**vue做了大量封装，将常用功能封装成v-指令:**

1.v-on 绑定事件---简写@

2.v-bind 在标记（组件）中，给**属性**绑定变量（对象）--简写 :

在HTML属性中不能使用{{}}，应使用v-bind指令

3.v-if / v-else / v-show添加或移除DOM结构

4.v-for 遍历对象或数组

5.v-model 双向数据绑定 表单项的value值绑定到指定变量

6.v-once 只绑定一次（渲染一次就解绑函数）

7.v-html 绑定HTML

8.v-text 与插值表达式类似{{ }}

**v-if / v-show的区别**

* v-if 是“真正的”条件渲染，因为它会确保在切换过程中事件监听器和子组件适当地被销毁和重建。
* 相比之下， v-show 不管初始条件是什么，元素总是会被渲染，并且只是简单地基于 CSS 进行切换。
* 一般来说， v-if 有更高的切换开销，而 v-show 有更高的初始渲染开销。所以，如果需要非常频繁地切换，则使用 v-show 较好；

如果在运行时条件不太可能改变，则使用 v-if 较好。

**什么是v-model?**

v-model是双向数据绑定。

它在内部用onInput做了封装，它会在onkeydown之后，onkeyup之前触发。

**v-model的lazy -修饰符（失去焦点才改变值）**

**v-model的number-修饰符（将输入input的字符串成数字）**

**v-model的trim-修饰符（去除两边空格）**

**vue中实现双向数据绑定原理**

vue是采用数据劫持 和 发布者-订阅者结合的方式，来做数据绑定，核心是通过Object.defineProperty()，劫持各个属性的getter和setter，在数据模型变化的时候，发布消息给订阅者（绑定了数据模型的DOM元素），触发相应的监听回调。

**什么是MVVM?（v-model如何实现双向数据绑定的？得益于MVVM）**

MVVM 是一种设计思想。

Model代表数据模型层，可以在 Model 中修改数据和 编写业务逻辑；

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中，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。

**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，这些问题。

区别：最大区别是JQ是获得DOM操作DOM，而react和vue中引入虚拟DOM。

场景：数据操作比较多的场景，更加便捷。

**Vue的特性如下：**

1. 轻量级的框架

2. 双向数据绑定

3. v指令

4. 插件化

5. 数据中心化管理

6. 简洁、快速和高效

**vue与React的区别**

相同点：

中心思想相同：一切都是组件，组件实例之间可以嵌套。

都提供了钩子函数，可以让开发者按照自己的需求定制。

都没有内置AJAX，Route等 具有功能的核心包，而是以插件的方式加载。

不同点：

1、react严格上针对的是mvc模式的view层，vue则是mvvm模式。

2、操作dom的方式不同,vue使用的是指令操作dom，react是通过js进行操作。

3、vue可以通过v-model实现双向数据绑定。

4、react中state是不能直接改变的，需要使用setState改变。

vue中的state不是必须的，数据主要是在data属性中管理。

**Vue预定义变量**

$parent 父组件对象

$children 子组件对象组成的数组

$on 绑定自定义事件

$emit ·触发自定义事件

**组件通信**

1.父组件通过**props**向子组件传递数据-->单向数据流

2.通过子组件的props，改变父组件的数据---对象（引用类型，存储、赋值、返回值都是地址）

3.**事件订阅**：想要给谁传数据，处理函数就在谁上定义

（1）**子给父传递数据**

--父组件中， 绑定子组件的自定义事件 在父组件写处理函数

--子组件中， 触发自定义事件

<子组件 @自定义事件="父组件处理函数">

（this.$emit(‘自定义事件名称’,’数据’)）

--父组件中， 绑定子组件的自定义事件

mounted(){

this.$refs.子组件的ref值.$on("自定义事件",(msg)=>{this.message=msg;})

}

--子组件中， 触发自定义事件

（this.$emit(‘自定义事件名称’,’数据’)）

(2)**子组件之间的通信（child2给child1传递数据）**

<1>将父组件App当做公共bus

在Child1中给父组件App绑定事件

this.$parent.$on("myEvent",(msg)=>{})

在Child2 中触发事件

this.$parent.$emit("myEvent","传递给Child1数据")

<2>Child1组件绑定自定义事件，在Child2中触发Child1上自定义事件

问题:如何在Child2中获取Child1对象

this.$parent.$refs.

