

1.vue优点?

答：轻量级框架：只关注视图层，是一个构建数据的视图集合，大小只有几十kb；

简单易学：国人开发，中文文档，不存在语言障碍，易于理解和学习；

双向数据绑定：保留了angular的特点，在数据操作方面更为简单；

组件化：保留了react的优点，实现了html的封装和重用，在构建单页面应用方面有着独特的优势；

视图，数据，结构分离：使数据的更改更为简单，不需要进行逻辑代码的修改，只需要操作数据就能完成相关操作；

虚拟DOM：dom操作是非常耗费性能的，不再使用原生的dom操作节点，极大解放dom操作，但具体操作的还是dom不过是换了另一种方式；

运行速度更快：相比较与react而言，同样是操作虚拟dom，就性能而言，vue存在很大的优势。

2.vue父组件向子组件传递数据?

答：通过props

3.子组件像父组件传递事件?

答：\$emit方法

4.v-show和v-if指令的共同点和不同点?

答：共同点：都能控制元素的显示和隐藏；

不同点：实现本质方法不同，v-show本质就是通过控制css中的display设置为none，控制隐藏，只会编译一次；v-if是动态的向DOM树内添加或者删除DOM元素，若初始值为false，就不会编译了。而且v-if不停的销毁和创建比较消耗性能。

总结：如果要频繁切换某节点，使用v-show(切换开销比较小，初始开销较大)。如果不需要频繁切换某节点使用v-if（初始渲染开销较小，切换开销比较大）。

5.如何让CSS只在当前组件中起作用?

答：在组件中的style前面加上scoped

6.的作用是什么?

答：keep-alive 是 Vue 内置的一个组件，可以使被包含的组件保留状态，或避免重新渲染。

7.如何获取dom?

答：ref="domName" 用法：this.\$refs.domName

8.说出几种vue当中的指令和它的用法?

答：v-model双向数据绑定；

v-for循环；

v-if v-show 显示与隐藏；

v-on事件；v-once: 只绑定一次。

9. vue-loader是什么？使用它的用途有哪些？

答：vue文件的一个加载器，将template/js/style转换成js模块。

用途：js可以写es6、style样式可以scss或less、template可以加jade等

10.为什么使用key?

答: 需要使用key来给每个节点做一个唯一标识, Diff算法就可以正确的识别此节点。

作用主要是为了高效的更新虚拟DOM。

11.axios及安装?

答: 请求后台资源的模块。npm install axios --save装好,

js中使用import进来, 然后.get或.post。返回在.then函数中如果成功, 失败则是在.catch函数中。

12.v-modal的使用。

答: v-model用于表单数据的双向绑定, 其实它就是一个语法糖, 这个背后就做了两个操作:

v-bind绑定一个value属性;

v-on指令给当前元素绑定input事件。

13.请说出vue.cli项目中src目录每个文件夹和文件的用法?

答: assets文件夹是放静态资源; components是放组件; router是定义路由相关的配置; app.vue是一个应用主组件; main.js是入口文件。

14.分别简述computed和watch的使用场景

答: computed:

当一个属性受多个属性影响的时候就需要用到computed

最典型的栗子: 购物车商品结算的时候

watch:

当一条数据影响多条数据的时候就需要用watch

栗子: 搜索数据

15.v-on可以监听多个方法吗?

答: 可以, 栗子: 。

16.\$nextTick的使用

答: 当你修改了data的值然后马上获取这个dom元素的值, 是不能获取到更新后的值,

你需要使用\$nextTick这个回调, 让修改后的data值渲染更新到dom元素之后在获取, 才能成功。

17.vue组件中data为什么必须是一个函数?

答: 因为JavaScript的特性所导致, 在component中, data必须以函数的形式存在, 不可以是对象。

组建中的data写成一个函数, 数据以函数返回值的形式定义, 这样每次复用组件的时候, 都会返回一份新的data, 相当于每个组件实例都有自己私有的数据空间, 它们只负责各自维护的数据, 不会造成混乱。而单纯的写成对象形式, 就是所有的组件实例共用了一个data, 这样改一个全都改了。

18.渐进式框架的理解

答: 主张最少; 可以根据不同的需求选择不同的层级;

19.Vue中双向数据绑定是如何实现的?

