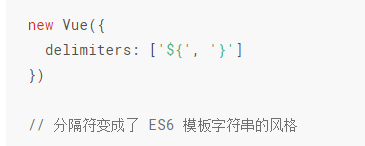
Render函数（createElement，context（this））

createElement函数， 三个参数， 第一个参数是html标签或自定义组件，第二个参数一个obj（包含props， on...等等）， 第三个参数children(通过createElement构建， 或者字符串)



* 68. [vue怎么改变插入模板的分隔符？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F435)

通过delimiters属性



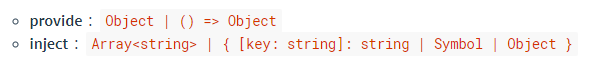
* 69. [组件中写name选项有什么作用？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F434)

使用keep-alive时，可搭配组件name进行缓存过滤 （include 和 exclude 属性）

DOM做递归组件时需要调用自身name

vue-devtools调试工具里显示的组件名称是由vue中组件name决定的

* 70. [说说你对provide和inject的理解](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F433)



这对选项需要一起使用，以允许一个祖先组件向其所有子孙后代注入一个依赖，不论组件层次有多深。provide 是一个对象或返回一个对象的函数。Inject是一个字符串数组，或一个对象。

用于在父组件中通过provide注入一些数据，然后所有子组件中都可以通过inject获取使用该参数传递过来的数据, 主要是为了解决一些循环组件比如tree, menu, list等, 传参困难, 并且难以管理的问题, 主要用于组件封装, 常见于一些ui组件库。

* 71. [开发过程中有使用过devtools吗？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F432)

有，devtools是个很好的插件，大力协助vue项目开发。观察组件，参数，数据等，用于调试vue应用，极大地提高我们的调试效率。

* 72. [说说你对slot的理解有多少？slot使用场景有哪些？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F431)

slot插槽, 在使用组件的时候, 在子组件内部建立插槽，父组件可以向子组件内部指定位置传递内容。<slot>元素作为承载分发内容的出口。  
组件封装的时候最常使用到

* 73. [你有使用过动态组件吗？说说你对它的理解](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F430)

用于在不同组件之间进行动态切换，通过绑定is属性切换组件。并且可以结合keep-live进行缓存。

* 74. [prop验证的type类型有哪几种？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F429)

String、Number、Boolean、Array、Object、Date、Function、Symbol、Promise、任何自定义构造函数、或上述内容组成的数组。

* 75. [prop是怎么做验证的？可以设置默认值吗？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F428)

prop 会在一个组件实例创建**之前**进行验证，（所以实例的属性 (如 data、computed 等) 在 default 或 validator 函数中是不可用的。）

使用type指定类型，使用default设置默认值，require设置必填项，validator设置自定义验证函数。

* 76. [怎么缓存当前打开的路由组件，缓存后想更新当前组件怎么办呢？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F427)

可以在路由meta中加入 meta: {keepAlive: true}参数, 或者使用keep-alive进行缓存,

通过actived钩子函数做些处理。

* 77. [说说你对vue组件的设计原则的理解](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F426)

第一: 容错处理, 这个要做好, 极端场景要考虑到, 不能我传错了一个参数你就原地爆炸  
第二: 缺省值(默认值)要有, 一般把应用较多的设为缺省值  
第三: 颗粒化, 把组件拆分出来.  
第四: 一切皆可配置, 如有必要, 组件里面使用中文标点符号, 还是英文的标点符号, 都要考虑到  
第五: 场景化, 如一个dialog弹出, 还需要根据不同的状态封装成success, waring, 等  
第六: 有详细的文档/注释和变更历史, 能查到来龙去脉, 新版本加了什么功能是因为什么  
第七: 组件名称, 参数prop, emit, 名称设计要通俗易懂, 最好能做到代码即注释这种程度  
第八: 可拓展性, 前期可能不需要这个功能, 但是后期可能会用上, 要预留什么, 要注意什么, 心里要有逼数  
第九: 规范化,我这个input组件, 叫on-change, 我另外一个select组件叫change, 信不信老子捶死你  
第十: 分阶段: 不是什么都要一期开发完成看具体业务, 如果一个select, 我只是个简单的select功能, 什么multi老子这个版本压根不需要, 别TM瞎折腾! 给自己加戏

* 78. [vue如何优化首页的加载速度？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F424)

路由懒加载，首页加载的时候，不让其他页面加载，减缓首页加载压力。还有分屏加载, 按需加载, 延时加载图片、图片进行压缩或者上传到CDN上等。

* 79. [vue打包成最终的文件有哪些？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F423)

Index.html，favicon.ico，js文件，css文件，图片文件夹，字体文件夹

* 80. [ajax、fetch、axios这三都有什么区别？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F422)

ajax核心是使用原生XMLHttpRequest对象,是对其的封装。

Fetch是浏览器提供的webAPI，使用了ES6中的promise对象。是基于promise设计的。

axios是一个基于 promise 的 HTTP 库，原理也是基于xmlHttpRequest的封装，使用更便捷。但它不是原生js，使用时需要对其进行安装。

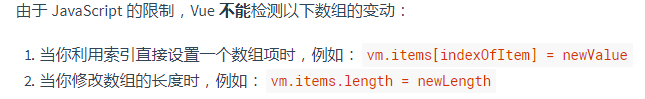
* 81. [vue能监听到数组变化的方法有哪些？为什么这些方法能监听到呢？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F421)push()pop()shift()unshift()splice()sort()reverse()这些方法会改变调用了这些方法的原始数组。Vue会响应的。原理：Vue使用了重写数组原型的方案代替。在新的原型里面拦截了数组的一些方法，并在这个过程中再去做通知变化等操作。

第一步：先获取原生 Array 的原型方法，因为拦截后还是需要原生的方法帮我们实现数组的变化。

第二步：对 Array 的原型方法使用 Object.defineProperty 做一些拦截操作。

第三步：把需要被拦截的 Array 类型的数据原型指向改造后原型。

filter()、concat() 和 slice() 。它们不会改变原始数组，而**总是返回一个新数组**。Vue不能响应，当使用非变异方法时，可以用新数组替换旧数组。



