**Vue.js课程**

**学习阶段：**

为了便于初学者入门和适应Vue.js，本课程主要分为两个阶段：

第一阶段：刀耕火种时代的学习

第二阶段：高科技时代式的学习

<https://cn.vuejs.org/v2/guide/>

第一阶段

**一 简介**

Vue (读音 /vjuː/，类似于 view) 是一套用于构建用户界面的渐进式框架。与其它大型框架不同的是，Vue 被设计为可以自底向上逐层应用。Vue 的核心库只关注视图层，不仅易于上手，还便于与第三方库或既有项目整合。另一方面，当与[现代化的工具链](https://cn.vuejs.org/v2/guide/single-file-components.html)以及各种[支持类库](https://github.com/vuejs/awesome-vue#libraries--plugins)结合使用时，Vue 也完全能够为复杂的单页应用提供驱动。

**二 安装**

下载开发版本，使用script标签引入

例：



**三 指令**

3.1 v-html 显示元素的innerHTML

例：<div v-html=’msg’></div>

3.2 v-text 显示元素的innerText

例：<div v-text=’msg’></div>

**练习**：创建一个小小的vue，分别使用v-html,v-text和{{}}的方法展示三个数据

3.3 v-show 根据表达式的结果的真假，控制元素的显示隐藏，本质上改变的display

3.4 v-if 根据条件判断的真假，控制元素的生成与否，可以和v-else配合使用

<p v-if=’ok’>显示内容</p>

<p v-else>您的购物车为空</p>

**练习**：创建一个小小的vue，使用v-show和v-if控制数据的显示

3.5 v-for 将数组渲染为列表，格式为v-for=’(value,index) in arr’



v-for遍历对象，格式为v-for=’(item,key,index) in obj’

**练习**：将“唯品会-鞋包品牌”部分使用v-for实现，要求效果100%接近原稿

3.6 v-on 绑定事件

例:<button v-on:click=’change’>按钮</button>

简写为@:click=’change’