答: vue 双向数据绑定是通过 数据劫持 结合 发布订阅模式的方式来实现的, 也就是说数据和视图同步, 数据发生变化, 视图跟着变化, 视图变化, 数据也随之发生改变;

核心：关于VUE双向数据绑定，其核心是 `Object.defineProperty()`方法。

20.单页面应用和多页面应用区别及优缺点

答：单页面应用（SPA），通俗一点说就是指只有一个主页面的应用，浏览器一开始要加载所有必须的html, js, css。所有的页面内容都包含在这个所谓的主页面中。但在写的时候，还是会分开写（页面片段），然后在交互的时候由路由程序动态载入，单页面的页面跳转，仅刷新局部资源。多应用于pc端。

多页面（MPA），就是指一个应用中有多个页面，页面跳转时是整页刷新

单页面的优点：

用户体验好，快，内容的改变不需要重新加载整个页面，基于这一点spa对服务器压力较小；前后端分离；页面效果会比较炫酷（比如切换页面内容时的专场动画）。

单页面缺点：

不利于seo；导航不可用，如果一定要导航需要自行实现前进、后退。（由于是单页面不能用浏览器的前进后退功能，所以需要自己建立堆栈管理）；初次加载时耗时多；页面复杂度提高很多。

21.v-if和v-for的优先级

答：当 v-if 与 v-for 一起使用时，v-for 具有比 v-if 更高的优先级，这意味着 v-if 将分别重复运行于每个 v-for 循环中。所以，不推荐v-if和v-for同时使用。

如果v-if和v-for一起用的话，vue中的会自动提示v-if应该放到外层去。

22.assets和static的区别

答：相同点：assets和static两个都是存放静态资源文件。项目中所需要的资源文件图片，字体图标，样式文件等都可以放在这两个文件下，这是相同点

不相同点：assets中存放的静态资源文件在项目打包时，也就是运行npm run build时会将assets中放置的静态资源文件进行打包上传，所谓打包简单点可以理解为压缩体积，代码格式化。而压缩后的静态资源文件最终也都会放置在static文件中跟着index.html一同上传至服务器。static中放置的静态资源文件就不会要走打包压缩格式化等流程，而是直接进入打包好的目录，直接上传至服务器。因为避免了压缩直接进行上传，在打包时会提高一定的效率，但是static中的资源文件由于没有进行压缩等操作，所以文件的体积也就相对于assets中打包后的文件提交较大点。在服务器中就会占据更大的空间。

建议：将项目中template需要的样式文件js文件等都可以放置在assets中，走打包这一流程。减少体积。而项目中引入的第三方的资源文件如iconfont.css等文件可以放置在static中，因为这些引入的第三方文件已经经过处理，我们不再需要处理，直接上传。

23.vue常用的修饰符

答：.stop：等同于JavaScript中的event.stopPropagation()，防止事件冒泡；

.prevent：等同于JavaScript中的event.preventDefault()，防止执行预设的行为（如果事件可取消，则取消该事件，而不停止事件的进一步传播）；

.capture：与事件冒泡的方向相反，事件捕获由外到内；

.self：只会触发自己范围内的事件，不包含子元素；

.once：只会触发一次。

24.vue的两个核心点

答：数据驱动、组件系统

数据驱动：ViewModel，保证数据和视图的一致性。

组件系统：应用类UI可以看作全部是由组件树构成的。

25.vue和jQuery的区别

答: jQuery是使用选择器 (\$) 选取DOM对象, 对其进行赋值、取值、事件绑定等操作, 其实和原生的HTML的区别只在于可以更方便的选取和操作DOM对象, 而数据和界面是在一起的。比如需要获取label标签的内容: `$("label").val();` 它还是依赖DOM元素的值。

Vue则是通过Vue对象将数据和View完全分离开来了。对数据进行操作不再需要引用相应的DOM对象, 可以说数据和View是分离的, 他们通过Vue对象这个vm实现相互的绑定。这就是传说中的MVVM。