解决1问题可以用 [vm.$set](https://cn.vuejs.org/v2/api/#vm-set) 实例方法; 为了解决第二类问题，你可以使用 splice。

由于 JavaScript 的限制，**Vue 不能检测对象属性的添加或删除**。

对于已经创建的实例，Vue 不允许动态添加根级别的响应式属性。但是，可以使用 Vue.set(object, propertyName, value) 方法向嵌套对象添加响应式属性。还可以使用 vm.$set 实例方法。可能需要为已有对象赋值多个新属性，比如使用 Object.assign() 或 \_.extend()。

Vue响应式原理

Vue响应式原理的核心就是Observer（Object.defineProperty）、Dep、Watcher。

* 82. [vue中是如何使用event对象的？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F420)

event是事件对象。普通的不带调用的绑定函数（v-on:click=‘click’）是可以在方法中拿到event对象的； 而带调用的绑定函数v-on:click="click()" 在方法中是拿不到event对象的（event为undefined），需要以参数的形式穿进去v-on:click="click($event)"。

1. 使用不带圆括号的形式，event 对象将自动当做实参传入；
2. 使用带圆括号的形式，我们需要使用 $event 变量显式传入 event 对象。

* 83. [vue首页白屏是什么问题引起的？如何解决呢？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F366)

**白屏可能是打包后的index.html，js，css文件路径不对，需要以下修改。**

Vue.cli 3.3以上 新建vue.config.js文件并在里面加上

module.exports={

publicPath : ’ ./ ’,

//baseUrl : ‘ ./ ‘, Vue.cli 3.0-3.3

}

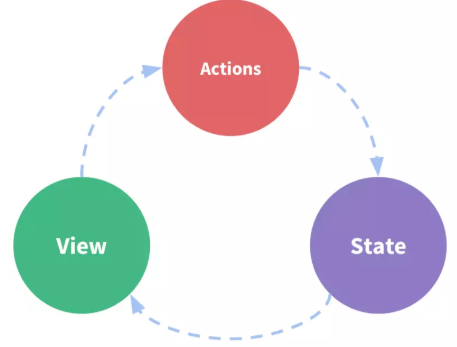
Router.js 里面把 mode：‘history’ 注释掉

自己配制webPack 解决js、css路径不对

Config/index.js中设置 assetsPublicPath : ‘ ./ ’

* 84. [说说你对单向数据流和双向数据流的理解](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F365)

单向数据流



简单的单向数据流（unidirectional data flow）是指用户访问View，View发出用户交互的Action，在Action里对state进行相应更新。state更新后会触发View更新页面的过程。这样数据总是清晰的单向进行流动，便于维护并且可以预测。当我们的应用遇到多个组件共享状态时，**单向数据流的简洁性很容易被破坏**：(1)多个视图依赖于同一状态；(2)来自不同视图的行为需要变更同一状态，影响所有视图。使用全局store来管理，如vuex。

优缺点：好处是所有状态变化都可以被记录、跟踪；状态变化可以通过手动调用通知；源头易追溯，没有“暗箱操作”。同时组件数据只有唯一的入口和出口，使得程序更直观更容易理解，有利于应用的可维护性。缺点则是代码量会相应的上升，数据的流转过程变长，从而出现很多类似的样板代码。同时由于对应用状态独立管理的严格要求(单一的全局store)，在处理局部状态较多的场景时(如用户输入交互较多的“富表单型”应用)，会显得啰嗦及繁琐。

2.双向数据流

双向数据绑定，带来双向数据流。数据（state）和视图（View）之间的双向绑定。

特点：无论数据改变，或是用户操作，都能带来互相的变动，自动更新。适用于项目细节，如：UI控件中(通常是类表单操作)。

优缺点：和单向相反

应用场景

单向绑定跟双向绑定在功能上基本上是互补的，所以我们可以在合适的场景下使用合适的手段。比如在 UI控件 中(通常是类表单操作)，我会使用双向的方式绑定数据；而其他场景则统一采用 **单向**  的方式构建应用。

* 85. [移动端ui你用的是哪个ui库？有遇到过什么问题吗？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F364)

Mui，mint-ui

* 86. [你知道nextTick的原理吗？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F363)

在下次 DOM 更新循环结束之后执行延迟回调(Vue.nextTick(callback))或者vm.$nextTick() 实例方法。在修改数据之后立即使用这个方法，获取更新后的 DOM。

Vue 在更新 DOM 时是**异步**执行的。只要侦听到数据变化，Vue 将开启一个队列，并缓冲在同一事件循环中发生的所有数据变更。如果同一个 watcher 被多次触发，只会被推入到队列中一次。在下一个的事件循环“tick”中，Vue 刷新队列并执行实际 (已去重的) 工作。Vue 在内部对异步队列尝试使用原生的 Promise.then、MutationObserver 和 setImmediate，如果执行环境不支持，则会用 setTimeout(fn, 0) 代替。

简单来说，就是当数据发生变化时，视图不会立即更新，而是等到同一事件循环中所有数据变化完成之后，再统一更新视图。

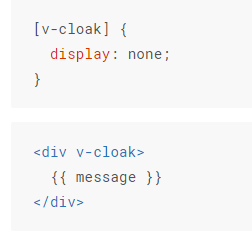
$nextTick() 返回一个 Promise 对象

$nextTick 触发的时机：同一事件循环中的代码执行完毕 -> DOM 更新 -> nextTick callback触发

* 87. [说说你对v-cloak和v-pre指令的理解](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F362)

v-pre：跳过这个元素和它的子元素的编译过程（<span v-pre>{{ this will not be compiled }}</span>）。有v-pre指令的标签内部的内容不会被编译，会原样输出。即显示原始 Mustache 标签。跳过大量没有指令的节点会加快编译。

v-cloak: 只是在标签中加入一个v-cloak自定义属性，在HTML还编译完成之后该属性会被删除。防止将{{message}}类似的输出到页面。



* 88. [写出你知道的表单修饰符和事件修饰符](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F361)