**练习**：写一个click事件，控制一个弹出层的显示隐藏

事件修饰符：

.stop 阻止冒泡

.once 事件只触发一次

.prevent 取消默认行为

.self 当事件发生在该元素本身时发生，而不是发生在子元素上

键盘事件修饰符：

.enter

.tab

.delete

.esc

.space

.up

.down

.left

.right

还可以自定义按键别名

.keyCodes.f1=112

表单元素中的修饰符

.number 将输入的值的类型转为Number，如果原值转换结果为NaN,则返回原值

.trim 过滤用户的首尾空格

.lazy 发生change时才会同步数据

3.7 v-model 在表单元素上创建双向数据绑定

1. 用在input上



1. 用在textarea上

c) 单个复选框，显示逻辑值



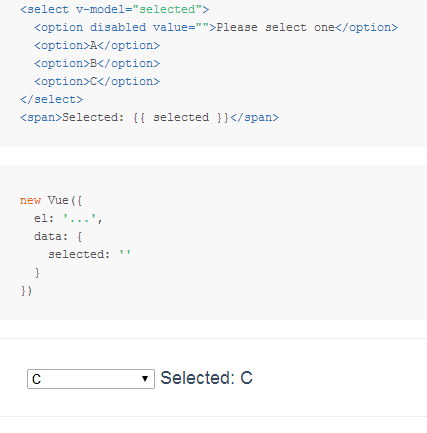
1. 多个复选框，绑定到同一个数组上



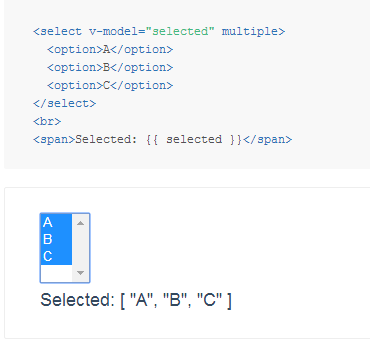
1. 单选框



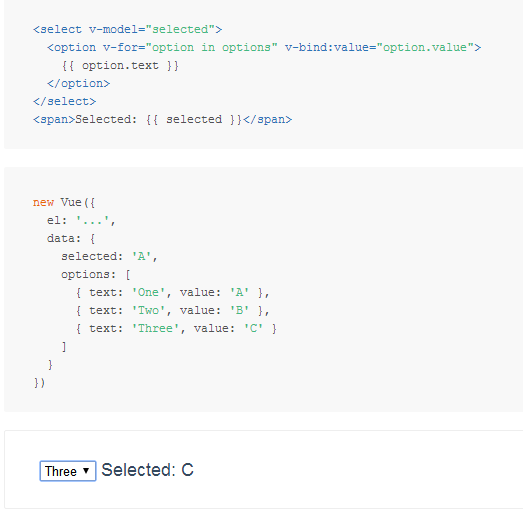
1. 下拉框，选择一项：



下拉框，选择多项，可以绑定到同一个数组：



动态渲染下拉框，结合v-for实现



单选框、复选框、下拉框被选中时，对应的value



**练习**：将个人信息页面的性别选择部分实现，要求向后台发送的性别数据为：男用0表示，女用1表示，如下图所示



3.8 v-bind 绑定属性

v-bind:class=’bg’ 简写为:class=’bg’

v-bind:class=’{bg:show}’

v-bind:class=’[col,bacg]’



也可以使用三元表达式控制样式的切换

QQ截图20170803180203.png

**练习**：窗口向下滚动到大于200的时候，给导航条添加样式，固定到窗口顶部，使用v-bind:class实现

v-bind:style=’styleObject’



v-bind:style=’{color:activeColor,fontSize:fontSize}’