26. 引进组件的步骤

答: 在template中引入组件;

在script的第一行用import引入路径;

用component中写上组件名称。

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

27.delete和Vue.delete删除数组的区别

答: delete只是被删除的元素变成了 empty/undefined 其他的元素的键值还是不变。Vue.delete 直接删除了数组 改变了数组的键值。

28.SPA首屏加载慢如何解决

答: 安装动态懒加载所需插件; 使用CDN资源。

29.Vue-router跳转和location.href有什么区别

答: 使用location.href='/url'来跳转, 简单方便, 但是刷新了页面;

使用history.pushState('/url'), 无刷新页面, 静态跳转;

引进router, 然后使用router.push('/url')来跳转, 使用了diff算法, 实现了按需加载, 减少了dom的消耗。

其实使用router跳转和使用history.pushState()没什么差别的, 因为vue-router就是用了history.pushState(), 尤其是在history模式下。

30. vue slot

答: 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。

31.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?

答: 根据vue-cli脚手架规范, 一个js文件, 一个CSS文件。

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

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

33.Vue2中注册在router-link上事件无效解决方法

答：使用@click.native。原因：router-link会阻止click事件，.native指直接监听一个原生事件。

34.RouterLink在IE和Firefox中不起作用（路由不跳转）的问题

答：方法一：只用a标签，不适用button标签；方法二：使用button标签和Router.navigate方法

35.axios的特点有哪些

答：从浏览器中创建XMLHttpRequests；

node.js创建http请求；

支持Promise API；

拦截请求和响应；

转换请求数据和响应数据；

取消请求；

自动换成json。

axios中的发送字段的参数是data跟params两个，两者的区别在于params是跟请求地址一起发送的，data的作为一个请求体进行发送

params一般适用于get请求，data一般适用于post put 请求。

36.请说下封装 vue 组件的过程？

答：1. 建立组件的模板，先把架子搭起来，写写样式，考虑好组件的基本逻辑。（os：思考1小时，码码10分钟，程序猿的准则。）

2. 准备好组件的数据输入。即分析好逻辑，定好 props 里面的数据、类型。

3. 准备好组件的数据输出。即根据组件逻辑，做好要暴露出来的方法。

4. 封装完毕了，直接调用即可

37.params和query的区别

答：用法：query要用path来引入，params要用name来引入，接收参数都是类似的，分别是this.\$route.query.name和this.\$route.params.name。

url地址显示：query更加类似于我们ajax中get传参，params则类似于post，说的再简单一点，前者在浏览器地址栏中显示参数，后者则不显示

注意点：query刷新不会丢失query里面的数据

params刷新 会 丢失 params里面的数据。

38.vue初始化页面闪动问题

答：使用vue开发时，在vue初始化之前，由于div是不归vue管的，所以我们写的代码在还没有解析的情况下会容易出现花屏现象，看到类似于{{message}}的字样，虽然一般情况下这个时间很短暂，但是我们还是有必要让解决这个问题的。

首先：在css里加上[v-cloak] {

display: none;

}。

如果没有彻底解决问题，则在根元素加上`style="display: none;" :style="{display: 'block'}"`

39.vue更新数组时触发视图更新的方法

答:push(); pop(); shift(); unshift(); splice(); sort(); reverse()

40.vue常用的UI组件库

答: Mint UI, element, VUX

41.vue修改打包后静态资源路径的修改

答: cli2版本: 将 config/index.js 里的 assetsPublicPath 的值改为 './'。

```
build: {  
  ...  
  assetsPublicPath: './',  
  ...  
}
```

cli3版本: 在根目录下新建vue.config.js 文件, 然后加上以下内容: (如果已经有此文件就直接修改)

```
module.exports = {  
  publicPath: '', // 相对于 HTML 页面 (目录相同) }  
}
```

42.axios与ajax的区别及优缺点

区别: axios是通过Promise实现对ajax技术的一种封装, 就像jquery对ajax的封装一样, 简单来说就是ajax技术实现了局部数据的刷新, axios实现了对ajax的封装, axios有的ajax都有, ajax有的axios不一定有, 总结一句话就是axios是ajax, ajax不止axios

