

Vue

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？

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。

四、组件之间的传值？

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

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**

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。

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；

使用**v-show**会更加节省性能上的开销；当只需要一次显示或隐藏时，使用**v-if**更加合理。

十、如何让**CSS**只在当前组件中起作用

将当前组件的

十一、 的作用是什么？

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:'
```

A custom component!

```
'})
```

3) 使用组件

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

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 })

十五、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 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

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？

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

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。

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。

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

（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：

`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

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

Vue如何创建自定义指令？

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：

可以用变量，例如（\$变量名称=值）；

可以用混合器，例如（）

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)”，简称为“VNode”。“虚拟 DOM”是由 Vue 组件树建立起来的整个 VNode 树的称呼。