

000-1810vue面试题

笔记本： 凡尘：vue

创建时间： 2018/10/8 14:32

更新时间： 2018/10/31 21:07

作者： resolveThen

001、v-show与v-if的区别

v-show:操作的是元素的display属性

v-if:操作的是元素的创建和插入

相比较而言v-show的性能要高

002、methods、computed、watch三者的区别

methods是个方法，执行的时候需要事件进行触发

computed是一个计算属性，是实时响应的，只要data中的属性发生了变化那么就会触发computed，计算属性是基于属性的依赖进行缓存的，methods调用的时候需要加()，而computed调用的时候是不需要加()

watch属性监听，watch用来监听属性的变化，当值发生变化的时候来执行特定的函数，watch监听属性的时候会有2个参数newVal和oldVal一个新值一个旧值

003、vue中自定义指令如何使用

Vue.directive():

参数1：指令名称

参数2：指令实现的函数，在回调函数中第一个参数是指令绑定的元素，参数二是一个对象其中对象中有value属性代表的是表达式返回的结果，还有modifiers属性是指令的修饰符

004、vue中组件创建的方式有哪些？

全局组件和局部组件

全局组件

Vue.component();参数1：组件名称 参数2：组件的配置项

局部组件

```
new Vue({  
  components:{} key值为组件名称 val值为组件的配置项  
})
```

005、vue中的过滤器如何使用

Vue.filter()

参数1：过滤器名称

参数2：过滤器实现的方法 该方法中有2个参数 参数1为需要过滤的数据，参数2为过滤器传递的参数

006、如何实现父子组件传值、非父子组件传值

父组件传递子组件：

通过属性进行传递 接受用props属性进行接受。 props属性用来接收外部属性。

在父组件中当子组件当作标签使用的时候给子组件添加属性，值为需要传递的值。在子组件内部通过props进行接受。接受的方式有2种，第一种为数组接受，第二种为对象接受，推荐用第二种，对象形式的可以限制数据的类型

子组件传父组件

//方案一

在父组件中当子组件当作标签使用的时候给子组件绑定自定义方法，子组件通过this.\$emit()进行触发。注意自定义方法是不需要加()的

//方案二

利用slot插槽作用域。

非父子组件传值

方案一：

创建公共的Vue对象。通过调用Vue中的\$on() \$emit()进行传值。需要接受值得一方调用\$on,需要传递值得一方调用\$emit()

方案二：

利用观察者模式封装\$on \$emit

007、请说出vue中生命周期函数(钩子函数)

- 1、beforeCreate():组件实例刚刚被创建 (el和data并未初始化)
- 2、created():组件创建完成,属性已绑定,但DOM还未生成,\$el属性还不存在 (完成data数据的初始化)
- 3、beforeMount():模板编译/挂载之前 (完成了el和data初始化)
- 4、Mounted():模板编译/挂载之后 (完成挂载)
- 5、beforeUpdate():组件更新之前
- 6、updated()：组件更新之后

7、beforeDestroy():组件销毁之前

8、destroyed():组件销毁之后

008、请解释一下vue的中单向数据流的理解

数据从父级组件传递给子组件,只能单向绑定,子组件不能直接修改从父级传递过来的数据

数据只能从一个方向来修改状态,如果父组件给N个子组件进行了数据传递。那么某一个子组件中修改了这个数据,那么就会导致其他组件的数据也会发生改变。因此数据只能从一个方向来修改状态

009、请说下什么是动态组件,如何使用,以及keep-alive的作用

动态组件:

让多个组件使用同一个挂载点,并动态进行切换。这就是动态组件

通过保留<component> </component> 元素,动态的绑定它的is特性,可以实现动态组件

keep-alive

包裹动态组件(组件)时,会缓存不活动的组件实例,而不是销毁它们

组件切换调用的时候本身会被销毁掉的,只要加上keep-alive进行包裹,就不会被销毁,而是被缓存起来,下一次使用的时候直接从缓存中调用

010、请说下组件的生命周期有哪些?