事件修饰符：

.stop(阻止事件继续传播)

.prevent(阻止事件的默认行为)

.capture(使用事件捕获)

.self(只当在 event.target 是当前元素自身时触发处理函数)

.once(事件只触发一次)

.passive 滚动事件的默认行为 (即滚动行为) 将会立即触发,而不会等待 `onScroll` 完成,这其中包含 `event.preventDefault()` 的情况

<div v-on:scroll.passive="onScroll">...</div>

注意：

不要把 .passive 和 .prevent 一起使用，因为 .prevent 将会被忽略，

使用修饰符时，顺序很重要；相应的代码会以同样的顺序产生。因此，用 v-on:click.prevent.self 会阻止**所有的点击**，而 v-on:click.self.prevent 只会阻止对元素自身的点击。

表单修饰符：

.lazy(将v-model绑定的input事件转为change事件（即失去焦点时触发数据同步）)

.number(将用户的输入值转为数值类型(内部调用parseFloat(),无法被解析时则会返回原始的值))

.trim(自动过滤用户输入的首尾空白字符)

* 89. [说说你对proxy的理解](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F359)

Proxy可以译为代理器。用于修改某些操作的默认行为，可以看做是在目标对象前架设了一层拦截层，外界对该对象的访问都要经过这层拦截。因此提供了一种对外界的访问进行过滤和改写的机制。有13个基本的API（get, set, construct, has, apply, defineProperty, deleteProperty, getPrototypeOf, setPrototypeOf, isExtensible, preventExtensions, ownKeys, getOwnPropertyDescriptor）

* 90. [用vue怎么实现一个换肤的功能？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F357)

方法一:用vuex存储对应颜色信息的id, 然后通过绑定类型的形式切换样式。

方法二：用less函数传参的方式来改变颜色值，不同的颜色对应不同的类名，然后将类名绑定在根组件上（app组件）

方法三：用vuex存储对应颜色信息，然后直接通过内联样式绑定在元素上

* 91. [你有自己用vue写过UI组件库吗？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F358)

没有

* 92. [有在vue中使用过echarts吗？踩过哪些坑？如何解决的？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F356)

没有用过。只知道ECharts提供了功能丰富的图表,  还提供了[自定义图表系列](https://echarts.baidu.com/option.html#series-custom)。

* 93. [如果让你教一个2-3年经验前端经验的同事使用vue，你该怎么教？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F355)

一起开发一个项目，让他看官方文档，不懂的地方一起交流

* 94. [vue性能的优化的方法有哪些？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F354)

首页优化：路由懒加载，首页加载的时候，不让其他页面加载，减缓首页加载压力。还有分屏加载, 按需加载, 延时加载图片、图片进行压缩或者上传到CDN上等。

后台传入大数据：利用Object.freeze()提升性能，使用分页懒加载，预加载。

其他：对项目代码中的JS/CSS/SVG(\*.ico)文件进行gzip压缩，SSR(服务端渲染)

v-show，v-if 用哪个？

不要在模板里面写过多的表达式与判断，不利于维护

v-for遍历时添加 key；避免 v-if 和 v-for 同时用在一个元素上

* 95. [SSR解决了什么问题？有做过SSR吗？你是怎么做的？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F353)

SSR服务端渲染，解决SEO问题，优化首屏加载速度，减少白屏时间。没有做过。

* 96. [说说你觉得认为的vue开发规范有哪些？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F352)

命名规范，结构化规范，注释规范，编码规范，CSS 规范。

目的：提高团队协作效率，便于后台人员添加功能及前端后期优化维护，输出高质量的文档。

* 97. [vue部署上线前需要做哪些准备工作？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F350)

设置router为hash

是否需要配置nginx ，cdn

静态资源 需要require

解决js，css打包后无法加载

Vue.cli 3.3以上 新建vue.config.js文件并在里面加上

module.exports={

publicPath : ’ ./ ’,

//baseUrl : ‘ ./ ‘, Vue.cli 3.0-3.3

}

Router.js 里面把 mode：‘history’ 注释掉

解决动态img的src打包后无法加载

写绝对路径：https：…..

自己配制webPack 解决js、css路径不对

Config/index.js中设置 assetsPublicPath : ‘ ./ ’

解决本地图片引用问题：

Build/utils.js 里面增加 publicPath : ‘ ../../ ’

* 98. [vue过渡动画实现的方式有哪些？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F349)

1使用vue的transition标签结合css样式完成动画

v-enter v-enter-active v-enter-to v-leave v-leave-active v-leave-to  
2.利用animate.css结合transition标签实现动画

enter-class

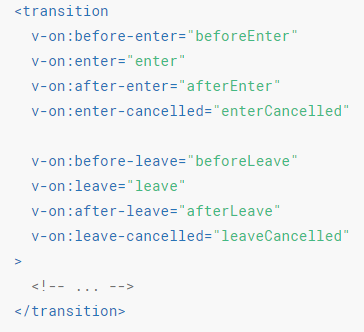
enter-active-class

enter-to-class (2.1.8+)

leave-class

leave-active-class

leave-to-class (2.1.8+)  
3.利用 vue中的钩子函数实现动画



* 99. [vue在created和mounted这两个生命周期中请求数据有什么区别呢？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F348)

Created生命周期只有vue实例和数据，然而，挂载阶段还没开始，页面视图未出现，如果请求信息过多，页面会长时间处于白屏状态。通常在这里面初始化某些属性值。

而mounted在模板渲染成html后调用，通常是初始化页面完成后，再对html的dom节点进行一些需要的操作。

* 100. [vue父子组件双向绑定的方法有哪些？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F347)

利用对象的引用关系来实现（原理是v-mode的使用）  
父子组件之间的数据传递(update:name)  
使用.sync修饰符（对上一步的优化）

* 101. [vue怎么获取DOM节点？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F346)