优缺点:

ajax:

- 1、本身是针对MVC编程, 不符合前端MVVM的浪潮
- 2、基于原生XHR开发, XHR本身的架构不清晰, 已经有了fetch的替代方案, jquery整个项目太大, 单纯使用ajax却要引入整个jquery非常不合理 (采取个性化打包方案又不能享受cdn服务)
- 3、ajax不支持浏览器的back按钮
- 4、安全问题ajax暴露了与服务器交互的细节
- 5、对搜索引擎的支持比较弱
- 6、破坏程序的异常机制
- 7、不容易调试

axios:

- 1、从node.js创建http请求
- 2、支持Promise API
- 3、客户端防止CSRF (网站恶意利用)
- 4、提供了一些并发请求的接口

43.如何给vue自定义组件添加点击事件?

需要在@click后面加上.native,官方对于native的解释为: .native —— 监听组件根元素的原生事件

正确写法: `<my-button @click.native="alert1()" name="删除" v-bind:item2="btdata">`

44. vue-cli 工程中常用的 npm 命令有哪些?

npm install: 下载 node_modules 资源包的命令

npm run dev: 启动 vue-cli 开发环境的 npm 命令

npm run build: vue-cli 生成生产环境部署资源的 npm 命令

45. Vue 打包命令是什么？Vue 打包后会生成哪些文件？

npm run build：Vue 打包命令Vue 打包后会在当前工作目录下生成一个 dist 文件夹，文件夹中会有 static 静态文件以及 index.html 初始页面。

46. Vue 如何优化首屏加载速度？

异步路由加载

不打包库文件

关闭 sourcemap

开启 gzip 压缩

47.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,

48.在Vue中使用插件的步骤

采用ES6的import ... from ...语法

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 })

49、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：用npm 下三个loader (sass-loader、css-loader、node-sass)

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：还是在同一个文件，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"

有哪几大特性：

1、可以用变量，例如（\$变量名称=值）；

2、可以用混合器，例如：

定义了字体的混合器

```
@mixin font-dpr($font-size){
  $font:$font-size/2;
  font-size: $font;
  [data-dpr="2"] & { font-size: $font+2px;}
  [data-dpr="3"] & { font-size: $font+4px;}
}
```

使用方法如下

```
.div{
```

```
@include font-dpr(24px);
```

```
}
```

3、可以嵌套

50，mint-ui是什么？怎么使用？说出至少三个组件使用方法？

答：基于vue的前端组件库。

npm安装，然后import样式和js，vue.use (mintUi) 全局引入。在单个组件局部引入：import {Toast} from 'mint-ui'。

组件一：Toast('登录成功'); 组件二：mint-header; 组件三：mint-swiper

51.v-model是什么？怎么使用？vue中标签怎么绑定事件？

答：可以实现双向绑定，指令 (v-class、v-for、v-if、v-show、v-on) 。vue的model层的数据属性。
绑定事件：<input @click=doLog() />

52，axios是什么？怎么使用？描述使用它实现登录功能的流程？

答：请求后台资源的模块。

npm install axios -S装好，然后发送的是跨域，需在配置文件中config/index.js进行设置。后台如果是Tp5则定义一个资源路由。js中使用import进来，然后.get或.post。返回在.then函数中如果成功，失败则是在.catch函数中

53. axios+tp5进阶中，调用axios.post('api/user')是进行的什么操作？ axios.put('api/user/8')呢？

答：跨域，添加用户操作，更新操作。

54.请说下封装 vue 组件的过程？

答：首先，组件可以提升整个项目的开发效率。能够把页面抽象成多个相对独立的模块，解决了我们传统项目开发：效率低、难维护、复用性等问题。

然后，使用Vue.extend方法创建一个组件，然后使用Vue.component方法注册组件。子组件需要数据，可以在props中接受定义。而子组件修改好数据后，想把数据传递给父组件。可以采用emit方法。

