

# 【必看】58 道 Vue 常见面试题集锦，涵盖入门到精通，自测 Vue 掌握程度

## 1.vue 优点？

答：

- 轻量级框架：只关注视图层，是一个构建数据的视图集合，大小只有几十 kb；
- 简单易学：国人开发，中文文档，不存在语言障碍，易于理解和学习；
- 双向数据绑定：保留了 angular 的特点，在数据操作方面更为简单；
- 组件化：保留了 react 的优点，实现了 html 的封装和重用，在构建单页面应用方面有着独特的优势；
- 视图，数据，结构分离：使数据的更改更为简单，不需要进行逻辑代码的修改，只需要操作数据就能完成相关操作；
- 虚拟 DOM：dom 操作是非常耗费性能的，不再使用原生的 dom 操作节点，极大解放 dom 操作，但具体操作的还是 dom 不过是换了另一种方式；
- 运行速度更快：相比较于 react 而言，同样是操作虚拟 dom，就性能而言，vue 存在很大的优势。

## 2.vue 父组件向子组件传递数据？

---

答：通过 props

## 3.子组件像父组件传递事件？

---

答：\$emit 方法

#### 4.v-show 和 v-if 指令的共同点和不同点?

---

答:

**共同点:** 都能控制元素的显示和隐藏;

**不同点:** 实现本质方法不同, `v-show` 本质就是通过控制 `css` 中的 `display` 设置为 `none`, 控制隐藏, 只会编译一次; `v-if` 是动态的向 `DOM` 树内添加或者删除 `DOM` 元素, 若初始值为 `false`, 就不会编译了。而且 `v-if` 不停的销毁和创建比较消耗性能。总结: 如果要频繁切换某节点, 使用 `v-show` (切换开销比较小, 初始开销较大)。如果不需要频繁切换某节点使用 `v-if` (初始渲染开销较小, 切换开销比较大)。

#### 5. 如何让 CSS 只在当前组件中起作用?

---

答: 在组件中的 `style` 前面加上 `scoped`

#### 6. <keep-alive></keep-alive> 的作用是什么?

---

答: `keep-alive` 是 `Vue` 内置的一个组件, 可以使被包含的组件保留状态, 或避免重新渲染。

#### 7.如何获取 dom?

---

答: `ref="domName"` 用法: `this.$refs.domName`

#### 8.说出几种 vue 当中的指令和它的用法?

---

答: `v-model` 双向数据绑定;

`v-for` 循环;

`v-if` `v-show` 显示与隐藏;

`v-on` 事件; `v-once`: 只绑定一次。

## 9. vue-loader 是什么？使用它的用途有哪些？

答：vue 文件的一个加载器，将 `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.为什么使用 key?

答：需要使用 `key` 来给每个节点做一个唯一标识，`Diff` 算法就可以正确的识别此节点。作用主要是为了高效的更新虚拟 DOM。

## 11.axios 及安装?

答：请求后台资源的模块。`npm install axios --save` 装好，`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。

## 12.v-model 的使用

答：`v-model` 用于表单数据的双向绑定，其实它就是一个语法糖，这个背后就做了两个操作：`v-bind` 绑定一个 `value` 属性；`v-on` 指令给当前元素绑定 `input` 事件。

## 13.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；`app.vue` 是一个应用主组件；`main.js` 是入口文件。

## 14.分别简述 computed 和 watch 的使用场景

答：

- `computed`：当一个属性受多个属性影响的时候就需要用到 `computed`，最典型的栗子：购物车商品结算的时候
- `watch`：当一条数据影响多条数据的时候就需要用 `watch`，栗子：搜索数据

### 15.v-on 可以监听多个方法吗？

---

答：可以，栗子：`<input type="text"`

`v-on="{ input:onInput,focus:onFocus,blur:onBlur, }">`。

### 16.\$nextTick 的使用

---

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

### 17.vue 组件中 data 为什么必须是一个函数？

---

答：因为 JavaScript 的特性所导致，在 `component` 中，`data` 必须以函数的形式存在，不可以是对象。组建中的 `data` 写成一个函数，数据以函数返回值的形式定义，这样每次复用组件的时候，都会返回一份新的 `data`，相当于每个组件实例都有自己私有的数据空间，它们只负责各自维护的数据，不会造成混乱。而单纯的写成对象形式，就是所有的组件实例共用了一个 `data`，这样改一个全都改了。

### 18.渐进式框架的理解

---

答：主张最少；可以根据不同的需求选择不同的层级；

### 19.Vue 中双向数据绑定是如何实现的？

---

答：vue 双向数据绑定是通过 数据劫持 结合 发布订阅模式的方式来实现的，也就是说数据和视图同步，数据发生变化，视图跟着变化，视图变化，数据也随之发生改变；核心：关于 VUE 双向数据绑定，其核心是 `Object.defineProperty()` 方法。