**练习**:写四个li，鼠标经过奇数个li的时候，背景颜色变为黄色，鼠标经过偶数个li的时候，背景颜色变为蓝色

3.9 v-once 元素和组件只渲染一次，以后的渲染都视为静态页，有助于优化

3.10 v-clock 和display:none一起使用，隐藏未编译的Mustache标签

例：



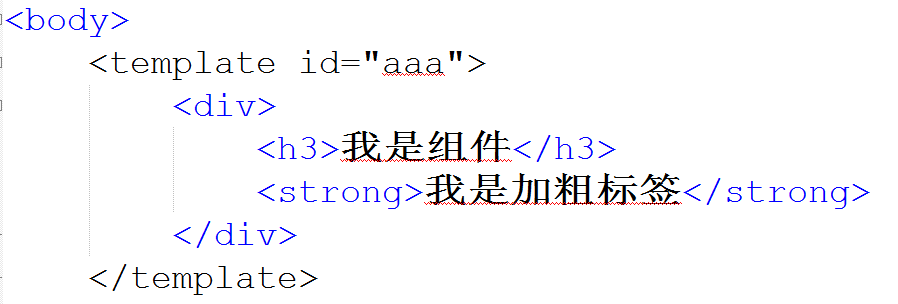
在未渲染成功之前，不会显示{{}}

**四 组件**

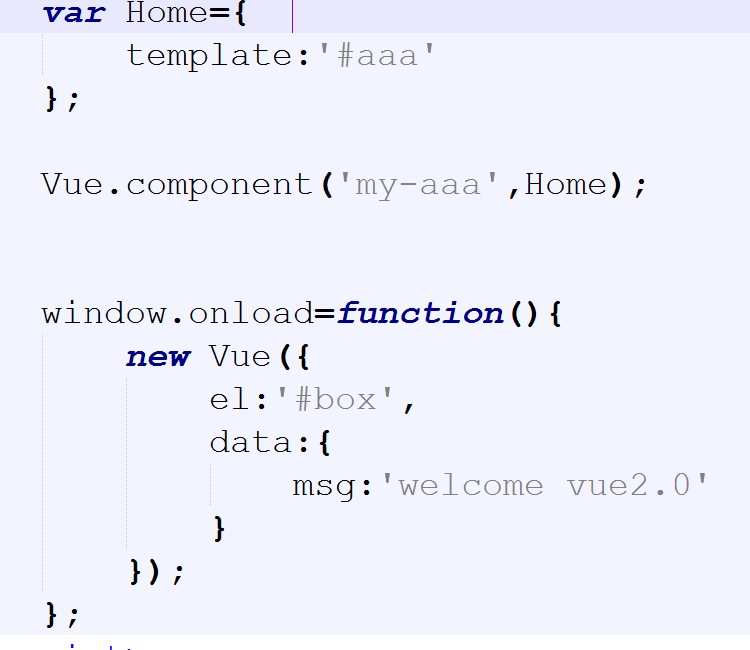
任何一个页面都可以看作是一个组件，组件可以封装重复使用的html代码

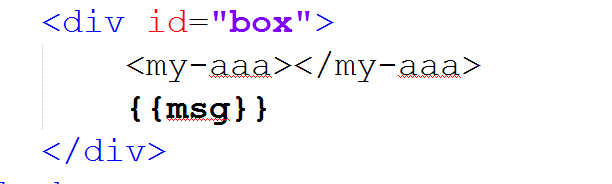
组件的使用分为：

4.1注册组件



4.2使用组件

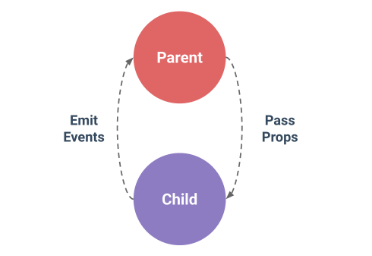




**注意**：在子组件中数据data必须定义成一个函数

**练习**： 有三个人的名字，点击每一个人名的时候弹出一个展示个人信息的遮罩层，要求使用组件实现

4.3组件之间的数据传递：



1. 父组件->子组件传递数据，通过属性props实现

父子组件的关系类似于标签的嵌套关系，外面的是父组件，嵌套的是子组件

**练习**： 将父组件里面的数据a：‘这是父组件的数据’ 展示在子组件中，点击子组件的按钮，将接收的父组件的数据a的值改为‘被改变了’

1. 子组件->父组件传递数据，通过$emit $on实现

**练习**： 将子组件的a数据发送出去，在父组件中接收数据，并展示出来

1. 兄弟组件之间的数据传递，也可以使用$emit $on实现

**练习**： 将组件a和组件b的数据发送出去，由组件c接收并展示出来

**五 路由**

路由是用来实现跳转链接的，通过vue-router实现

<https://router.vuejs.org/>