55.hash路由和history路由实现原理说一下

location.hash的值实际就是URL中#后面的东西。

history实际采用了HTML5中提供的API来实现，主要有history.pushState()和history.replaceState()。

56.Vue中事件绑定的原理

Vue 的事件绑定分为两种一种是原生的事件绑定，还有一种是组件的事件绑定, 理解:

1.原生 dom 事件的绑定,采用的是 addEventListener 实现

2.组件绑定事件采用的是 \$on 方法

56。Vue父子组件生命周期调用顺序

加载渲染过程

父beforeCreate->父created->父beforeMount->子beforeCreate->子created->子beforeMount->子mounted->父mounted

子组件更新过程

父beforeUpdate->子beforeUpdate->子updated->父updated

父组件更新过程

父beforeUpdate->父updated

销毁过程

父beforeDestroy->子beforeDestroy->子destroyed->父destroyed

理解:

组件的调用顺序都是先父后子,渲染完成的顺序肯定是先子后父

组件的销毁操作是先父后子, 销毁完成的顺序是先子后父

57.vue创建组件的时候data中为什么会被return出一个对象?

可以保证组件的每一次调用都是创建一个新对象, 组件之间不会产生影响;

58.vue中有哪些内置组件:

component slot transtion fliters

59. 什么是路由懒加载:

路由懒加载是通过异步的方式来加载对应的路由组件, 提高页面相应速度

60.ssr是什么

服务端渲染

61.哪个生命周期 hook 最适合从 API 调用中获取数据?

尽管这取决于组件的用途及, 但是创建的生命周期 hook 内通常非常适合放置 API 调用。这时可以使用组件的数据和响应性功能, 但是该组件尚未渲染。

62.. 什么时候调用 “updated” 生命周期 hook ?

在更新响应性数据并重新渲染虚拟 DOM 之后, 将调用更新的 hook。它可以用于执行与 DOM 相关的操作, 但是(默认情况下)不能保证子组件会被渲染, 尽管也可以通过在更新函数中使用 `this.$nextTick` 来确保。

63.在 Vue 实例中编写生命周期 hook 或其他 option/propertie 时, 为什么不使用箭头函数?

箭头函数自己没有定义 `this` 上下文, 而是绑定到其父函数的上下文中。当你在 Vue 程序中使用箭头函数 (`=>`) 时, `this` 关键字不会绑定到 Vue 实例, 因此会引发错误。所以强烈建议改用标准函数声明。

64.什么时候使用keep-alive元素?

例如在创建时从 API 调用中引入数据的组件。你可能不希望每次动态切换这个组件进行渲染时都调用此 API。这时你可以将组件包含在 `keep-alive` 元素中。`keep-alive` 元素缓存该组件并从那里获取它, 而不是每次都重新渲染它。

65. 什么是异步组件?

当大型程序使用大量组件时, 从服务器上同时加载所有组件可能是没有意义的。在这种情况下, Vue 允许我们在需要时定义从服务器异步加载的组件。在声明或注册组件时, Vue 接受提供 `Promise` 的工厂函数。然后可以在调用该组件时对其进行“解析”。

通过仅加载基本组件并把异步组件的加载推迟到未来的调用时间, 可以节省带宽和程序加载时间。

66.绑定 class 的数组用法

对象方法 `v-bind:class="{ 'orange': isRipe, 'green': isNotRipe }"`

数组方法 `v-bind:class="[class1, class2]"`

行内 `v-bind:style="{ color: color, fontSize: fontSize+'px' }"`

67. 计算属性和 watch 的区别

区别来源于用法, 只是需要动态值, 那就用计算属性; 需要知道值的改变后执行业务逻辑, 才用 `watch`

68.computed 是一个对象时，它有哪些选项？

computed 和 methods 有什么区别？

computed 是否能依赖其它组件的数据？

watch 是一个对象时，它有哪些选项？

有get和set两个选项

methods是一个方法，它可以接受参数，而computed不能，computed是可以缓存的，methods不会。

computed可以依赖其他computed，甚至是其他组件的data

watch 配置

handler

deep 是否深度

immediate 是否立即执行

69.、js数组主要有哪些方法？主要参数你了解吗？

具体你可以查一下，常用的一般有如下：

1、shift():删除数组的第一个元素,返回删除的值。这里是0

2、unshift(3,4):把参数加载数组的前面，返回数组的长度。现在list:中是3,4,0,1,2

3、pop():删除数组的最后一个元素，返回删除的值。这里是2.

