# 关于Vue.js面试常见问题及答案解析（一）

****1、vue与react的对比,如何选型？从性能，生态圈，数据量，数据的传递上，作比较****

（1）React 和 Vue 有许多相似之处，它们都有：

使用 Virtual DOM

提供了响应式（Reactive）和组件化（Composable）的视图组件。

将注意力集中保持在核心库，伴随于此，有配套的路由和负责处理全局状态管理的库。

（2）性能：

到目前为止，针对现实情况的测试中，Vue 的性能是优于 React 的。

（3）生态圈

Vue.js: ES6 +Webpack +unit/e2e +Vue +vue-router +单文件组件 +vuex+iVew

React: ES6 +Webpack +Enzyme +React +React-router +Redux

（4）什么时候选择Vue.js

如果你喜欢用（或希望能够用）模板搭建应用，请使用Vue

如果你喜欢简单和“能用就行”的东西，请使用Vue

如果你的应用需要尽可能的小和快，请使用Vue

如果你计划构建一个大型应用程序，请使用React

如果你想要一个同时适用于Web端和原生App的框架，请选择React

如果你想要最大的生态圈，请使用React

****2、vue slot是做什么的?****

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。

****3、vue和angular的优缺点以及适用场合?****

参见：《在选择框架的时候要从哪方面入手》一题

****4、vue路由实现原理?****

以官方仓库下 examples/basic 基础例子来一点点具体分析整个流程。

和流程相关的主要需要关注点的就是 components 、 history 目录以及 create-matcher.js 、 create-route-map.js、index.js 、 install.js。

从入口，作为插件，实例化 VueRouter，实例化 History，实例化 Vue，defineReactive 定义 \_route，router-link 和 router-view 组件等几个方面展开分析

****5、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？****

根据vue-cli脚手架规范，一个js文件，一个CSS文件。

****6、vue遇到的坑，如何解决的？****

Vue1.0升级2.0有很多坑: 生命周期；路由中引入静态js，全局组件，全局变量，全局function；v-for循环的key，value值互换了位置，还有track-by；filter过滤器；遍历数组时，key值不能做model；父子通信等。

****7、vue的双向绑定的原理，和angular的对比****

在不同的 MVVM 框架中，实现双向数据绑定的技术有所不同。

AngularJS 采用“脏值检测”的方式，数据发生变更后，对于所有的数据和视图的绑定关系进行一次检测，识别是否有数据发生了改变，有变化进行处理，可能进一步引发其他数据的改变，所以这个过程可能会循环几次，一直到不再有数据变化发生后，将变更的数据发送到视图，更新页面展现。如果是手动对 ViewModel 的数据进行变更，为确保变更同步到视图，需要手动触发一次“脏值检测”。

VueJS 则使用 ES5 提供的 Object.defineProperty() 方法，监控对数据的操作，从而可以自动触发数据同步。并且，由于是在不同的数据上触发同步，可以精确的将变更发送给绑定的视图，而不是对所有的数据都执行一次检测。

****8、vue-cli，脚手架****

安装：$ npm install -g vue-cli

使用：$ vue init <template-name> <project-name>

****9、Vue里面router-link在电脑上有用，在安卓上没反应怎么解决？****

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

===============================================================================

# Vue.js面试题整理（二）

### **一、什么是MVVM？**

MVVM是Model-View-ViewModel的缩写。MVVM是一种设计思想。Model 层代表数据模型，也可以在Model中定义数据修改和操作的业务逻辑；View 代表UI 组件，它负责将数据模型转化成UI 展现出来，ViewModel 是一个同步View 和 Model的对象。

在MVVM架构下，View 和 Model 之间并没有直接的联系，而是通过ViewModel进行交互，Model 和 ViewModel 之间的交互是双向的， 因此View 数据的变化会同步到Model中，而Model 数据的变化也会立即反应到View 上。

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

### **二、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

### **三、vue的优点是什么？**

1.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。

2.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。

3.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。

4.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。

### **四、 组件之间的传值？**

1.父组件与子组件传值  
 父组件通过标签上面定义传值  
 子组件通过props方法接受数据

2.子组件向父组件传递数据  
 子组件通过$emit方法传递参数

### **五、路由之间跳转**

声明式（标签跳转） 编程式（ js跳转）

### **六、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（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

### **七、vue如何实现按需加载配合webpack设置**

webpack中提供了require.ensure()来实现按需加载。以前引入路由是通过import 这样的方式引入，改为const定义的方式进行引入。  
不进行页面按需加载引入方式：import home from '../../common/home.vue'  
进行页面按需加载的引入方式：const home = r => require.ensure( [], () => r (require('../../common/home.vue')))