在mounted钩子函数里面通过原生js方法获取Dom结点。

或者 this.$refs.xxx 获取 如果在普通的 DOM 元素上使用，引用指向的就是 DOM 元素；如果用在子组件上，引用就指向组件实例。

* 102. [vue项目有做过单元测试吗？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F345)

没有 <https://segmentfault.com/a/1190000014112447?utm_source=channel-newest>

单元测试是针对 **程序的最小单元** 来进行正确性检验的测试工作

优点：

1单元测试能避免出现一些代码运行结果与预期不符的错误，通常是一些比较低级但又难以发现的问题。

2单元测试能够避免在升级更新、修复 BUG 的时候引入一些意料之外的问题。

3单元测试对提高代码质量很有帮助。因为，好的代码一般是便于测试的。如果在进行单元测试过程中发现自己的一些代码不方便进行测试，那么你可能需要重新审视这些代码，看是否有一些设计上不合理或者可以优化的地方

* 103.[vue项目有使用过npm run build --report吗？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F344)

没有 vue中可以通过npm run build --report来调用webpack-bundle-analyzer插件

* 104. [如何解决vue打包vendor过大的问题？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F343)

**通过路由的懒加载或者**在webpack.base.conf.js新增externals配置，表示不需要打包的文件，然后在index.html中通过CDN引入

105 [webpack打包vue速度太慢怎么办？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F342)

<https://www.cnblogs.com/imwtr/p/9189670.html>

* 106. [vue在开发过程中要同时跟N个不同的后端人员联调接口（请求的url不一样）时你该怎么办？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F341)

相当于跨域，如果你的前端应用和后端 API 服务器没有运行在同一个主机上，你需要在开发环境下将 API 请求代理到 API 服务器。这个问题可以通过 vue.config.js 中的 devServer.proxy 选项来配置。参考[http-proxy-middleware](https://github.com/chimurai/http-proxy-middleware#proxycontext-config)

module.exports = {

1. devServer: {
2. host: 'localhost',
3. port: 8080,
4. proxy: {
5. '/api': {
6. target: 'https://xxx.com',// 要跨域的域名
7. changeOrigin: true, // 是否开启跨域
8. ws:true,
9. pathRewrite: {
10. '^/api': '/api/' // 在请求的时候 凡是/api开头的地址都可以跨域
11. }
12. },
13. }
14. }

* 107. [vue要做权限管理该怎么做？如果控制到按钮级别的权限怎么做？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F340)

**定义一个全局方法,配合v-if实现。**即在用户登录成功后,获取用户的（按钮）权限(数组格式),存储到store中，在DOM结构上配合v-if绑定的值来决定按钮的显现与否。

**可以使用自定义指令。**获取用户的权限，然后检查指令绑定的值是否在权限数组中，来删除或显现按钮。

Vue.directive('per', {

bind: (el, binding, vnode) => {

if (!hasPermission(binding.value)) {

el.parentNode.removeChild(el);

}

}

});

const hasPermission = userPermission => {

// 当前用户的权限列表

let permissionList = store.state.permission；

return permissionList.indexof(userPermission)>-1；

};

// 全局判断方法

Vue.prototype.$\_has = hasPermission;

在main.js中引入

<div v-per="admin">

<div>

* 108 [说下你的vue项目的目录结构，如果是大型项目你该怎么划分结构和划分组件呢？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F339)

build 项目构建(webpack)相关代码

node\_modules npm 加载的项目依赖模块

src 这里是我们要开发的目录，基本上要做的事情都在这个目录里。里面包含了几个目录及文件：

assets： 放置一些iconfont文件，css，js，img等文件。

components: 目录里面放置组件文件，可以根据功能细分。

App.vue: 项目入口文件

main.js: 项目的核心文件。

Router.js 路由注册文件

Store.js 状态管理文件

还有一些配置文件

大型项目中，将按照组件的功能换分多个文件夹，store.js里面也可以按模块划分多个js文件。请求可以放在一个或多个js文件中。

* 109. [在移动端使用vue，你觉得最佳实践有哪些？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F338)

组件命名规范

组件代码的规范

v-for设置key

[避免 v-if 和 v-for 用在一起](https://cn.vuejs.org/v2/style-guide/#%E9%81%BF%E5%85%8D-v-if-%E5%92%8C-v-for-%E7%94%A8%E5%9C%A8%E4%B8%80%E8%B5%B7-%E5%BF%85%E8%A6%81)

使用Vuex管理状态

首屏优化（路由懒加载，还有分屏加载, 按需加载, 延时加载图片、图片进行压缩或者上传到CDN上等）

* 110. [你们项目为什么会选vue而不选择其它的框架呢？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F337)

Vue上手容易些,官方文档清晰好看，速度快，轻量，注重html，css，js的分离（相对于react的一切都是js）。（结果发现也不容易）

* 111. [对于即将到来的vue3.0特性你有什么了解的吗？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F336)

3.x 的代码库将用 TypeScript 来编写。ts支持,

更小巧，更快速；支持摇树优化；支持 Fragments 和跨组件渲染；支持自定义渲染器。

3.0 将带来一个基于 Proxy 的 observer 实现（双向数据绑定用的proxy）

IE 11 将受到支持

* 112. [vue开发过程中你有使用什么辅助工具吗？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F335)

Vue-devtools

* 113. [vue和微信小程序写法上有什么区别？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F334)

钩子函数不同 小程序的钩子函数简单onLoad/onShow/onReady/onHide/onUnload

<https://www.jianshu.com/p/d4a053599572>

* 114. [怎么缓存当前的组件？缓存后怎么更新？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F333)

1可以在路由meta中加入 meta: {keepAlive: true}参数,

2 Keep-alive 通过actived钩子函数做些处理。

* 115. [你了解什么是高阶组件吗？可否举个例子说明下？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F332)

<https://segmentfault.com/p/1210000012743259/read>

* 116. [为什么我们写组件的时候可以写在.vue里呢？可以是别的文件名后缀吗？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F331)

