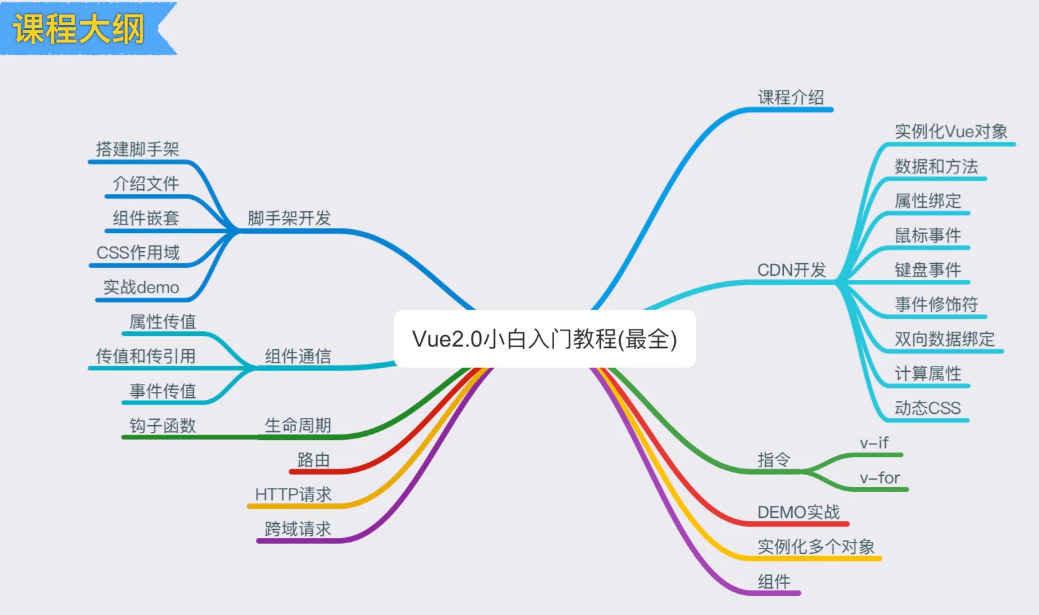
# 什么是vue

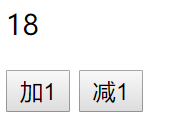


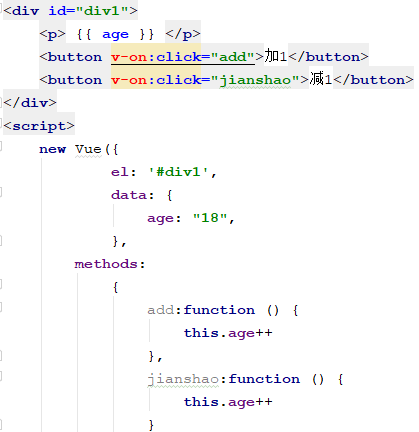


vue 是js的一个框架.

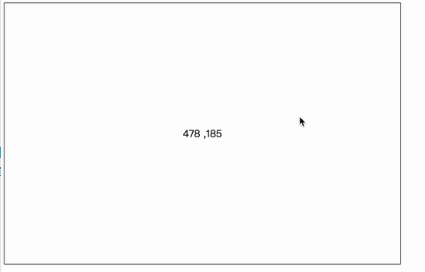
=======================================🡺

事件:

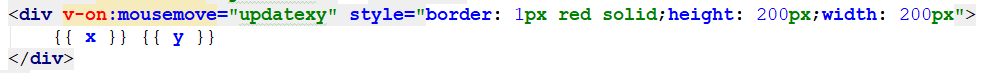


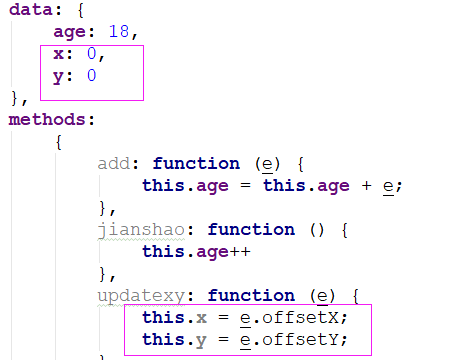


=======================================🡺

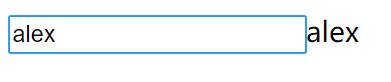


鼠标在里面显示坐标:

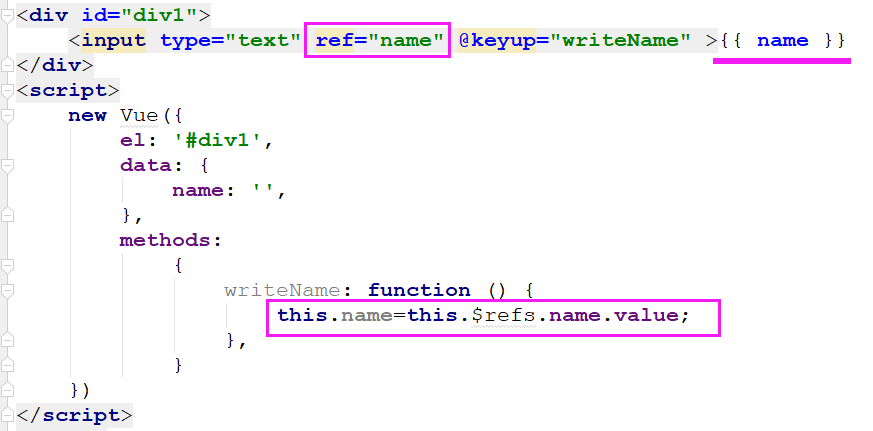




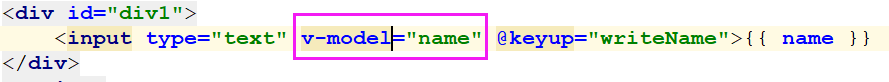
=======================================🡺



左边文本框输入什么,右边span显示什么,代码:



方法2. 用v-model='属性名' 效果一样



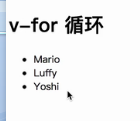
=======================================🡺

循环遍历: 可以遍历数组和字典,和字典中的单个对象

遍历数组:

characters是[]数组



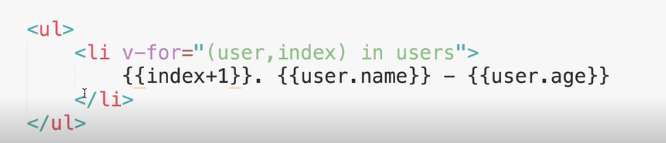


遍历字典users:



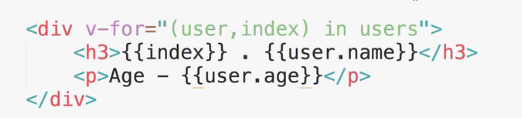


2个参数, 一个是对象,一个是索引:





不是只能遍历li,也可以遍历div:



也可以遍历对象:



=======================================🡺

关于this指向的问题，与vm实例没有任何关系。而是与箭头函数和普通函数的区别。

1、es5的普通函数，this指向是指向了当前调用的对象. let v1=new Vue( ), this指向v1 .

2、箭头函数的this指向它的调用者所在的类的父类.即window.

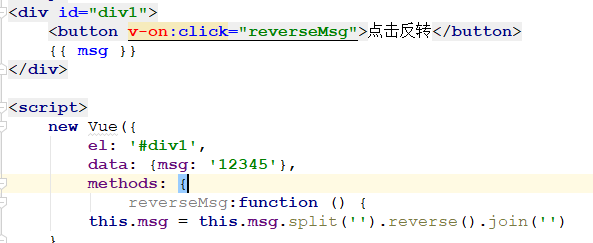
=======================================🡺

# 指令系统

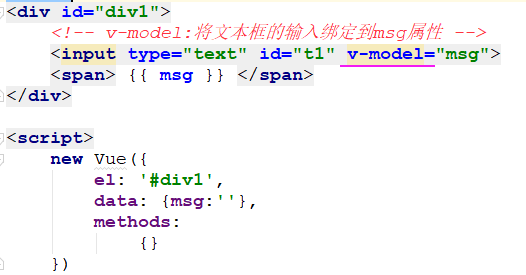




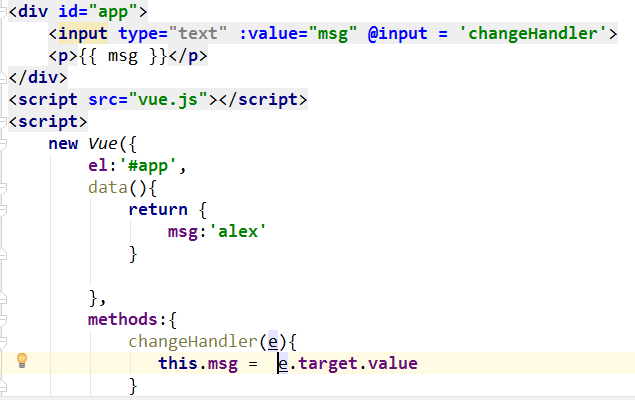




：

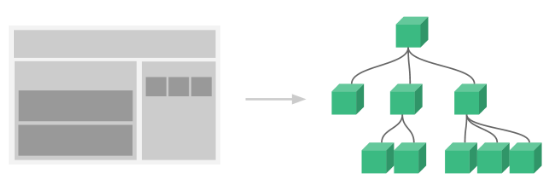


如果不用v-model用普通代码实现:



# 组件:

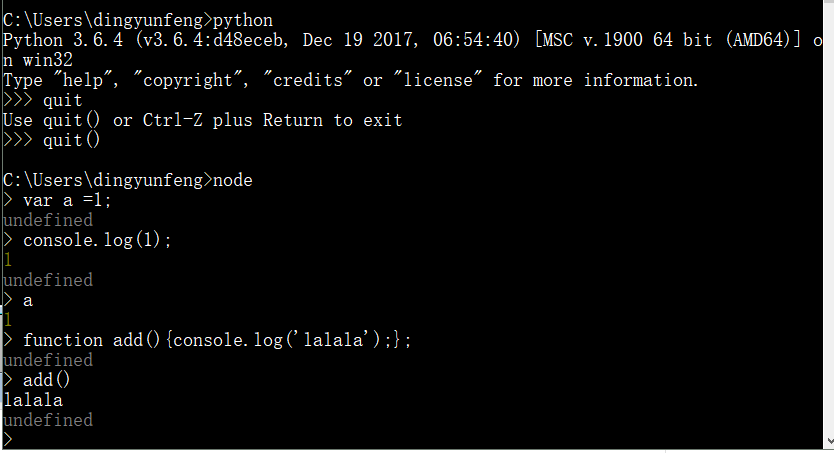
任意类型的应用界面都可以抽象为一个组件树：



在 Vue 里，一个组件本质上是一个拥有预定义选项的一个 Vue 实例。在 Vue 中注册组件很简单：

定义组件：在componet中定义组件：在componet里面的template中写组件的内容

Vue.component('组件名', {  
 template: '组件代码'  
})



node相当于在cmd里执行python。 调用解释器。

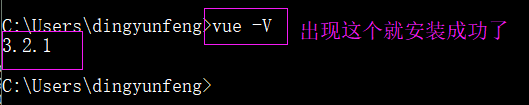
npm是安装命令，类似py中的pip3

2个ctrl c 退出。

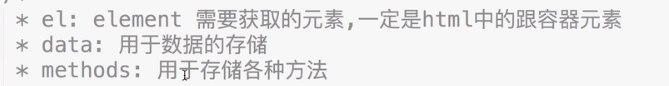
### 

步骤：安装node.js 8版本—安装淘宝镜像—安装脚手架。

### 



=========================================🡺



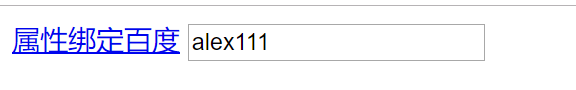
=========================================🡺

属性绑定:

给标签属性绑定一个值,不能用href={{ data的website属性 }},要用v-bind: href={{ data的website属性 }}

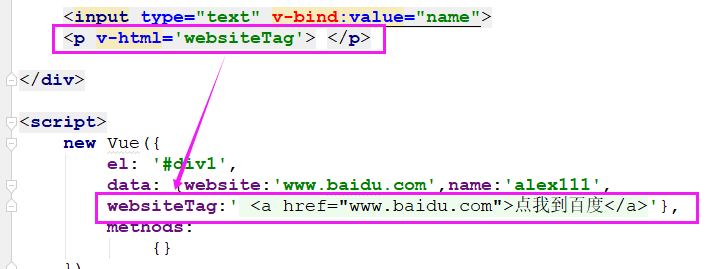


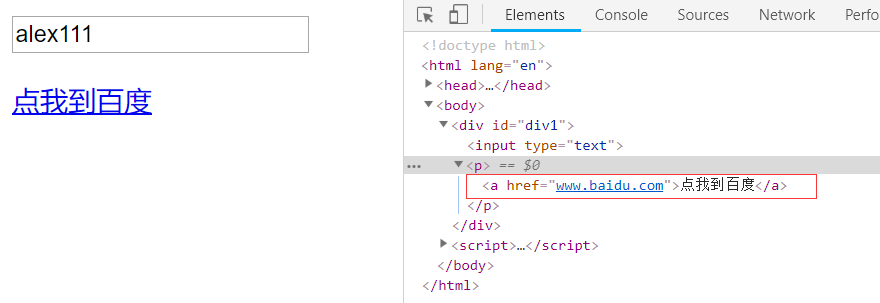




=========================================🡺

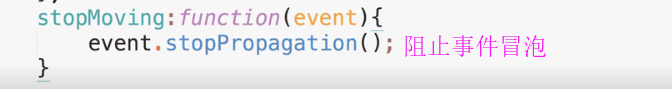
v-html: 说明这里要放一段html代码正常渲染





=========================================🡺

事件修饰符:



当鼠标在这个span中移动的时候,阻止事件冒泡,也就是,阻止这个span中的事件.



事件修饰符有:

.stop

.prevent

.capture

.self

.once

.passive

<!-- 阻止单击事件继续传播 -->

<a v-on:click.stop="doThis"></a>

<!-- 提交事件不再重载页面 -->

<form v-on:submit.prevent="onSubmit"></form>

<!-- 修饰符可以串联 -->

<a v-on:click.stop.prevent="doThat"></a>

<!-- 只有修饰符 -->

<form v-on:submit.prevent></form>

<!-- 添加事件监听器时使用事件捕获模式 -->

<!-- 即元素自身触发的事件先在此处理，然后才交由内部元素进行处理 -->

<div v-on:click.capture="doThis">...</div>

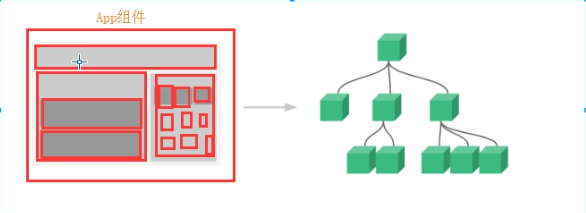
<!-- 只当在 event.target 是当前元素自身时触发处理函数 -->

<!-- 即事件不是从内部元素触发的 -->

<div v-on:click.self="doThat">...</div>

使用修饰符时，顺序很重要；相应的代码会以同样的顺序产生。因此，用 v-on:click.prevent.self 会阻止所有的点击，而 v-on:click.self.prevent 只会阻止对元素自身的点击。

========================================🡺

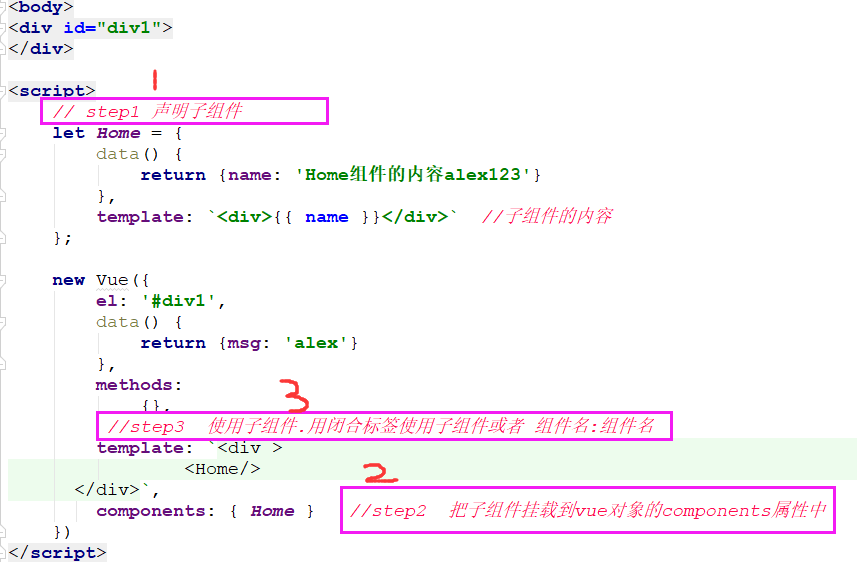


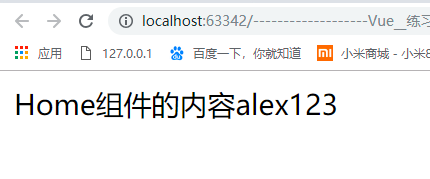
一个页面就是由一个组件树构成的.组价和组件嵌套,每个组件中有自己的属性和方法.