## 20.单页面应用和多页面应用区别及优缺点

---

答：单页面应用（SPA），通俗一点说就是指只有一个主页面的应用，浏览器一开始要加载所有必须的 html, js, css。所有的页面内容都包含在这个所谓的主页面中。但在写的时候，还是会分开写（页面片段），然后在交互的时候由路由程序动态载入，单页面的页面跳转，仅刷新局部资源。多应用于 pc 端。

多页面（MPA），就是指一个应用中有多个页面，页面跳转时是整页刷新

单页面的优点：用户体验好，快，内容的改变不需要重新加载整个页面，基于这一点 spa 对服务器压力较小；前后端分离；页面效果会比较炫酷（比如切换页面内容时的专场动画）。

单页面缺点：不利于 seo；导航不可用，如果一定要导航需要自行实现前进、后退。（由于是单页面不能用浏览器的前进后退功能，所以需要自己建立堆栈管理）；初次加载时耗时多；页面复杂度提高很多。

## 21.v-if 和 v-for 的优先级

---

答：当 `v-if` 与 `v-for` 一起使用时，`v-for` 具有比 `v-if` 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。所以，不推荐 `v-if` 和 `v-for` 同时使用。如果 `v-if` 和 `v-for` 一起用的话，vue 中的会自动提示 `v-if` 应该放到外层去。

## 22.assets 和 static 的区别

---

答：**相同点：** `assets` 和 `static` 两个都是存放静态资源文件。项目中所需要的资源文件图片，字体图标，样式文件等都可以放在这两个文件下，这是相同点

**不相同点：** `assets` 中存放的静态资源文件在项目打包时，也就是运行 `npm run build` 时会将 `assets` 中放置的静态资源文件进行打包上传，所谓打包简单点可以理解为压缩体积，代码格式化。而压缩后的静态资源文件最终也都会放置在 `static` 文件中跟着 `index.html` 一同上传至服务器。`static` 中放置的静态资源文件就不会要走打包压缩格式化等流程，而是直接进入打包好的目录，直接上传至服务器。因为避免了压缩直接进行上传，在打包时会提高一定的效率，但是 `static` 中的资源文件由于没有进行压缩等操作，所以文件的体积也就相对于 `assets` 中打包后的文件提交较大点。在服务器中就会占据更大的空间。

**建议：** 将项目中 `template` 需要的样式文件 js 文件等都可以放置在 `assets` 中，走打包这一流程。减少体积。而项目中引入的第三方的资源文件如 `iconfont.css` 等文件可以放置在 `static` 中，因为这些引入的第三方文件已经经过处理，我们不再需要处理，直接上传。

## 23.vue 常用的修饰符

---

答：

- `.stop`：等同于 JavaScript 中的 `event.stopPropagation()`，防止事件冒泡；
- `.prevent`：等同于 JavaScript 中的 `event.preventDefault()`，防止执行预设的行为（如果事件可取消，则取消该事件，而不停止事件的进一步传播）；
- `.capture`：与事件冒泡的方向相反，事件捕获由外到内；
- `.self`：只会触发自己范围内的事件，不包含子元素；
- `.once`：只会触发一次。

## 24.vue 的两个核心点

---

答：数据驱动、组件系统

**数据驱动：** ViewModel，保证数据和视图的一致性。

**组件系统：** 应用类 UI 可以看作全部是由组件树构成的。

## 25.vue 和 jQuery 的区别

---

答：jQuery 是使用选择器（`$`）选取 DOM 对象，对其进行赋值、取值、事件绑定等操作，其实和原生的 HTML 的区别只在于可以更方便的选取和操作 DOM 对象，而数据和界面是在一起的。比如需要获取 label 标签的内容：`$("#lable").val()`；它还是依赖 DOM 元素的值。

Vue 则是通过 Vue 对象将数据和 View 完全分离开来了。对数据进行

操作不再需要引用相应的 DOM 对象，可以说数据和 View 是分离的，他们通过 Vue 对象这个 vm 实现相互的绑定。这就是传说中的 MVVM。

## 26. 引进组件的步骤

---

答：在 template 中引入组件；在 script 的第一行用 import 引入路径；用 component 中写上组件名称。

## 27.delete 和 Vue.delete 删除数组的区别

---

答：`delete` 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。`Vue.delete` 直接删除了数组 改变了数组的键值。

## 28.SPA 首屏加载慢如何解决

---

答：安装动态懒加载所需插件；使用 CDN 资源。

## 29.Vue-router 跳转和 location.href 有什么区别

---

答：使用 `location.href= /url` 来跳转，简单方便，但是刷新了页面；使用 `history.pushState( /url )`，无刷新页面，静态跳转；引入 router，然后使用 `router.push( /url )` 来跳转，使用了 `diff` 算法，实现了按需加载，减少了 dom 的消耗。其实使用 router 跳转和使用 `history.pushState()` 没什么差别的，因为 vue-router 就是用了 `history.pushState()`，尤其是在 history 模式下。