因为有vue-loader解析和转换.vue文件。提取出其中的逻辑代码 script、样式代码 style、以及 HTML 模版 template，再分别把它们交给对应的 Loader 去处理。

好处是：结构代码清晰，组件复用变得简单，

配合相应的loader 可以写任意的文件名后缀的文件

* 117. [vue-loader是什么？它有什么作用？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F330)

是一个加载器，解析和转换 .vue 文件，提取出其中的逻辑代码 script、样式代码 style、以及 HTML 模版 template，再分别把它们交给对应的 Loader 去处理。

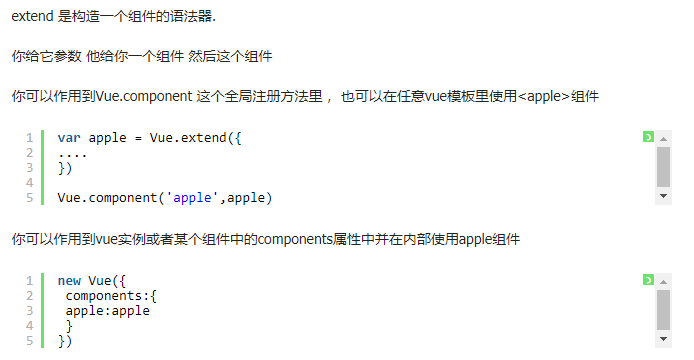
* 118. [说说你对vue的extend（构造器）的理解，它主要是用来做什么的？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F329)

使用基础 Vue 构造器，创建一个“子类”。参数是一个包含组件选项的对象。

Vue.extend() 中data必须是函数。Vue.extend 返回的是一个“扩展实例构造器”， 主要服务于Vue.component用来生成组件。可以简单理解为当在模板中遇到该组件名称作为标签的自定义元素时，会自动调用“扩展实例构造器”来生产组件实例，并挂载到自定义元素上。

跟自定义组件很像，但是他没有传递任何参数，只是个静态标签。

主要用于生成组件



* 119. [如果将axios异步请求同步化处理？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F328)

使用async /await

* 120. [怎么捕获组件vue的错误信息？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F327)

errorCaptured (err: Error, vm: Component, info: string) => ?boolean

当捕获一个来自子孙组件的错误时被调用。此钩子会收到三个参数：错误对象、发生错误的组件实例以及一个包含错误来源信息的字符串。此钩子可以返回 false 以阻止该错误继续向上传播。

errorHandler(err, vm, info) 指定组件的渲染和观察期间未捕获错误的处理函数。这个处理函数被调用时，可获取错误信息和 Vue 实例。**会捕获组件生命周期钩子里的错误;会捕获 Vue 自定义事件处理函数内部的错误;会捕获 v-on DOM 监听器内部抛出的错误.**

renderError只在开发者环境下工作。当 render 函数遭遇错误时，提供另外一种渲染输出。其错误将会作为第二个参数传递到 renderError。这个功能配合 hot-reload 非常实用。

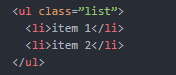
* 121. [为什么vue使用异步更新组件？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F326)

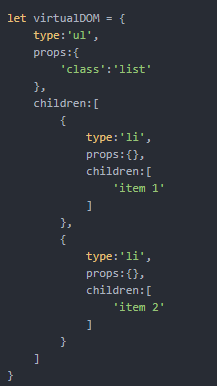
批量更新 ，去除重复数据，避免不必要的计算和 DOM 操作，减少开销。

* 122. [如何实现一个虚拟DOM？说说你的思路](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F325)

vdom是树状结构，其节点是虚拟节点，vnode和浏览器DOM中的Node一一对应。它通过JS的Object对象模拟DOM中的节点，然后再通过特定的渲染方法将其渲染成真实的DOM节点。如果有事件发生修改了虚拟DOM；比较两棵虚拟DOM树的差异，得到差异对象；把差异对象应用到真正的DOM树上。

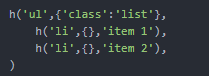
第一步：用一个对象便是真实的DOM（type表节点名称，props表示节点的属性，children表示子节点；如果子节点就是文字，比如上面那个li里面没有子节点只有文字，那么这个对象的children属性就是一个文字）





第二步：可以自己写一个虚拟DOM生成函数。





* 123. [写出多种定义组件模板的方法](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F324)
* 字符串（用单双引号，写在一行）
* 模板字面量（反引号 可多行 结构清晰）
* x-template（使用此方法，你的模板被定义在例如 index.html 文件中的 script 标签里。此 script 标签使用 text/x-template 标记，并由组件定义的 id 引用。通过template引用到组件里）
* 内联模板（通过在组件中添加 inline-template 属性，你可以向 Vue 指示内部内容是其模板，而不是将其视为分布式内容）
* render 函数（render 函数和createElement函数 将模板定义为 JavaScript 对象）
* JSX（也是利用render函数，另一种变形）
* 单文件组件

<https://blog.csdn.net/alphapersonality/article/details/80248005>

* 124 [SPA单页面的实现方式有哪些？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F323)

Hash方式和history方式

单页面应用是一种特殊的Web应用。它将所有的活动局限于一个Web页面中，仅在该Web页面初始化时加载相应的HTML、JavaScript、CSS。一旦页面加载完成，SPA不会因为用户的操作而进行页面的重新加载或跳转，而是利用JavaScript动态的变换HTML（采用的是div切换显示和隐藏），从而实现UI与用户的交互。

* 125. [说说你对SPA单页面的理解，它的优缺点分别是什么？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F322)
* 单页面应用是一种特殊的Web应用。它将所有的活动局限于一个Web页面中，仅在该Web页面初始化时加载相应的HTML、JavaScript、CSS。一旦页面加载完成，SPA不会因为用户的操作而进行页面的重新加载或跳转，而是利用JavaScript动态的变换HTML（采用的是div切换显示和隐藏），从而实现UI与用户的交互。