### **八、vuex面试相关**

（1）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

（2）vuex有哪几种属性？

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

1. 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

2.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

3. 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？

1.可维护性会下降，想修改数据要维护三个地方；

2.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；

3.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。

### **九、 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

### **十、 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>

### **十一、<keep-alive></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。

### **十二、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>

### **十三、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

### **十四、在Vue中使用插件的步骤**

1.采用ES6的import ... from ...语法或CommonJSd的require()方法引入插件

2.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 })

### **十五、请列举出3个Vue中常用的生命周期钩子函数**

1.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

2.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。

3.activated: keep-alive组件激活时调用

### **十六、active-class是哪个组件的属性？**

vue-router模块的router-link组件。

### **十七、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。

### **十八、vue-router有哪几种导航钩子？**

一是全局导航钩子：router.beforeEach(to,from,next)，作用：跳转前进行判断拦截。  
 二：组件内的钩子；  
 三：单独路由独享组件

### **十九、生命周期相关面试题**

总共分为8个阶段创建前/后，载入前/后，更新前/后，销毁前/后。

1.创建前/后： 在beforeCreate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。

2.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。

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

4.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

（1）、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。

（2）、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。

（3）、vue生命周期总共有几个阶段

答：可以总共分为8个阶段：创建前/后, 载入前/后,更新前/后,销毁前/销毁后

（4）、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

（5）、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。

（6）、简单描述每个周期具体适合哪些场景

答：生命周期钩子的一些使用方法：

1.beforecreate : 可以在这加个loading事件，在加载实例时触发

2.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

3.mounted : 挂载元素，获取到DOM节点

4.updated : 如果对数据统一处理，在这里写上相应函数

5.beforeDestroy : 可以做一个确认停止事件的确认框

6.nextTick : 更新数据后立即操作dom

### **二十、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

### **二十一、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。  
用途：js可以写es6、style样式可以scss或less、template可以加jade等

### **二十二、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。  
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:

1.可以用变量，例如（$变量名称=值）；

2.可以用混合器，例如（）

3.可以嵌套

### **二十三、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。

### **二十四、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，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。

### **二十五、VNode是什么？虚拟 DOM是什么？**

Vue在 页面上渲染的节点，及其子节点称为“虚拟节点 (Virtual Node)”，简写为“VNode”。“虚拟 DOM”是由 Vue 组件树建立起来的整个 VNode 树的称呼。

**Vue 常见面试问题，你可能都知道概念，但都能回答准确吗？（三）**

一般常见和面试官会常问的Vue 基础知识点大致会包括这几个常见的

1.计算属性（computed）、方法（methods）和侦听属性（watch）的区别与使用场景

2.Vue 生命周期的理解

3.Vue 双向绑定，为什么不能通过修改下标来通知视图发生变化

4.简述 Vue 中的 MVVM 模型

5.Vue 路由中 hash 模式和 history 模式区别

6.Vue 路由中 router 的区别

7.组件

**一、计算属性（computed）、方法（methods）和侦听属性（watch）的区别与使用场景**

**methods VS 计算属性**

我们可以将同一函数定义为一个 method 而不是一个计算属性。对于最终的结果，两种方式确实是相同的。

然而，不同的是计算属性是基于它们的依赖进行缓存的。计算属性只有在它的相关依赖发生改变时才会重新求值。这就意味着只要 message 还没有发生改变，多次访问 reversedMessage 计算属性会立即返回之前的计算结果，而不必再次执行函数。