## 30. vue slot

---



答：简单来说，假如父组件需要在子组件内放一些 DOM，那么这些 DOM 是显示、不显示、在哪个地方显示、如何显示，就是 slot 分发负责的活。

### 31.你们 vue 项目是打包了一个 js 文件，一个 css 文件，还是有多个文件？

---

答：根据 vue-cli 脚手架规范，一个 js 文件，一个 CSS 文件。

### 32.Vue 里面 router-link 在电脑上有用，在安卓上没反应怎么解决？

---

答：Vue 路由在 Android 机上有问题，babel 问题，安装 babel polypill 插件解决。

### 33.Vue2 中注册在 router-link 上事件无效解决方法

---

答：使用 `@click.native`。原因：router-link 会阻止 click 事件，.native 指直接监听一个原生事件。

### 34.RouterLink 在 IE 和 Firefox 中不起作用（路由不跳转）的问题

---

答：方法一：只用 a 标签，不适用 button 标签；方法二：使用 button 标签和 Router.navigate 方法

### 35.axios 的特点有哪些

---

答：

从浏览器中创建 XMLHttpRequests;

node.js 创建 http 请求;

支持 Promise API;

拦截请求和响应;

转换请求数据和响应数据;

取消请求;

自动换成 json。

axios 中的发送字段的参数是 data 跟 params 两个，两者的区别在于 params 是跟请求地址一起发送的，data 的作为一个请求体进行发送

params 一般适用于 get 请求，data 一般适用于 post put 请求。

### **36.请说下封装 vue 组件的过程?**

---

答:

1. 建立组件的模板, 先把架子搭起来, 写写样式, 考虑好组件的基本逻辑。(os: 思考 1 小时, 码码 10 分钟, 程序猿的准则。)
2. 准备好组件的数据输入。即分析好逻辑, 定好 props 里面的数据、类型。
3. 准备好组件的数据输出。即根据组件逻辑, 做好要暴露出来的方法。

4. 封装完毕了，直接调用即可

### 37.params 和 query 的区别

---

答：用法：query 要用 path 来引入，params 要用 name 来引入，接收参数都是类似的，分别

是 `this.$route.query.name` 和 `this.$route.params.name`。url 地址显示：

query 更加类似于我们 ajax 中 get 传参，params 则类似于 post，说的再简单一点，前者在浏览器地址栏中显示参数，后者则不显示

注意点：query 刷新不会丢失 query 里面的数据 params 刷新 会 丢失 params 里面的数据。

### 38.vue 初始化页面闪动问题

---

答：使用 vue 开发时，在 vue 初始化之前，由于 `div` 是不归 `vue` 管的，所以我们写的代码在还没有解析的情况下会容易出现花屏现象，看到类似于 `{{message}}` 的字样，虽然一般情况下这个时间很短暂，但是我们还是有必要让解决这个问题的。首先：在 css 里加上 `[v-cloak]`

`{ display: none; }`。如果没有彻底解决问题，则在根元素加上 `style="display: none;" :style="{display: block}"`

### 39.vue 更新数组时触发视图更新的方法

---

答:push(); pop(); shift(); unshift(); splice(); sort(); reverse()

### 40.vue 常用的 UI 组件库

---

答：Mint UI, element, VUX

## 41.vue 修改打包后静态资源路径的修改

---

答：cli2 版本：将 config/index.js 里的 assetsPublicPath 的值改为 ./ 。

```
build: {  
  // ...  
  assetsPublicPath: './',  
  // ...  
}
```

cli3 版本：在根目录下新建 vue.config.js 文件，然后加上以下内容：  
(如果已经有此文件就直接修改)

```
module.exports = {  
  publicPath: './', // 相对于 HTML 页面（目录相同）  
}
```

## 生命周期函数面试题篇

### 42.什么是 vue 生命周期？有什么作用？

答：每个 Vue 实例在被创建时都要经过一系列的初始化过程——例如，需要设置数据监听、编译模板、将实例挂载到 DOM 并在数据变化时更新 DOM 等。同时在这个过程中也会运行一些叫做 生命周期钩子 的函数，这给了用户在不同阶段添加自己的代码的机会。（ps：生命周期钩子就是生命周期函数）例如，如果要通过某些插件操作