3. slot作用域插槽

4.组件通信--大型项目—vuex

**Vue Slot是做什么的？（Vue Slot应用场景）**

用于组件间传值。

应用场景，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，就用到了slot插槽。

**vuex 是什么（vuex的原理）？怎么使用？哪种功能场景使用它（vuex的作用）？**

Vuex也是组件间传值的一种方式。

通过dispatch发送action给actions做异步处理，之后，通过commit，将action发送给Mutations做处理，之后呢,得到了新的state，然后重新渲染。

---------------------------

使用：首先，安装vuex cnpm i vuex -S

在入口文件写，import Vuex from 'vuex'

注册：Vue.use(Vuex);

new Vuex.Store({ })

应用场景：当我们的组件之间进行复杂传值，很困难的时候，我们可以用vuex进行组件间传值。可以解决不同组件之间数据传递和共享的问题。

**不用Vuex会带来什么问题？**

1.可维护性差，一个数据修改，需要改动多个地方。

2.可读性会下降，根本就看不出来数据是从哪来的。

3.不使用vuex会增加耦合，本来Vue用 Component 就是为了减少耦合，现在不使用vuex，和组件化的初衷相背。

**vuex 有哪几种属性？（vuex的核心包括以下几点）**

* state
* mutations
* getters
* actions
* modules  
  state里面存放的就是状态数据  
  mutations就是如何更改状态数据  
  getters就是从state中派生出新状态，比如将state中的某个状态进行过滤然后获取新的状态。  
  actions 可以进行异步操作。  
  modules 就是说 当我们的业务逻辑非常复杂的时候，那我们就可以把业务逻辑按照功能分成几块。这和我们创建js模块是一个目的，让代码结构更清晰。

**如何让CSS只在当前组件中起作用**

将当前组件的<style>修改为<style **scoped**>

**<keep-alive></keep-alive>的作用是什么?**

<keep-alive></keep-alive> 包裹动态组件时，会做缓存，主要用于保留组件状态或避免重新渲染。

**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 this.$route.params.id。

**路由之间跳转**

1> <router-link></router-link>

2> this.$router.push({

path: `/describe/${id}`,

})

3> this.$router.replace{path：‘/’ }

$route和$router的区别答：$route是“路由信息对象”，包括path，params，hash，query，fullPath，matched，name等路由信息参数。而$router是“路由实例”对象，包括了路由的跳转方法，钩子函数等。

**Vue的生命周期？**

**答：beforeCreate 创建前**

* **Created 创建后**
* **beforeMount 挂载前**
* **Mounted 挂载后**
* **beforeUpdate 更新前**
* **Updated 更新后**
* **beforeDestroy 销毁前**
* **Destroyed 销毁后**

##### **在beforeCreate和created之间**

在created的时候数据已经和data属性进行绑定

此时还没有el选项

对data和methods里面的数据进行初始化，所以如果要操作数据和调用方法最早是在created里面。

##### **created钩子函数和beforeMount间的生命周期**

首先会判断对象是否有el选项。如果有的话就继续向下编译，如果没有el选项，则停止编译。

1. **beforeMount和mounted 钩子函数间的生命周期**

给vue实例对象**添加$el成员**，并且替换掉挂在的DOM元素。

##### d.**beforeUpdate钩子函数和updated钩子函数间的生命周期**

当vue发现data中的数据发生了改变，会触发对应组件的重新渲染，先后调用beforeUpdate和updated钩子函数

##### **e. beforeDestroy和destroyed钩子函数间的生命周期**

beforeDestroy钩子函数在实例销毁之前调用。在这一步，实例仍然完全可用。  
destroyed钩子函数在Vue 实例销毁后调用。调用后，Vue 实例指示的所有东西都会解绑定，所有的事件监听器会被移除，所有的子实例也会被销毁。