5.1安装

下载vue-router或者使用cdn都可以



5.2使用

使用<router-link to=’’></router-link>代替a标签

使用<router-view></router-view>提供路由展示的出口

例：

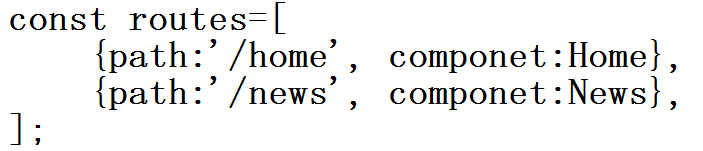


5.3创建路由

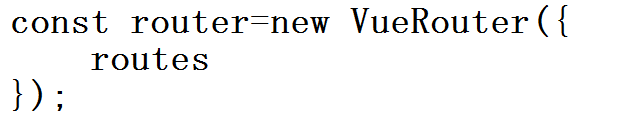
5.3.1创建组件

5.3.2配置路由

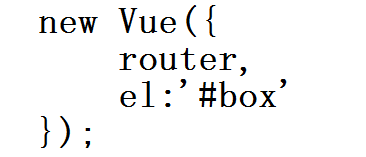
1. 配置路由



1. 生成路由实例



1. 挂载到vue上



5.3.3路由重定向

重定向的意思是：当用户访问任何一个页面的时候，都会跳转到/home页面

{path:'\*', redirect:'/home'}

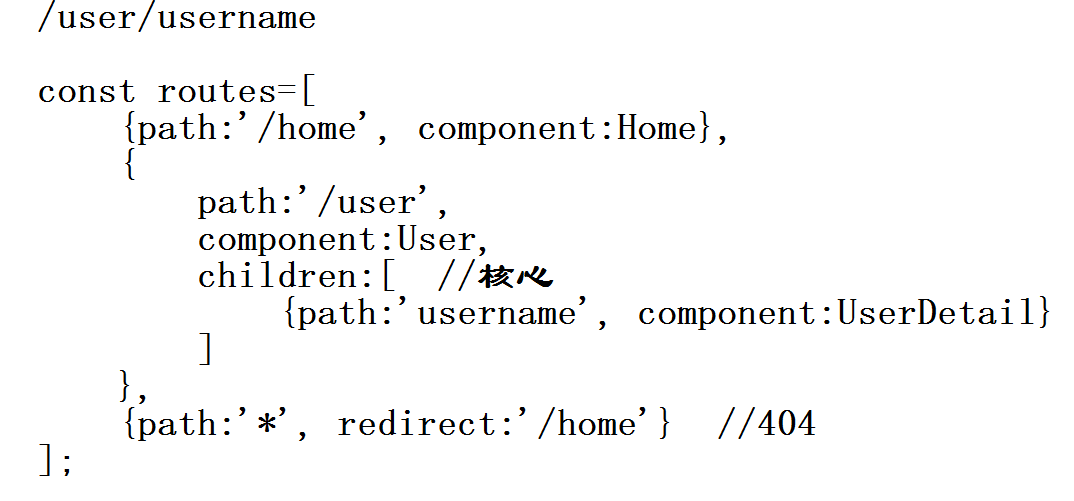
从/a重定向到/b



5.3.4路由嵌套

有的组件中可以展示其他的组件路由，这时就要用到路由嵌套

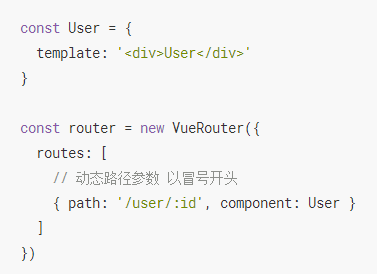
例：从/user中跳转到/username



5.3.5带参数的路由



使用方法：



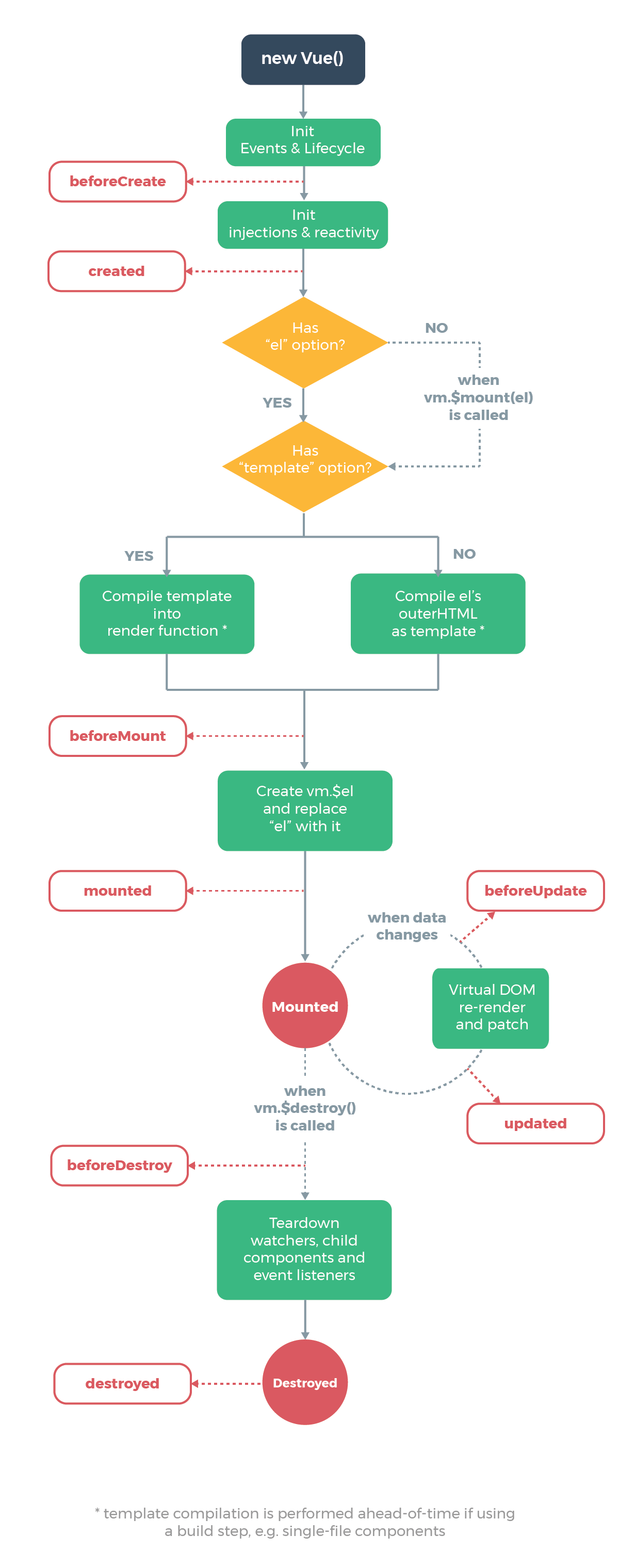
**练习1**：在组件a中使用链接，跳转到组件b和c

**练习2：**配置一个带有下拉菜单的导航

**练习3：**写三个路由，根据姓名和年龄跳转到不同的个人中心页面

**六 生命周期**

生命周期即钩子函数，把一个网页渲染完成的过程分为几个阶段，即生命周期



6.1 beforeCreate 实例创建之前

6.2 created 实例创建之后调用，挂载未开始

6.3 beforeMount 挂载之前

6.4 mounted 元素已挂载到实例上

