

Vue框架部分

1. 常用指令及其含义

```
1 v-text,v-html: 给标签添加文本
2 v-bind: 强制数据绑定:
3 v-on: 事件监听 @
4 v-model: 数据双向绑定 input select textarea
5 v-for: 循环遍历 遍历数组和对象 :key
6 v-if,v-else: 条件渲染
7 v-show: 条件渲染
8 v-pre: 跳过这个元素和他的子元素的编译过程
9 v-once: 只渲染元素或组件一次, 之后的重新渲染, 元素或组件及其所有的字节点将会被视为静态内容并跳过
10 ref属性: 获取ref属性标记的dom元素
11
```

2. Vue实例有哪些常见的配置选项

```
1 (1) el: 指定挂载的元素
2 (2) data: 存放初始数据
3 (3) computed: 计算属性
4 (4) methods: 定义方法
5 (5) watch: 监听数据
6 (6) filters: 过滤器
7 (7) components: 自定义组件
8 (8) 生命周期函数
9 (beforeCreate、created、beforeMount、mounted、beforeUpdate、updated、activated、beforeDestroy、destroyed)
```

3. v-show和v-if的区别

```
1 v-if是通过控制dom节点的存在与否来控制元素的显隐; v-show是通过设置DOM元素的display样式, block为显示, none为隐藏;
```

4. 路由的两个组件是什么?

```
1 <router-view>: 路由占位符
2 <router-link>: 会被渲染成a标签, to属性会渲染成a标签的href属性
```

5. 编程式路由

```
1 编程式路由通过this.$router.push()跳转页面
2 this.$router.go(-1)后退
3 router.push('/reg') //通过路径跳转
4 router.push({name:'reg'})//通过路由名称跳转
```

6. vue中router与route区别

```
1 $router是vue-router中的一个全局对象，其中包含了很多属性和对象（比如 history 对象），任何页面也都可以调用其 push(), replace(), go() 等方法
2
3 route是一个跳转的路由，每一个路由都有一个route对象，它是一个局部对象，可以获取当前路由的 path, name, params, query 等属性。
```

7. vue生命周期

```
1 beforeCreate( 创建前 )
2 在实例初始化之后，数据观测和事件配置之前被调用，此时组件的选项对象还未创建，el 和 data 并未初始化，因此无法访问 methods, data, computed等上的方法和数据。
3
4 created( 创建后 )
5 实例已经创建完成之后被调用，在这一步，实例已完成以下配置：数据观测、属性和方法的运算，watch/event事件回调，完成了data 数据的初始化，el没有。然而，挂在阶段还没有开始，$el属性目前不可见，这是一个常用的生命周期，因为你可以调用methods中的方法，改变data中的数据，并且修改可以通过vue的响应式绑定体现在页面上，，获取computed中的计算属性等等，通常我们可以在这里对实例进行预处理，也有一些童鞋喜欢在这里发ajax请求，值得注意的是，这个周期中是没有什么方法来对实例化过程进行拦截的，因此假如有某些数据必须获取才允许进入页面的话，并不适合在这个方法发请求，建议在组件路由钩子beforeRouteEnter中完成
6
7 beforeMount
8 挂在开始之前被调用，相关的render函数首次被调用（虚拟DOM），实例已完成以下的配置：编译模板，把data里面的数据和模板生成html，完成了el和data 初始化，注意此时还没有挂在html到页面上。
9
10 mounted
11 挂在完成，也就是模板中的HTML渲染到HTML页面中，此时一般可以做一些ajax操作，mounted只会执行一次。
12
13 beforeUpdate
14 在数据更新之前被调用，发生在虚拟DOM重新渲染和打补丁之前，可以在该钩子中进一步地更改状态，不会触发附加地重渲染过程
15
16 updated( 更新后 )
17 在由于数据更改导致地虚拟DOM重新渲染和打补丁只会调用，调用时，组件DOM已经更新，所以可以执行依赖于DOM的操作，然后在大多是情况下，应该避免在此期间更改状态，因为这可能会导致更新无限循环，该钩子在服务器端渲染期间不被调用
18
19 beforeDestroy( 销毁前 )
20 在实例销毁之前调用，实例仍然完全可用，
21
22 这一步还可以用this来获取实例，
23 一般在这一步做一些重置的操作，比如清除掉组件中的定时器 和 监听的dom事件
24 destroyed( 销毁后 )
25 在实例销毁之后调用，调用后，所以的事件监听器会被移出，所有的子实例也会被销毁，该钩子在服务器端渲染期间不被调用
26
```

8. vue全家桶都有什么

```
1 1.vue
2 2.vue-cli
3 3.vue-router
4 4.vuex
5 5.axios
```

9. 计算属性和属性侦听

```
1 computed是用来声明式的描述一个值依赖其它的值，当所有依赖的值或者变量改变的时候，计算属性也会随着该变；watch主要是监听data里面的定义的量，当该变量变化的时候，触发watch
```

10. 父子组件传参

- 1 1.父组件给子组件传参
- 2 - 父组件传值给子组件可以通过子组件的属性给子组件传值，分静态和动态传值
- 3 - 子组件通过子组件的props属性接收父组件的传值
- 4
- 5 2.子组件向父组件传递数据
- 6 - 子组件通过自定义事件向父组件传递信息
- 7 - 父组件监听子组件的事件
- 8 - 通过\$emit传递参数

11. 组件中的data为什么不是一个对象而是一个函数

- 1 组件中的data写成一个函数，数据使用return返回，这样每次复用组件的时候，都会返回一个新的数据，相当于每个组件实例之间的数据不会受影响，如果是对象的话，一个实例的数据改变了，其他的也会跟着变化

12. 谈谈你对MVVM开发模式的理解

- 1 MVVM分为Model、View、ViewModel三者。
- 2
- 3 Model：代表数据模型，数据和业务逻辑都在Model层中定义；
- 4 View：代表UI视图，负责数据的展示；
- 5 ViewModel：负责监听Model中数据的变化并且控制视图的更新，处理用户交互操作；
- 6
- 7 Model和View并无直接关联，而是通过ViewModel来进行联系的，Model和ViewModel之间有着双向数据绑定的联系。因此当Model中的数据改变时会触发View层的刷新，View中由于用户交互操作而改变的数据也会在Model中同步。

13. Vue请求初始化数据通常在哪个生命周期发送请求

- 1 created和mounted