activated && deactivated:

activated: 当组件为活跃状态的时候触发(活跃状态: 进入页面的时候)

deactivated: 缓存状态的时候触发

011、vue中solt的使用方式,以及solt作用域插槽的用法

使用方式:

当组件当做标签进行使用的时候,用slot可以用来接受组件标签包裹的内容,当给solt标签添加name属性的时候,可以调换响应的位置

插槽作用域:

作用域插槽其实就是带数据的插槽,父组件接收来自子组件的slot标签上通过v-bind绑定进而传递过来的数据,父组件通过scope来进行接受子组件传递过来的数据

012、为什么在组件内部data是一个函数而不是一个对象?

因为每次调用组件的时候都会重新生成一个对象,如果是一个对象的情况下, data数据会进行复用(因为对象是引用数据类型),而当data是一个函数的时候每次调用的时候就会返回一个新的对象

013、vue中动画如何实现

哪个元素需要动画就给那个元素加transition标签

进入时class的类型分为以下几种

`<name>-enter` `<name>-enter-active` `<name>-enter-to`

离开时class的类型分为以下几种

`<name>-leave` `<name>-leave-active` `<name>-leave-to`

如果需要一组元素发生动画需要用标签`<transition-group>`

014、浅谈对路由的理解

什么是路由？

根据不同的url地址展示不同的页面或者数据。

路由分为前端路由, 和后端路由

前端路由：

- 1、前端路由多用于单页面开发, 也就是SPA。
- 2、前端路由是不涉及到服务器的, 是前端利用hash或者HTML5的historyApi来实现的, 一般用于不同的内容展示和切换

015、路由跳转的方式有哪几种？

1、a标签进行跳转。例如`首页`

2、router-link进行跳转 例如：`<router-link to="/home">首页</router-link>`

3、编程式路由 例如：`this.$router.push()`

016、路由传值的方式有哪几种

1、path路径属性传值。例如：`path:"/home/:id/name"`; 接受的时候通过 `this.$route.params`

2、query传值。因为在url中?后面的参数不会被解析, 因此我们可以通过query进行传值。 接受的时候通过 `this.$route.query`

3、路由解耦。在配置路由的时候添加props属性为true, 在需要接受参数的组件页面通过props进行接受

4、编程式导航 `this.$router.push({path:"/home",query:{}});`

017、请说出路由配置项常用的属性及作用

路由配置参数：

`path`：跳转路径

`component`：路径相对于的组件

`name`:命名路由

`children`:子路由的配置参数(路由嵌套)

`props`:路由解耦

`redirect`：重定向路由

018、编程式导航使用的方法以及常用的方法

路由跳转：`this.$router.push()`

路由替换：`this.$router.replace()`

后退：`this.$router.back()`

前进：`this.$router.forward()`

019、如何重定向路由

通过配置路由项中的`redirect`进行重定向

020、如何实现路由解耦

在路由的配置项中设置`props:true` 在需要接受组件的内部通过`props`进行接受

021、如何检测路由参数的变化

通过属性监听来实现或者`beforeRouterUpdate()`

```
watch:{  
  "$router"(){
```

```
    }  
  }  
  
  beforeRouterUpdate(to,from,next);
```

022、什么是路由守卫？路由的钩子函数有哪些？分别说出使用的场景，及用法

1、什么是路由守卫？

路由跳转前后做的一些验证

2、路由常见的钩子函数

beforeRouteEnter、beforeRouteUpdate、beforeRouteLeave

使用的场景

beforeRouteEnter:当路由进入之前:登陆验证、热力图的记录、

beforeRouteUpdate:当路由进行更新的时候。如果当前路由发生了变化，但是不需要组件的创建销毁的过程的时候，就需要用到这个钩子函数

beforeRouterLeave:当路由离开的时候、当用户没有进行支付离开的时候、当用户填写完信息没有保存的时候.....