DOM 节点，如想在页面渲染完后弹出广告窗，那我们最早可在 mounted 中进行。

#### 43.第一次页面加载会触发哪几个钩子？

---

答：beforeCreate, created, beforeMount, mounted

#### 44.简述每个周期具体适合哪些场景

---

答：

**beforeCreate：** 在 new 一个 vue 实例后，只有一些默认的生命周期钩子和默认事件，其他的東西都还没创建。在 beforeCreate 生命周期执行的时候，data 和 methods 中的数据都还没有初始化。不能在这个阶段使用 data 中的数据和 methods 中的方法

**create：** data 和 methods 都已经被初始化好了，如果要调用 methods 中的方法，或者操作 data 中的数据，最早可以在这个阶段中操作

**beforeMount：** 执行到这个钩子的时候，在内存中已经编译好了模板了，但是还没有挂载到页面中，此时，页面还是旧的

**mounted：** 执行到这个钩子的时候，就表示 Vue 实例已经初始化完成了。此时组件脱离了创建阶段，进入到了运行阶段。如果我们想要通过插件操作页面上的 DOM 节点，最早可以在和这个阶段中进行

**beforeUpdate:** 当执行这个钩子时，页面中的显示的数据还是旧的，data 中的数据是更新后的，页面还没有和最新的数据保持同步

**updated:** 页面显示的数据和 data 中的数据已经保持同步了，都是最新的

**beforeDestroy:** Vue 实例从运行阶段进入到了销毁阶段，这个时候上所有的 data 和 methods，指令，过滤器 .....都是处于可用状态。还没有真正被销毁

**destroyed:** 这个时候上所有的 data 和 methods，指令，过滤器 .....都是处于不可用状态。组件已经被销毁了。

#### 45.created 和 mounted 的区别

---

答：created:在模板渲染成 html 前调用，即通常初始化某些属性值，然后再渲染成视图。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。

#### 46.vue 获取数据在哪个周期函数

---

答：一般 created/beforeMount/mounted 皆可。比如如果你要操作 DOM，那肯定 mounted 时候才能操作。

#### 47.请详细说下你对 vue 生命周期的理解？

---

答：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：** 在 beforeCreated 阶段，vue 实例的挂载元素 `$el` 和 **数据对象** data 都为 undefined，还未初始化。在 created 阶段，vue 实例的数据对象 data 有了，`$el` 还没有。

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

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

**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。

## vue 路由面试题篇

### 48.mvvm 框架是什么？

---

答：vue 是实现了双向数据绑定的 mvvm 框架，当视图改变更新模型层，当模型层改变更新视图层。在 vue 中，使用了双向绑定技术，就是 View 的变化能实时让 Model 发生变化，而 Model 的变化也能实时更新到 View。

### 49.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

### 50.active-class 是哪个组件的属性？

---

答：vue-router 模块的 router-link 组件。children 数组来定义子路由

### 51.怎么定义 vue-router 的动态路由？怎么获取传过来的值？

---

答：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。

### 52.vue-router 有哪几种导航钩子？

---

答：三种，

**第一种：** 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

### 53.`\$route` 和 `\$router` 的区别

---

答：\$router 是 VueRouter 的实例，在 script 标签中想要导航到不同的 URL,使用 \$router.push 方法。返回上一个历史 history 用 \$router.to(-1)

\$route 为当前 router 跳转对象。里面可以获取当前路由的 name,path,query,parmas 等。

### 54.vue-router 的两种模式

---

答：**hash 模式：** 即地址栏 URL 中的 # 符号；



**history 模式：** window.history 对象打印出来可以看到里边提供的方法和记录长度。利用了 HTML5 History Interface 中新增的 pushState() 和 replaceState() 方法。（需要特定浏览器支持）。

## 55.vue-router 实现路由懒加载（动态加载路由）

---

答:三种方式

**第一种：** vue 异步组件技术 ==== 异步加载，vue-router 配置路由，使用 vue 的异步组件技术，可以实现按需加载。但是,这种情况下一个组件生成一个 js 文件。

**第二种：** 路由懒加载(使用 import)。

**第三种：** webpack 提供的 require.ensure(), vue-router 配置路由，使用 webpack 的 require.ensure 技术，也可以实现按需加载。这种情况下，多个路由指定相同的 chunkName，会合并打包成一个 js 文件。

## vuex 常见面试题篇

### 56.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.js，..... 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

## 57.vuex 有哪几种属性?

---

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

state => 基本数据(数据源存放地)

getters => 从基本数据派生出来的数据

mutations => 提交更改数据的方法，同步！

actions => 像一个装饰器，包裹 mutations，使之可以异步。

modules => 模块化 Vuex

## 58.Vue.js 中 ajax 请求代码应该写在组件的 methods 中还是 vuex 的 actions 中?

---

答：如果请求来的数据是不是要被其他组件公用，仅仅在请求的组件内使用，就不需要放入 vuex 的 state 里。

如果被其他地方复用，这个很大几率上是需要，如果需要，请将请求放入 action 里，方便复用。