4、push(3):将参数加载到数组的最后，返回数组的长度，现在List中时：0,1,2,3

5、concat(3,4):把两个数组拼接起来。

6、splice(start,deleteCount,val1,val2,...): 从start位置开始删除deleteCount项，并从该位置起插入val1,val2,...

7、reverse：将数组反序

var a = [1,2,3,4,5]; var b = a.reverse(); //a: [5,4,3,2,1] b: [5,4,3,2,1]

8、sort(orderfunction): 按指定的参数对数组进行排序 var a = [1,2,3,4,5]; var b = a.sort(); //a: [1,2,3,4,5] b: [1,2,3,4,5]

9、slice(start,end): 返回从原数组中指定开始下标到结束下标之间的项组成的新数组

var a = [1,2,3,4,5]; var b = a.slice(2,5); //a: [1,2,3,4,5] b: [3,4,5]

70.js迭代的方法

every()、filter()、forEach()、map()、some()

71，谈谈对vue 组件化的理解。

组件是可复用的 Vue 实例，准确讲它们是VueComponent的实例，继承自Vue。可以增加代码的复用性、可维护性和可测试性。提高开发效率，方便重复使用，简化调试步骤，提升整个项目的可维护性，便于协同开发，是高内聚、低耦合代码的实践。

72，谈谈对Vue 路由的理解。

根据不同的url展示不同的页面或者数据。

分类：路由分为前端路由和后端路由。

前端路由：主要用于单页面的开发，根据用户请求的地址来渲染不同的页面。前端路由不会经过后端，而是根据hash值的变化进行页面数据的渲染，所以不会刷新，不跳转。

原理：通过hashRouter (onhashchange) 或者history路由 (h5 historyAPI) 进行页面的切换。

后端路由：根据用户请求的路径返回不同的页面或数据。

73，created 和 mounted 的区别

created:在模板渲染成html前调用,即通常初始化某些属性值,然后再渲染成视图。 mounted:在模板渲染成html后调用,通常是初始化页面完成后,再对html的dom节点进行一些需要的操作

74, created 和 mounted 分别什么时候用?

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。

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。

75, vue 和 angular 分别采用什么技术实现双向数据绑定?

76.描述下 vue 从初始化页面-修改数据-刷新页面 UI 的过程?

当 Vue 进入初始化阶段时, 一方面 Vue 会遍历 data 中的属性, 并用 Object.defineProperty 将它转化成 getter/setter 的形式, 实现数据劫持(暂不谈 Vue3.0 的 Proxy); 另一方面, Vue 的指令编译器 Compiler 对元素节点的各个指令进行解析, 初始化视图, 并订阅 Watcher 来更新视图, 此时 Watcher 会将自己添加到消息订阅器 Dep 中, 此时初始化完毕。

当数据发生变化时, 触发 Observer 中 setter 方法, 立即调用 Dep.notify(), Dep 这个数组开始遍历所有的订阅者, 并调用其 update 方法, Vue 内部再通过 diff 算法, patch 相应的更新完成对订阅者视图的改变。

77. Vue中watch、methods 和 计算属性的区别是什么?

1、computed和methods

共同点: computed能实现的methods也能实现;

不同点: computed是基于它的依赖进行缓存的。computed只有在它的相关依赖发生变化才会重新计算求值。而只要它的相关依赖没有发生变化, 多次访问会立即返回之前的计算结果, 而不必再次执行计算。相比之下, 每当触发重新渲染时, 调用方法将总会再次执行函数。也就是说当我们不希望有缓存, 用方法来替代。

2、watch和computed

共同点: 都是以Vue的依赖追踪机制为基础的, 都是希望在依赖数据发生改变的时候, 被依赖的数据根据预先定义好的函数, 发生“自动”的变化。、;

不同点: watch擅长处理的场景: 一个数据影响多个数据; computed擅长处理的场景: 一个数据受多个数据影响。虽然计算属性在大多数情况下更合适, 但有时也需要一个自定义的侦听器, 当需要在数据变化时执行异步或开销较大的操作时, 通过侦听器最有用。

78.vue中怎么重置data

第一种, 手动去给表单双向绑定的对象赋值为空

this.\$options.data()调用它我们可以获取到页面data在刚初始化时的状态那么这样我们就可以将data中的某个数据重置到初始状态

如果我想把整个data都重置到初始化状态呢?

