

• vue的生命周期

定义：vue的生命周期是vue实例从创建--初始化数据--编译模板--挂载dom--渲染、更新--渲染、销毁的一系列过程，称为vue的生命周期，各个阶段都有各个阶段的钩子函数

1. beforeCreate：初始化vue实例，并将this指向该实例，此时不能访问到data，message，prop里面的方法和数据（常用于非响应式数据）
2. created：实例创建完成，能访问到data，message，prop里面的方法和数据，未挂载到DOM，不能访问到\$el属性，\$ref属性内容为空（简单的ajax请求）
3. beforeMount：在挂载之前调用，找到相应的template，并编译成render函数
4. mounted：实例挂载到DOM上，可以获取到DOM节点，\$ref属性可以访问（ajax请求，获取节点信息及操作）
5. beforeUpdate：响应数据更新时调用，组件DOM还没有更新，发生在虚拟dom打补丁之前
6. updated：虚拟DOM重新渲染后调用，组件DOM已经更新，可依赖于DOM操作
7. beforeDestroy：实例销毁前，此时还能实例还能用，还能通过this获取到实例的属性
8. destroyed：实例销毁后调用，

• vue-router 是什么?它有哪些组件

vue-router是路由模块，有router-link，router-view

• 怎么定义 vue-router 的动态路由? 怎么获取传过来的值

在router目录下面的index.js文件中，在路径后面加上/:id，获取时用params.id获取

• vue-router 有哪几种导航钩子?

1. 全局守卫钩子：beforeEach(to, from, next),afterEach(to,from, next)，多用于路由拦截
2. 组件内导航钩子：beforeRouteEnter,beforeRouteUpdate,beforeRouteLeave，beforeRouteEnter不能正常获取组件实例this

```
methods:{},
beforeRouteEnter(to, from, next){}
```

• \$route 和 \$router 的区别

\$route表示当前的路由信息，可以获取路由的params,name,path等信息；\$router是RouterVue的实例方法，当导航到不同的url时，可以用this.\$router.push()方法

vue-router传参?

1. 编程式导航：router.push

```

<!-- 使用name加params -->
var router = new VueRouter({
  routes: [
    {
      path: '/detail',
      name: 'detail',
      component: Detail
    }
  ]
})
//使用this.$router.push({name:'detail',params:{id:'123'}}),
//第一次没有问题，第二次刷新参数丢失
//解决方法，修改path:"/detail/:id"

```

```

<!-- 使用path加query -->
this.$router.push({path: '/detail', query: {id: '123'}})

```

- 使用name加params

-

2. 声明式导航

```

<router-link to='detail'></router-link>
<router-link :to='{name:"detail",params:{id:123}}'></router-link>
<router-link :to='{path:"/detail",query:{id:123}}'></router-link>

```

• vue-router两种模式？

1. hash模式：原理是hashchange事件，window.onhashchange事件监听，在url后面随意加一个#xxx触发事件
2. history模式
3. 区别：hash发生变化的url都会被浏览器记录下来，因此后退或者前进时页面颜色也会变化，尽管并没有请求服务器，是单页面的标配，且随意修改#后面的内容，并不会请求服务器；history不能随意修改url，不然找不到会刷出404来，

• 路由懒加载？

1. import
2. require

• vue的优缺点？

- 优点：性能好，简单易用，前后端分离，单页面开发
- 缺点：不利于SEO

• v-if和v-show的区别？

1. v-if：每次都会重新删除和创建元素，有较高的切换性能，涉及元素的频繁切换不建议使用
2. v-show：只是切换display: none样式，有较高的初始渲染消耗

• <keep-alive> </keep-alive>的作用是什么？

组件切换过程中将状态保存，防止重复渲染DOM

• ue-loader是什么？使用它的用途有哪些？

是vue的一个加载器，把css/js/template展缓成js模块

用途：js可以写css，es6或者scss和less，template可以加jade等

v-model语法糖？

v-model是一个语法糖

```
<!-- 简写v-model -->
<input v-model='name' />

<!-- 实际上是给输入框的value绑定name值，通过@input事件把输入的值赋值给name -->
<input v-bind-value='name' v-on:input='name = $event.target.value' />
```