优点：

1).良好的交互体验

用户不需要重新刷新页面，获取数据也是通过Ajax异步获取，页面显示流畅。

2).前后端分离

3).减轻服务器压力

服务器只用出数据就可以，不用管展示逻辑和页面合成

4).共用一套后端程序代码  
 不用修改后端程序代码就可以同时用于Web界面、手机、平板等多种客户端；

5）可以缓存较多数据，减少服务器压力

缺点：

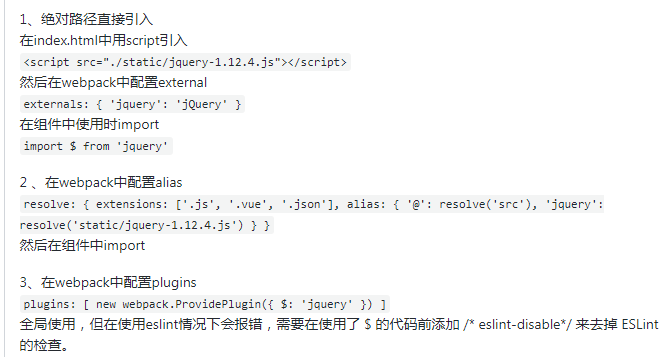
以SPA方式开发的网站不容易管理也不够安全。  
 1.不利于SEO

2.首次加载较慢  
 3. 页面复杂度提高，复杂逻辑难度提高  
 4. 前进、后退管理

* 126. [在vue项目中如何引入第三方库（比如jQuery）？有哪些方法可以做到？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F320)

1 在main.js里面引入（全局的），Vue.use（）

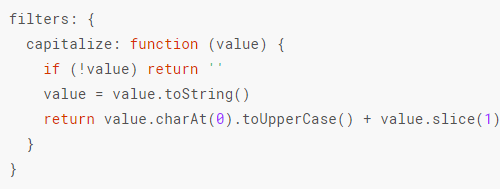
2 可以在组件里面局部引入



* 127. [使用vue手写一个过滤器](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F319)
* 全局



局部



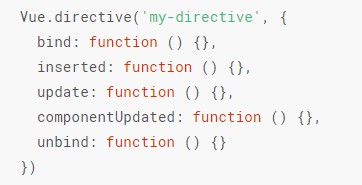
* 128. [你有使用过render函数吗？有什么好处？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F318)

没有 render函数是字符串模板的代替方案（解决字符串模板冗余重复的问题），允许你发挥 JavaScript 最大的编程能力。

* 129. [写出你常用的指令有哪些？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F317)

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

* 130. [手写一个自定义指令及写出如何调用](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F316)



<div v-my-directive></div>

* 131. [组件进来请求接口时你是放在哪个生命周期？为什么？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F315)

一般放在created, 因为在这个生命周期我们常用到的（比如数据，事件）都已经初始化好了  
如果涉及dom操作 那就放在mounted里面，这时的dom以及渲染好了。

* 132. [说说vue的优缺点分别是什么？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F313)

1、简单易用  
2、灵活渐进式  
3、轻量高效

4、数据驱动视图(MVVM)

5、组件化

缺: VUE不支持IE8, 生态环境不如angular和react

* 133. [DOM渲染在哪个周期中就已经完成了？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F312)

mounted生命周期

* 134. [第一次加载页面时会触发哪几个钩子？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F311)

beforeCreate, created, beforeMount, mounted

* 135. [vue生命周期总共有几个阶段？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F310)

8个阶段，对应八个钩子函数

* 136. [vue生命周期的作用是什么？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F309)

给用户在不同阶段添加自己的代码的机会。

* 137. [如何引入scss？引入后如何使用？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F307)

1.Vue-cli 可以直接在创建项目时直接配置 然后再style标签里面加入 lang= “sass”

2.先npm安装sass，然后在webpack里面配置相应的loader，再在style标签里面加入 lang= “sass”，并在下面通过importy引入相应的.scss文件

* 138. [使用vue开发过程你是怎么做接口管理的？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F306)

将请求按照模块封装放在不同的js文件里，统一处理；然后在组件里面引入，使用封装好的方法进行请求，请求时也可以加上async/await。

* 139. [为何官方推荐使用axios而不用vue-resource？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F305)

vue-resource只支持浏览器（体积小，支持主流的浏览器，支持Promise API和URI Templates，支持拦截器），axios可以在浏览器和 Node.js 中使用。而且axios功能更丰富。如下：

支持浏览器和node.js（在浏览器中发送 XMLHttpRequests 请求；在 node.js 中发送 http请求）

支持 [Promise](https://link.jianshu.com/?t=https%3A%2F%2Fdeveloper.mozilla.org%2Fen-US%2Fdocs%2FWeb%2FJavaScript%2FReference%2FGlobal_Objects%2FPromise) API

支持拦截请求和响应

能转换请求和响应数据

能取消请求

自动转换 JSON 数据

客户端支持防御 [CSRF](https://link.jianshu.com/?t=http%3A%2F%2Fen.wikipedia.org%2Fwiki%2FCross-site_request_forgery)

* 140. [你了解axios的原理吗？有看过它的源码吗？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F304)

axios是一个基于Promise的http请求库,可用于浏览器和 Node。原理是基于xmlHttpRequest的封装。没有。

* 141. [你有封装过axios吗？主要是封装哪方面的？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F303)
* 142. [如何中断axios的请求？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F302)

背景：因为vue在路由到某个路由的时候，通过ajax请求的数据还没有返回就跳转到另外一个路由页面了， 但是上一次的axios异步请求还在处理，所以不符合预期，需要中断上次请求。