Object.assign(this.\$data, this.\$options.data())

那就用这个方法

this.\$data 获取的是当前的data对象

Object.assign则可以将一个对象赋值给另一个对象

两个参数，可以将参数2 赋值给 参数1

79.Vue组件中写 name 选项有什么作用？

当项目使用keep-alive时，可搭配组件name进行缓存过滤

DOM递归迭代时需要调用自身name

当你用vue-devtools调试工具里显示的组件名称是由vue中组件name决定的

80. vuex 的 store 特性是什么

vuex 就是一个仓库，仓库里放了很多对象。其中 state 就是数据源存放地，对应于一般 vue 对象里面的 data

state 里面存放的数据是响应式的，vue 组件从 store 读取数据，若是 store 中的数据发生改变，依赖这相数据的组件也会发生更新

它通过 mapState 把全局的 state 和 getters 映射到当前组件的 computed 计算属性

81. Vuex 如何区分 state 是外部直接修改，还是通过 mutation 方法修改的？

Vuex中修改state的唯一渠道就是执行 commit('xx', payload) 方法，其底层通过执行 this.withCommit(fn) 设置committing标志变量为true，然后才能修改state，修改完毕还需要还原 committing 变量。外部修改虽然能够直接修改state，但是并没有修改committing标志位，所以只要 watch一下state，state change时判断是否_committing值为true，即可判断修改的合法性。

82. 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。

83. Class 与 Style 如何动态绑定？

Class 可以通过对象语法和数组语法进行动态绑定：

对象语法：

```
data: { isActive: true, hasError: false }
```

数组语法：

```
data: { activeClass: 'active', errorClass: 'text-danger' }
```

Style 也可以通过对象语法和数组语法进行动态绑定

对象语法：

```
data: { activeColor: 'red', fontSize: 30 }
```

数组语法：

```
data: { styleColor: { color: 'red' }, styleSize:{ fontSize:'23px' } }
```

生命周期函数面试题

1.什么是 vue 生命周期？有什么作用？

答：每个 Vue 实例在被创建时都要经过一系列的初始化过程——例如，需要设置数据监听、编译模板、将实例挂载到 DOM 并在数据变化时更新 DOM 等。同时在这个过程中也会运行一些叫做 生命周期钩子 的函数，这给了用户在不同阶段添加自己的代码的机会。（ps：生命周期钩子就是生命周期函数）例如，如果要通过某些插件操作DOM节点，如想在页面渲染完后弹出广告窗，那我们最早可在mounted 中进行。

2.第一次页面加载会触发哪几个钩子？

答：beforeCreate，created，beforeMount，mounted

3.简述每个周期具体适合哪些场景

答：beforeCreate：在new一个vue实例后，只有一些默认的生命周期钩子和默认事件，其他的東西都还没创建。在beforeCreate生命周期执行的时候，data和methods中的数据都还没有初始化。不能在这个阶段使用data中的数据和方法

create：data 和 methods都已经被初始化好了，如果要调用 methods 中的方法，或者操作 data 中的数据，最早可以在这个阶段中操作

beforeMount：执行到这个钩子的时候，在内存中已经编译好了模板了，但是还没有挂载到页面中，此时，页面还是旧的

mounted：执行到这个钩子的时候，就表示Vue实例已经初始化完成了。此时组件脱离了创建阶段，进入到了运行阶段。如果我们想要通过插件操作页面上的DOM节点，最早可以在和这个阶段中进行