• vue-cli的src目录每个文件夹和文件的用法？

1. asset：存放静态资源
2. components：存放组件
3. router：定义路由相关配置
4. view：视图
5. app.vue：主组件
6. main.js：文件入口

• methods、computed和watch的联系和区别？

- methods是通过触发某一个事件或者函数回调来实现数据更新；watch是监听一个值的改变时触发；computed是最快最先执行的，HTML渲染的时候就执行了
- computed是基于他们的依赖进行缓存，当依赖改变时，computed的值也会改变，会缓存结果，必须用return返回最终结果；
- computed适用于一个属性受多个属性影响时（如购物车清单）；watch适用于一条数据影响多条数据时（如搜索数据）

• \$nextTick的使用

定义：是DOM更新后执行延迟回调

• vue组件中data为什么必须是一个函数

避免在不同组件中使用data的属性不会相互受到干扰，每用一次组件会给组件实例创建一个私有的数据空间

• vue事件对象

1. 事件：@click,@mousemove,@mouseenter等
2. prevent：阻止默认事件
3. stop：阻止事件冒泡
4. once：只执行一次
5. 键盘事件：@keydown/@keyup(鼠标按下和抬起)，enter（回车）

• 组件间的通信

1. 父→子：props；子→父：emit；公共组件；vuex;provide和inject

渐进式框架的理解

就像搭积木，根据不同的需求，在需要的时候引入他的官方或者第三方插件，就像给一个空房子，根据自己的需要，往里面增添不同的东西

• Vue中双向数据绑定是如何实现的

vue是MVVM（模型-视图-视图模型）模式的，模型相当于后台返回的数据，视图模型是一个中介，用来传递前台和后台的数据

原理：Object.defineProperty()来劫持各个属性的setter和getter

• 单页面和多页面之间的优缺点

1. 单页面：前后端分离、用户体验好，局部刷新，路径格式是多使用hash，维护相对容易；不利于SEO优化
2. 多页面：多页面跳转时需要加载所有公共资源，页面加载比较慢，用户体验不好，路径格式使用history格式，

• delete和Vue.delete删除数组的区别

1. 区别：delete删除后数据变成empty/undefined，键还在，splice和this.\$delete是把键值都删除了

```
var arr1 = [1,2,3,4]
var arr2 = [1,2,3,4]
```

```
var arr3 = [1,2,3,4]

delete this.arr1[1]
console.log(this.arr1, 'arr1'); //[1,empty,3,4]
this.arr2.splice(1,1)
console.log(this.arr2, 'arr2');//[1,3,4]
this.$delete(this.arr3,1)
console.log(this.arr3, 'arr3');//[1,3,4]
```

• location.href和vue-router跳转有什么区别

使用location跳转，简单方便，但刷新了页面；router.push按需加载，

• vuex是什么？怎么使用？哪种功能场景使用它？

定义：状态管理插件，解决不同组件之间的数据共享和持久化

如何使用：vuex有四个属性

1. state：相当于vue的data，用来存放数据，不能在里面直接修改数据
2. getters：相当于vue的computed属性，用来过滤一些数据
3. mutations：在里面定义的方法可以动态修改state里面的数据，是同步方法
4. actions：异步方法，里面是提交的mutation里面的方法，通过commit来分发action
5. modules：可以让每个模块都有自己的state、getters等属性

• 使用vuex的好处？

- 对于代行项目来说，数据的集中处理更易于维护
- 兄弟组件之间的状态会更好维护
- 缓存一些需要用到的数据（刷新时会自动销毁）

• actions和mutations的区别？

- action和mutation类似，只是action提交的是mutation的任务，而不是直接变更数据，action可以包含任意的异步操作