6.5 beforeUpdate 数据更新时调用

6.6 updated 数据更新完以后调用

6.7 activated keep-alive 组件激活时调用

6.8 deactivated keep-alive 组件停用时调用

6.9 beforeDestroy 实例销毁之前

6.10 destroyed 实例销毁之后

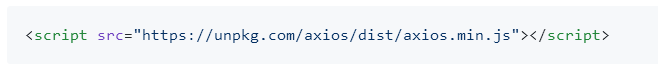
**练习1**：页面打开的时候弹出一个对话框，提示：欢迎您的到来

**七 交互 Axios**

<https://github.com/axios/axios>

7.1 安装

使用cdn



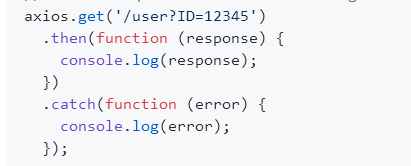
7.2 用法

axios代替了原来vue中的vue-resource,交互的方法不变

7.2.1 get请求

axios.get(url,data).then().catch()

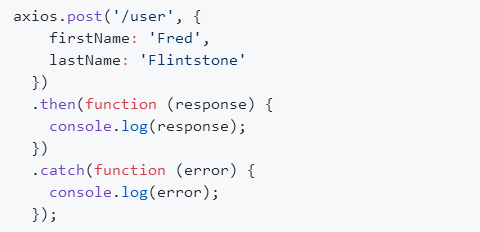
例：



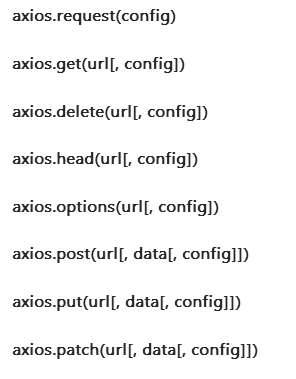
7.2.2 post请求

axios.post(url,data).then().catch()

例：



7.2.3 所有的请求方法：



第二阶段

**一 环境搭建**

1.1推荐使用vue-cli搭建环境，它会帮我们所需要的模板框架

首先要确保已经安装了node.js

http://nodejs.cn/

使用命令行搭建

npm install –g vue-cli

**＊建议安装npm淘宝镜像，或者是使用yarn**

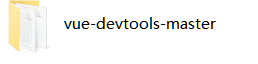
1.2 建议在你的浏览器中安装Chorme的扩展程序vue-devtools，帮助我们对数据结构进行解析和调试