**使用cancel token API**取消一个请求

* 143. [axios是什么？怎样使用它？怎么解决跨域的问题？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F301)

axios是一个基于Promise的http请求库,可用于浏览器和 Node。原理是基于xmlHttpRequest的封装。

使用：下包，引入就可以

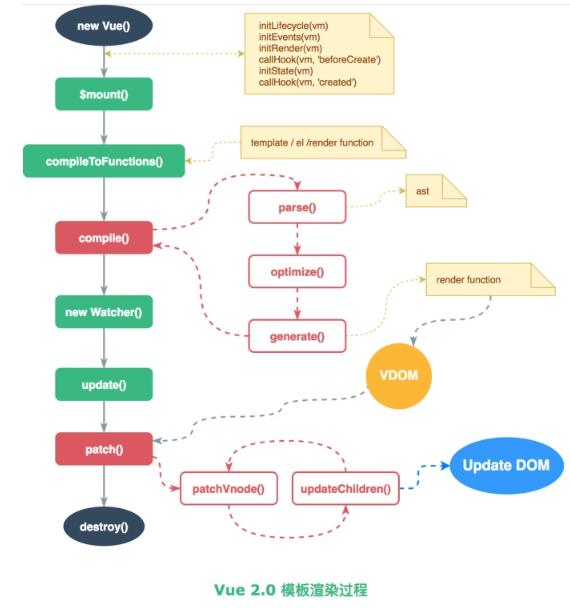
解决跨域：axios不支持jsonp，跨域是浏览器的安全限制，与axios无关。

1参考106题

2 CORS（跨域资源共享） 或者nginx代理，在axios的拦截器中做处理



* 144. [说说你对vue的template编译的理解？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F292)



首先，通过compile编译器里的parse函数把template编译成AST语法树，然后经过optimize函数优化静态内容,再经过generate 函数创建 render 函数字符串，然后得到虚拟DOM。虚拟DOM经过patch函数更新真实DOM。

* 145. [v-on可以绑定多个方法吗？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F291)

可以

* 146. [vue常用的修饰符有哪些？列举并说明](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F290)

事件修饰符



表单修饰符

.lazy .number .trim

* 147. [你认为vue的核心是什么？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F289)

虚拟DOM，数据驱动视图，组件系统

* 148. [v-model是什么？有什么用呢？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F288)

 v-model 指令在表单 <input>、<textarea> 及 <select> 元素上创建双向数据绑定。但 v-model 本质上不过是v-on和v-bind的语法糖。它负责监听用户的输入事件以更新数据。

* 149. [说说你对vue的mixin的理解，有什么应用场景？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F287)

混入 (mixin) 提供了一种非常灵活的方式，来分发 Vue 组件中的可复用功能。一个混入对象可以包含任意组件选项。当组件使用混入对象时，所有混入对象的选项将被“混合”进入该组件本身的选项。

Mixin对于封装一小段想要复用的代码来讲是有用的。可以用来扩充组件。

比如：  有两个非常相似的组件，他们的基本功能是一样的，但他们之间又存在着足够的差异性，此时的你就像是来到了一个分岔路口：我是把它拆分成两个不同的组件呢？还是保留为一个组件，然后通过props传值来创造差异性从而进行区分呢？

      两种解决方案都不够完美：如果拆分成两个组件，你就不得不冒着一旦功能变动就要在两个文件中更新代码的风险，这违背了 DRY 原则。反之，太多的props传值会很快变得混乱不堪，从而迫使维护者（即便这个人是你）在使用组件的时候必须理解一大段的上下文，拖慢写码速度。<https://www.jianshu.com/p/1bfd582da93e>

* 150. [删除数组用delete和Vue.delete有什么区别？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F285)

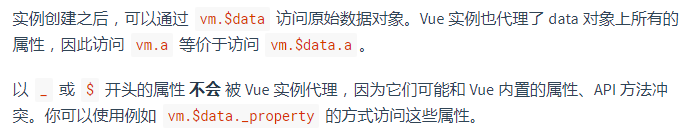
Delet：只是被删除的元素变成了 empty/undefined， 其他的元素的键值还是不变，数组长度也不变。

Vue.delete：直接删除数组成员，并改变了数组的键值，数组长度相应减少。

（对象也是一样，数组/对象是响应式的，确保删除能触发更新视图，这个方法主要用于避开 Vue 不能检测到属性被删除的限制）

* 151. [动态给vue的data添加一个新的属性时会发生什么？怎样解决？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F284)

直接添加属性会因为新的属性没有监听导致页面不能响应该数据的变化。通过全局Vue.set(object, propertyName, value)方法来动态添加响应式属性（还可以使用 vm.$set 实例方法）。



* 152. [组件和插件有什么区别？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F283)

Vue组件（component）用来构成你的App的业务模块，它的目标是App.vue。

Vue插件(plugin) 用来增强你的技术栈的功能模块， 它的目标是Vue本身。（插件是对Vue的功能的增强和补充，一个Vue插件可以是一堆Vue组件的集合）。

* 153. [说说你使用vue过程中遇到的问题（坑）有哪些，你是怎么解决的？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F274)
* 154. [说说你对选项el,template,render的理解](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F273)

el: 把vue实例挂载在对应的DOM元素上。可以是 CSS 选择器，也可以是一个 HTMLElement 实例。

template：一个字符串模板作为 Vue 实例的标识使用。模板将会 **替换** 挂载的元素。挂载元素的内容都将被忽略，除非模板的内容有分发插槽。如果值以 # 开始，则它将被用作选择符，并使用匹配元素的 innerHTML 作为模板。常用的技巧是用 <script type="x-template"> 包含模板。

Render: 字符串模板的代替方案，允许你发挥 JavaScript 最大的编程能力。该渲染函数接收一个 createElement 方法作为第一个参数用来创建 VNode。如果组件是一个函数组件，渲染函数还会接收一个额外的 context 参数，为没有实例的函数组件提供上下文信息。

## ****优先级顺序：el < template < render****

* 155. [vue实例挂载的过程是什么？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F272)

vue实例挂载主要执行mount方法。在执行$mount的时候，首先限制 el 不能为 body、html 这类根节点。然后判断render函数是否存在，若存在，则执行render编译模板，若不存在，则会把 el 或者 template 字符串通过compileToFunctions 方法实现render在线编译，得到虚拟DOM，最后通过patch方法更新DOM。在这过程中，初始化时调用了Watcher类来充当观察者模式，一旦vm实例中的数据发生变化，Watcher就执行回调函数。