相比而言，只要发生重新渲染，method 调用总会执行该函数。总之，重新计算开销很大的话请选计算属性，不希望有缓存的请选methods。

**watch VS 计算属性**

当你在模板内使用了复杂逻辑的表达式时，你应当使用计算属性。

侦听属性是一个对象，键是需要观察的表达式，值是对应回调函数。值也可以是方法名，或者包含选项的对象。

当你有一些数据需要随着其它数据变动而变动时，或者当需要在数据变化时执行异步或开销较大的操作时，你可以使用 watch。

**二、Vue 生命周期的理解**

如何解释vue的生命周期才能令面试官满意？

**beforeCreate**

在实例初始化之后，数据观测 (data observer) 和 event/watcher 事件配置之前被调用。

**created**

在实例创建完成后被立即调用。在这一步，实例已完成以下的配置：数据观测 (data observer)，属性和方法的运算，watch/event 事件回调。然而，挂载阶段还没开始，$el 属性目前不可见。

**beforeMount**

在挂载开始之前被调用：相关的 render 函数首次被调用。

该钩子在服务器端渲染期间不被调用。以下周期在服务端渲染期间都不被调用。

**mounted**

el 被新创建的 vm.el 也在文档内。注意 mounted 不会承诺所有的子组件也都一起被挂载。如果你希望等到整个视图都渲染完毕，可以用 vm.$nextTick 替换掉 mounted。

**beforeUpdate**

数据更新时调用，发生在虚拟 DOM 打补丁之前。这里适合在更新之前访问现有的 DOM，比如手动移除已添加的事件监听器。

**updated**

由于数据更改导致的虚拟 DOM 重新渲染和打补丁，在这之后会调用该钩子。

**activated**

keep-alive 组件激活时调用。

**deactivated**

keep-alive 组件停用时调用。

**beforeDestroy**

实例销毁之前调用。在这一步，实例仍然完全可用。

**destroyed**

Vue 实例销毁后调用。调用后，Vue 实例指示的所有东西都会解绑定，所有的事件监听器会被移除，所有的子实例也会被销毁。

**三、Vue 双向绑定，为什么不能通过修改下标来通知视图发生变化**

Vue为什么不能检测数组变动

变异方法：

Vue 包含一组观察数组的变异方法，所以它们也将会触发视图更新。这些方法如下：

push(), pop(), shift() ,unshift(), splice(), sort(), reverse()

替换数组：

filter(), concat() 和 slice() 。这些不会改变原始数组，但总是返回一个新数组。当使用非变异方法时，可以用新数组替换旧数组：

注意事项:

由于 JavaScript 的限制，Vue 不能检测以下变动的数组：

当你利用索引直接设置一个项时，例如：vm.items[indexOfItem] = newValue

当你修改数组的长度时，例如：vm.items.length = newLength

变通方法：

Vue.set(vm.items, indexOfItem, newValue)

vm.items.splice(indexOfItem, 1, newValue)

**四、简述 Vue 中的 MVVM 模型**

Vue是以数据为驱动的，Vue自身将DOM和数据进行绑定，一旦创建绑定，DOM和数据将保持同步，每当数据发生变化，DOM会跟着变化。

ViewModel是Vue的核心，它是Vue的一个实例。Vue实例是作用在某个HTML元素上的，这个HTML元素可以是body，也可以是某个id所指代的元素。

DOM Listeners和Data Bindings是实现双向绑定的关键。DOM Listeners监听页面所有View层DOM元素的变化，当发生变化，Model层的数据随之变化；Data Bindings监听Model层的数据，当数据发生变化，View层的DOM元素随之变化。

**五、Vue 路由中 hash 模式和 history 模式区别**

hash模式

hash 出现在 URL 中，但不会被包含在 http 请求中，对后端完全没有影响，因此改变 hash 不会重新加载页面。

特点：hash虽然在URL中，但不被包括在HTTP请求中；用来指导浏览器动作，hash不会重加载页面。

history模式

history 利用了 html5 history interface 中新增的 pushState() 和 replaceState() 方法。这两个方法应用于浏览器记录栈，在当前已有的 back、forward、go 基础之上，它们提供了对历史记录修改的功能。只是当它们执行修改时，虽然改变了当前的 URL ，但浏览器不会立即向后端发送请求。

原理：

hash 模式的原理是 onhashchange 事件，可以在 window 对象上监听这个事件。

history ：hashchange 只能改变 # 后面的代码片段，history api （pushState、replaceState、go、back、forward） 则给了前端完全的自由，通过在window对象上监听popState()事件。

**六、Vue 路由中 router 的区别**

router是“路由实例”对象包括了路由的跳转方法，钩子函数等。

===============================================================================