**请列举出3个Vue中常用的生命周期钩子函数**

 **created:** 实例创建完成之后调用,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。

 **activated:** keep-alive组件激活时调用

**什么是vue生命周期**

答： Vue 实例从创建、更新到销毁的过程，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。

**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

**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

**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。

**Vue中**[**计算属性**](https://cn.vuejs.org/v2/guide/computed.html#%E8%AE%A1%E7%AE%97%E5%B1%9E%E6%80%A7)**（什么时候会用到计算属性？）**

在模板中放入太多的逻辑会让模板过重且难以维护。这时候会使用**计算属性**

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

安装 babel polypill（怕一tu-） 插件去解决。

**对于vue组件而言，数据一共有三种形式，分别是哪三种？**

1.子组件中的props来接收父组件的数据

2.修饰符.once绑定和.sync双向绑定等

3、组件通信，也就是非父子组件之间的通信，通过事件订阅来实现

**$nextTick的作用**

$nextTick是在下次 DOM 更新循环结束之后执行延迟回调，在修改数据之后使用 $nextTick，则可以在回调中获取更新后的 DOM。

**Mint-ui是什么？怎么使用？说出至少三个组件使用方法？**

**解析：**

**Mint-ui:**基于vue的前端组件库。

**使用方法：**npm安装，然后import 它的样式和js，vue.use（mintUi）全局引入。

**组件使用方法：**

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；

**Vue项目负责哪些模块？主要是什么功能？遇到过什么困难？**

**看笔记**

**什么是微信小程序？**

微信小程序（wei xin xiao cheng xu），简称小程序，英文名Mini Program，是一种不需要下载安装即可使用的应用，它实现了应用“触手可及”的梦想，用户扫一扫或搜一下即可打开应用。微信小程序是一种全新的连接用户与服务的方式，它可以在微信内被便捷地获取和传播，同时具有出色的使用体验。

**小程序的双向绑定和vue哪里不一样？**

（1）设置值

在vue中,只需要再表单元素上加上v-model,然后再绑定data中对应的一个值，当表单元素内容发生变化时，data中对应的值也会相应改变；

小程序中，当表单内容发生变化时，会触发表单元素上绑定的方法，然后在该方法中，通过this.setData({key:value})来将表单上的值赋值给data中的对应值。当页面表单元素很多的时候，更改值就是一件体力活了。

1. 取值

vue中，通过this.reason取值；小程序中，通过this.data.reason取值

**小程序本地存储有哪几种方法？**

wx.setStorage(OBJECT)将数据存储在本地缓存中指定的 key 中，会覆盖掉原来该 key 对应的内容，这是一个异步接口。

wx.setStorageSync(KEY,DATA)将 data 存储在本地缓存中指定的 key 中，会覆盖掉原来该 key 对应的内容，这是一个同步接口。

**小程序如何实现下拉刷新？**

设置"enablePullDownRefresh":true;

Page.onPullDownRefresh()，在 Page 中定义 onPullDownRefresh 处理函数，监听该页面用户下拉刷新事件。

wx.startPullDownRefresh(OBJECT)开始下拉刷新，调用后触发下拉刷新动画，效果与用户手动下拉刷新一致。

wx.stopPullDownRefresh()停止当前页面下拉刷新。

**小程序和vue写法的区别？**

动态绑定一个变量的值为元素的某个属性的时候，vue会在变量前面加上冒号：；小程序会用两个大括号括起来，如果不加括号，则被认为是字符串。

控制元素的显示和隐藏，vue使用v-if 和v-show；小程序使用wx-if和hidden。

绑定事件，vue使用v-on:event，或者@event；小程序全用bindtap(bind+event)，或者catchtap(catch+event)绑定事件。