beforeUpdate：当执行这个钩子时，页面中的显示的数据还是旧的，data中的数据是更新后的，页面还没有和最新的数据保持同步

updated：页面显示的数据和data中的数据已经保持同步了，都是最新的

beforeDestory：Vue实例从运行阶段进入到了销毁阶段，这个时候上所有的 data 和 methods，指令，过滤器都是处于可用状态。还没有真正被销毁

destroyed：这个时候上所有的 data 和 methods，指令，过滤器都是处于不可用状态。组件已经被销毁了。

4.created和mounted的区别

答：created:在模板渲染成html前调用，即通常初始化某些属性值，然后再渲染成视图。

mounted:在模板渲染成html后调用，通常是初始化页面完成后，再对html的dom节点进行一些需要的操作。

5.vue获取数据在哪个周期函数

答：一般 created/beforeMount/mounted 皆可。

比如如果你要操作 DOM，那肯定 mounted 时候才能操作。

6.请详细说下你对vue生命周期的理解？

答：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beforeCreated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。

vue路由面试题

1.mvvm 框架是什么？

答：vue是实现了双向数据绑定的mvvm框架，当视图改变更新模型层，当模型层改变更新视图层。在vue中，使用了双向绑定技术，就是View的变化能实时让Model发生变化，而Model的变化也能实时更新到View。

2.vue-router 是什么?它有哪些组件

答：vue用来写路由一个插件。router-link、router-view

3.active-class 是哪个组件的属性？ 嵌套路由怎么定义？ children

答：vue-router模块的router-link组件。children数组来定义子路由

4.怎么定义 vue-router 的动态路由? 怎么获取传过来的值？

答：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。

5.vue-router 有哪几种导航钩子？

答：三种，

第一种：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

6.\$route 和 \$router 的区别

答：\$router是VueRouter的实例，在script标签中想要导航到不同的URL,使用\$router.push方法。返回上一个历史history用\$router.to(-1)

\$route为当前router跳转对象。里面可以获取当前路由的name,path,query,parmas等。

7.vue-router的两种模式

答:hash模式：即地址栏 URL 中的 # 符号；

history模式：window.history对象打印出来可以看到里边提供的方法和记录长度。利用了 HTML5 History Interface 中新增的 pushState() 和 replaceState() 方法。（需要特定浏览器支持）。

8.vue-router实现路由懒加载（ 动态加载路由 ）

答:三种方式

第一种：vue异步组件技术 ==== 异步加载，vue-router配置路由，使用vue的异步组件技术，可以实现按需加载 .但是,这种情况下一个组件生成一个js文件。

第二种：路由懒加载(使用import)。

第三种：webpack提供的require.ensure(), vue-router配置路由，使用webpack的require.ensure技术，也可以实现按需加载。这种情况下，多个路由指定相同的chunkName，会合并打包成一个js文件。

9. vue的路由使用步骤？

1.下载vue-router路由模块；

2.创建路由对象；

3.配置路由规则;

4.将路由对象注册为vue实例对象的成员属性;

vuex常见面试题

1.vuex是什么? 怎么使用? 哪种功能场景使用它?

答: vue框架中状态管理。在main.js引入store, 注入。

新建了一个目录store.js, 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?

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

state => 基本数据(数据源存放地)

getters => 从基本数据派生出来的数据

mutations => 提交更改数据的方法, 同步!

actions => 像一个装饰器, 包裹mutations, 使之可以异步。

modules => 模块化Vuex

vuex的State特性是?

state就是存放数据的地方, 类似一个仓库, 特性就是当mutation修改了state的数据的时候, 他会动态的去修改所有的调用这个变量的所有组件里面的值 (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)

vuex的Getter特性是?

getter用来获取数据, mapgetter经常在计算属性中被使用

vuex的Mutation特性是?

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.Vue.js中ajax请求代码应该写在组件的methods中还是vuex的actions中?

答: 如果请求来的数据是不是要被其他组件公用, 仅仅在请求的组件内使用, 就不需要放入vuex 的state里。如果被其他地方复用, 这个很大几率上需要的, 如果需要, 请将请求放入action里, 方便复用