a) 下载

<https://github.com/vuejs/vue-devtools#vue-devtools>

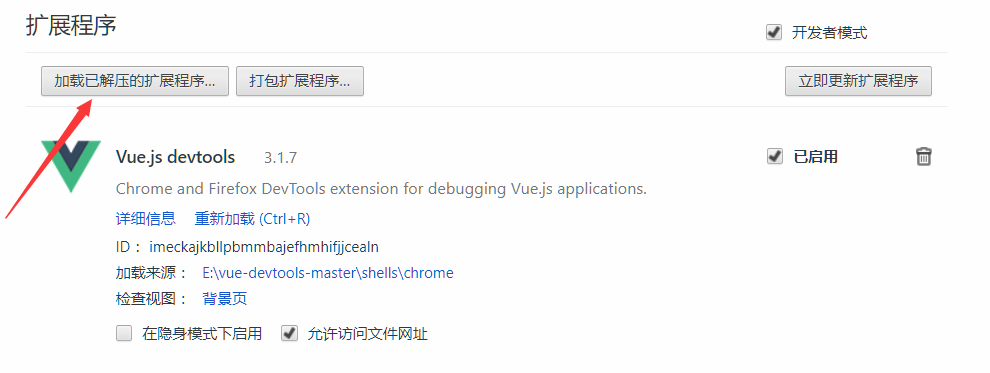
b) 修改配置

从github下载完后，找到下载的vue-devtools-master包



找到shells>chrome>mainfest.json，把里面的"persistent":false改成true

c) 添加chrome扩展程序



d)打开创建的vue项目，用控制台即可调试vue项目

**二 创建组件**

搭建好环境后，就可以在原来的基础上增加组件，完成项目

**三 插件**

项目中也可以结合一些基于vue的插件来完成功能

Element-ui <http://element-cn.eleme.io/#/zh-CN>

Mint-ui <http://mint-ui.github.io/#!/zh-cn>

Iview <https://www.iviewui.com/>

Wepy <https://github.com/Tencent/wepy>

**四 Vuex**

4.1 简介

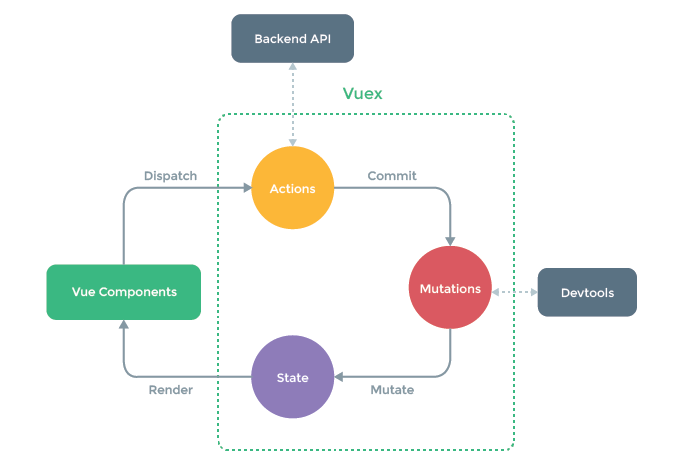
Vuex相当于一个前端数据库，专门用来存放公共的数据，比如A模块要用数据data，B模块也要用数据data，所以我们可以把data分离出来存放到Vuex中，任何一个模块都可以拿来使用

4.1.1 Vuex 是一个专为 Vue.js 应用程序开发的状态管理模式。它采用集中式存储管理应用的所有组件的状态，并以相应的规则保证状态以一种可预测的方式发生变化。Vuex 也集成到 Vue 的官方调试工具 [devtools extension](https://github.com/vuejs/vue-devtools)，提供了诸如零配置的 time-travel 调试、状态快照导入导出等高级调试功能。

4.1.2 当多个组件共享状态时，原来的组件间的数据传递方式明显会力不从心

对于多层嵌套的组件原始的数据传递方式会非常繁琐，尤其是兄弟组件的数据传递无法进行

因此可以把组件的共享状态抽取出来，以一个全局单例模式管理



4.2 什么情况下用vuex

如果要构建中大型的项目，需要在组件外部管理状态，vuex可以发挥作用，如果项目比较小，不建议使用vuex

4.3 安装

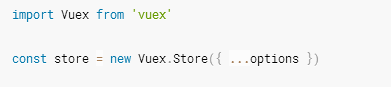
cnpm install -S vuex

4.4 配置

import Vuex from 'vuex'

Vue.use(Vuex)