023、什么是全局守卫

beforeEach:全局守卫。验证用户是否登陆

```
router.beforeEach((to,from,next)=>{  
  //登陆状态  
  let status = false;  
  //所以路由组建的name名字  
  const nextRouter = ["two","three","detail"];  
  
  if(nextRouter.indexOf(to.name)>=0){  
    if(!status){  
      router.push({name:"login"})  
    }else{  
      next();  
    }  
  }else{  
    next();  
  }  
})
```

024、axios与jquery的ajax有什么不同

axios的优点：

- 1、从nodejs中创建http请求
- 2、支持promiseAPI
- 3、提供了一些并发请求的接口
- 4、自动转换json数据
- 5、客户端支持防止CSRF/XSRF

ajax的缺点：

- 1、jQuery项目庞大，单纯的使用ajax却要引入整个Jquery非常不合理
- 2、基于原生的XHR开发，但是XHR架构并不清晰

025、简述vuex的数据传递流程

当组件进行数据修改的时候我们需要调用dispatch来触发actions里面的方法。actions里面的每个方法中都会有一个commit方法，当方法执行的时候会通过commit来触发mutations里面的方法进行数据的修改。mutations里面的每个函数都会有一个state参数，这样就可以在mutations里面进行state的数据修改，当数据修改完毕后，会传导给页面。页面的数据也会发生改变

026、双向数据绑定的原理

vue实现双向数据绑定的原理就是利用了 `Object.defineProperty()` 这个方法重新定义了对象获取属性值(get)和设置属性值(set)的操作来实现的。

`Object.defineProperty()` 方法会直接在一个对象上定义一个新属性，或者修改一个对象的现有属性，并返回这个对象。

027、使用vue的时候会一下子加载所有的东西使得初始化页面很卡，该如何解决？

vue-router解决首次加载缓慢的问题。懒加载简单来说就是按需加载。

像vue这种单页面应用，如果没有应用懒加载，运用webpack打包后的文件将会异常的大，造成进入首页时，需要加载的内容过多，时间过长，会出现长时间的白屏，即使做了loading也是不利于用户体验，而运用懒加载则可以将页面进行划分，需要的时候加载页面，可以有效的分担首页所承担的加载压力，减少首页加载用时。

用法：在配置路由时使用：`component:resolve=>require(["@components/路由的路径"], resolve)`。就是用了懒加载后打完包直接运行那个index.html会报错，报文件引用错误其实是打包时候路径配置有点问题，找到build下面的webpack.prod.conf.js 添加 `publicPath : "./",`

028、vue中的scoped

1. 在vue组件中，在style标签上添加scoped属性，以表示它的样式作用于当下的模块，很好的实现了样式私有化的目的，这是一个非常好的机制。
2. scoped肯定是解决了样式私有化的问题，但同时也引入了新的问题---样式不易（可）修改，而很多时候，我们是对公共组件的样式做微调的。所以我才说要谨慎使用

029、vue路由传值params和query的区别

1.query传参和接收参数

传参: `this.$router.push({ path:'/xxx' query:{ id:id } })`

接收参数: `this.$route.query.id`

注意:传参是`this.$router`,接收参数是`this.$route`,这里千万要看清了!!!

2.params传参和接收参数

传参: `this.$router.push({ name:'xxx' params:{ id:id } })`

接收参数: `this.$route.params.id`

二者还有点区别，直白的来说query相当于get请求，页面跳转的时候，可以在地址栏看到请求参数，而params相当于post请求，参数不会再地址栏中显示

030、跨域的方式

1、jsonp

2、正向代理

用http-proxy-middleware模块

`app.use("/api",proxy({`

```
  target:"协议+域名",
  changeOrigin:true
}))
```

3、cors PHP中

031、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 })

032、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` 也在文档内。
- * **activated**: `keep-alive`组件激活时调用

033、第一次页面加载会触发哪几个钩子

第一次页面加载时会触发 `beforeCreate`, `created`, `beforeMount`, `mounted` 这几个钩子

034、简单描述每个周期具体适合哪些场景

- * **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

035、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。

036、vuex有哪几种属性(modules)

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、mapGetters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。

持续更新.....