图见144题。

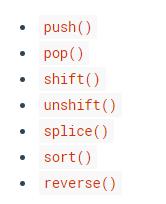
* 156. [vue在组件中引入插件的方法有哪些？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F271)

见126题

* 157. [v-if和v-for的优先级是什么？如果这两个同时出现时，那应该怎么优化才能得到更好的性能？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F270)

v-for 优先级高。 当它们处于同一节点，v-if 将分别重复运行于每个 v-for 循环中。如果你的目的是有条件地跳过循环的执行，那么可以将 v-if 置于外层元素。还可以先使用计算属性过滤不满足的。

* 158. [分别说说vue能监听到数组或对象变化的场景，还有哪些场景是监听不到的？无法监听时有什么解决方案？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F269)

这些变异数组方法vue能监听

而filter()、concat() 和 slice()不能监听，它们不会改变原始数组，而**总是返回一个新数组**。

解决办法当使用非变异方法时，可以用新数组替换旧数组。

对于对象，由于 JavaScript 的限制，**Vue 不能检测对象属性的添加或删除。添加属性使用Vue.Set（方法）**



* 159. [为什么data属性必须声明为返回一个初始数据对应的函数呢？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F267)

保证每个组件实例都有自己单独的数据。

对象为引用类型，当重用组件时，由于数据对象都指向同一个data对象，当在一个组件中修改data时，其他重用的组件中的data会同时被修改；而使用返回对象的函数，由于每次返回的都是一个新对象（Object的实例），引用地址不同，则不会出现这个问题。

* 160. [怎么在watch监听开始之后立即被调用？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F266)



在选项参数中指定 immediate: true 将立即以表达式的当前值触发回调

设置vm.$watch()方法里面的immediate:true。

* 161. [watch怎么深度监听对象变化？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F265)

设置vm.$watch()方法里面的deep:true。

* 162. [watch和计算属性有什么区别？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F264)

一个是计算属性，一个是侦听属性

一个是为了应对复杂的逻辑计算，一个是对数据的变化作出响应

一个有缓存，一个没有缓存

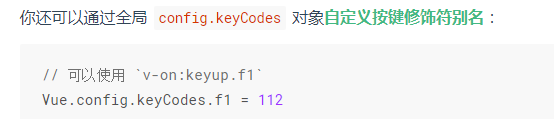
**计算属性是基于它们的响应式依赖进行缓存的**。只在相关响应式依赖发生改变时它们才会重新求值。当需要在数据变化时执行异步或开销较大的操作时，watch是最有用的。

watch适合处理的场景是，侦听一个数据的变化，当该数据变化，来处理其他与之相关数据的变化（该数据影响别的多个数据）

computed适合处理的场景是，获得一个值或者结果，该结果受其他的依赖的影响。（一个数据受多个数据影响）

* 163. [vue如何监听键盘事件？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F263)

在 v-on 在监听键盘事件时添加按键修饰符或者添加keyCode（按键码）。





* 164. [v-for循环中key有什么作用？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F262)

加快diff算法的效率，让vue在更新数据更快。

* 165. [怎么在vue中使用插件？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F261)

npm 安装 然后再main.js 引入 最后 vue.use(插件名)

* 166. [你有写过自定义组件吗？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F260)

没有

* 167. [怎么使css样式只在当前组件中生效？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F258)

在style标签中加入Scoped属性

* 168. [你有写过自定义指令吗？自定义指令的生命周期（钩子函数）有哪些？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F234)

有 全局：Vue.directive(‘focus’,{}),局部：directives:{focus:{}}



* 169. [v-show和v-if有什么区别？使用场景分别是什么？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F232)

v-if 是“真正”的条件渲染，因为它会确保在切换过程中条件块内的事件监听器和子组件适当地被销毁和重建。

不同的是带有 v-show 的元素始终会被渲染并保留在 DOM 中。v-show 只是简单地切换元素的 CSS 属性 display。

v-if 也是**惰性的**：如果在初始渲染时条件为假，则什么也不做——直到条件第一次变为真时，才会开始渲染条件块。

v-show 就简单得多——不管初始条件是什么，元素总是会被渲染，并且只是简单地基于 CSS 进行切换。

一般来说，v-if 有更高的切换开销，而 v-show 有更高的初始渲染开销。因此，如果需要非常频繁地切换，则使用 v-show 较好；如果在运行时条件很少改变，则使用 v-if 较好。

* 170. [说说你对MVC、MVP、MVVM模式的理解](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F231)

<https://blog.csdn.net/pkxiuluo01/article/details/49383783>

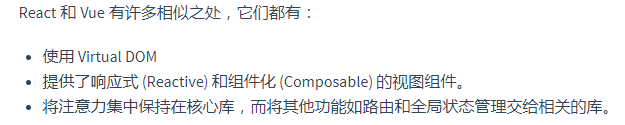
* 171. [说下你对指令的理解？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F230)

指令 (Directives) 是带有 v- 前缀的特殊特性。指令特性的值预期是**单个 JavaScript 表达式** (v-for是例外情况，稍后我们再讨论)。指令的职责是，当表达式的值改变时，将其产生的连带影响，响应式地作用于 DOM。

* 172. [vue组件之间的通信都有哪些？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F228)
* Props/$emit
* $emit/$on
* ( $parents/$children ) / $refs
* Vuex
* eventBus
* ( provide/inject )
* ( $attrs/$listeners )
* slot

<https://segmentfault.com/a/1190000019208626?utm_source=tag-newest#articleHeader9>

* 173. [vue和react有什么不同？使用场景是什么？](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Fhaizlin%2Ffe-interview%2Fissues%2F225)



vue:双向绑定，数据驱动视图；  
react: all for in js.

Vue生态不如react，Vue使用template，react使用jsx。

使用场景：