4.5 创建store

store相当于一个公共的仓库，用来存放公共的数据、方法、事件

在根组件中注入store



所有的子组件通过this.$store都可以访问到

**＊**store中至少要有两项：state和mutation

4.6 用法

既然数据被分离出来，所以对于数据的改变和存储用以前的办法是行不通的，vuex提供了专门用来改变、存储、获取数据的方法

4.6.1 Mutation 用来向仓库中存入数据，也就是修改state里面的数据

vuex中不能直接改变store的状态，必须通过mutation来改变



**＊**mutation 默认第一个参数是state，还可以接收其他的参数

**＊**mutation里面的函数就是进行状态更改的，例如这里的increment(),并且回调函数的第一个参数是state

**＊**state是用来初始化状态的，所有状态都要在state中注册

commit触发 mutation 🡺 state

4.6.2 Action 负责处理数据的，处理完的数据会扔给mutation，由mutation保存到state中

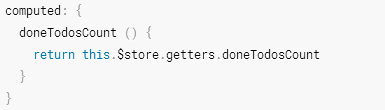


提交给mutation

dispatch 触发action 🡺 commit触发 mutation 🡺 state

4.6.3 Getter 是个纯函数，返回state中的数据，只有一个参数state，谁用数据，谁就可以从getters中获取数据

在计算属性computed中获取getters

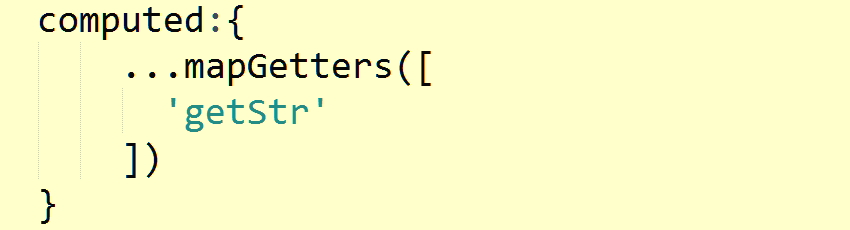


除了使用原始的方法获取数据以外，也可以使用vuex提供的辅助函数mapGetters来获取数据。

例：

先引入mapGetters





使用辅助函数获取数据在写法上更加简洁

vue-cli脚手架 ：单页面应用环境一次性搭建

1. npm i vue-cli –g
2. 安装cnpm (自始至终只用一个，不能和npm混用)

npm install -g cnpm --registry=https://registry.npm.taobao.org

1. 创建一个基于 webpack 模板的新项目 在www目录下

vue init webpack my-project

1. cd my-project
2. cnpm i
3. npm run dev
4. package.json 项目说明文件 script、dependencies、devDependencies
5. index.html 项目起始文件
6. .babelyc babel-loader的配置文件
7. node\_modules 包安装目录
8. config 分别配置dev（开发）prod（生产）模式
9. build webpack的配置文件
10. src 主要开发目录，我们都在此目录下开发
    1. assets 存放静态文件 图标/字体图标/logo
    2. components 组件文件夹
    3. router 路由组件
    4. App.vue 主组件 全局组件 放置公共样式/公共组件
    5. main.js 入口文件
11. static文件夹 演示文件/图片 使用时候路径： static/banner.jpg

公共组件、页面级组件

路径里面 @==‘./src/’

vue组件

template：HTML模板

script: vue组件定义 以ES6模块方式定义

style :CSS 定义的样式只针对当前组件，不会影响其他组件

ajax

1. 借助于axios组件处理ajax
2. 自己封装一个ajax方法
3. 使用jQuery的$.ajax()

# [vue-axios](https://github.com/imcvampire/vue-axios)

axios 返回的JSON数据是封装的数据，多了一层data

vue处理跨域请求：

1. webpack-dev-server 基于express构建的服务器
2. 网络代理 vpn(翻墙)

配置文件在config/index.js

现在用了vue-cli些vue项目，也知道是在config的index.js文件里面更改

proxyTable: {

'/api': {

target: 'http://www.api.com/api',

changeOrigin: true,

pathRewrite: {

'^/api': ''

}

}

},

然后调接口的时候

this.$**http**.get('/api/menu/get\_list').**then**(**function**(**data**){

console.log(data)

})

'/api/menu/get\_list' 🡺 http:// localhost:3000/api/menu/get\_list

POST 事件驱动 methods

GET vue生命周期mounted

vue UI框架

element-ui 饿了么团队出品 PC端管理系统

iview 主要服务于 PC 界面的中后台产品

Weex /vux vue手机端框架

1. Vue项目功能开发完成，最终要打包（npm run build），打包完成之后，会在目录下面生成一个dist文件夹
2. 将dist文件夹里面的文件，放置到服务器的静态目录，如果是nodejs服务端，应该放置到public文件夹下