模块化开发就是组件化开发.

========================================🡺

定义和使用组件:

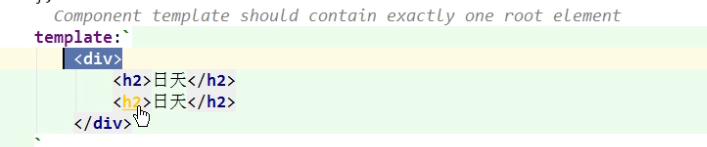




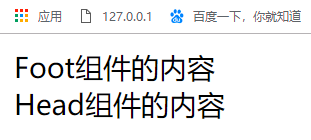
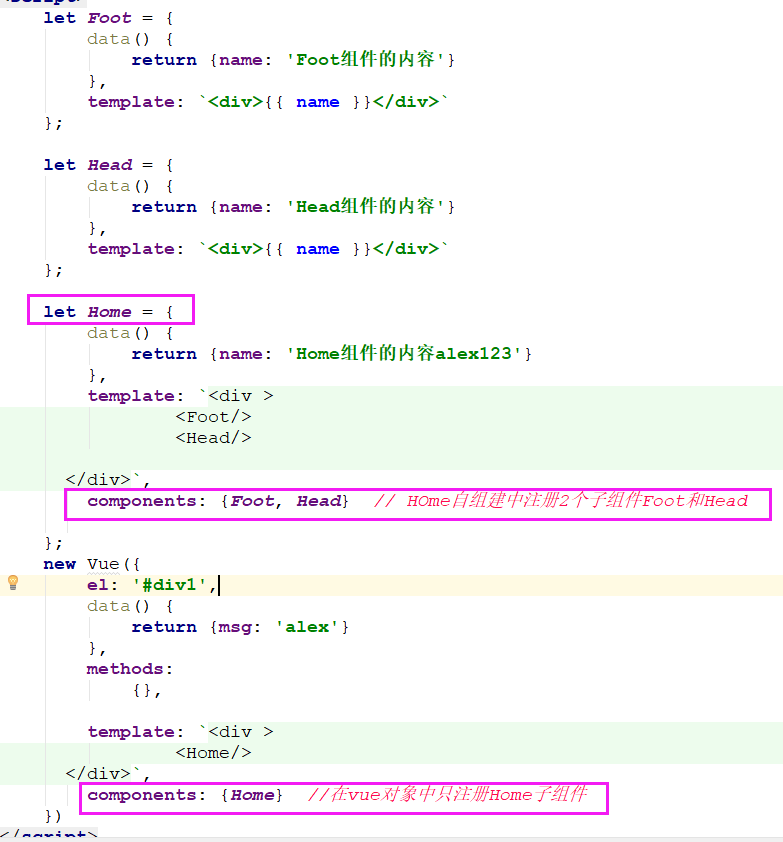


=========================================🡺

组件的模板内容必须要用一个容器包裹起来做为一个整体: 例如用div包裹两个h2



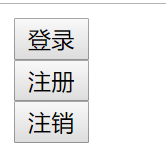
=========================================🡺



=========================================🡺

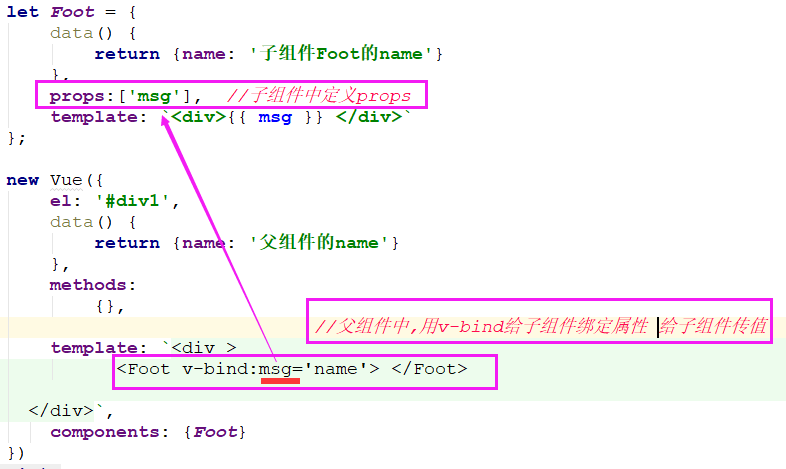
slot

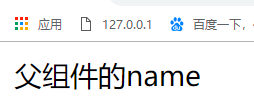




==========================================🡺

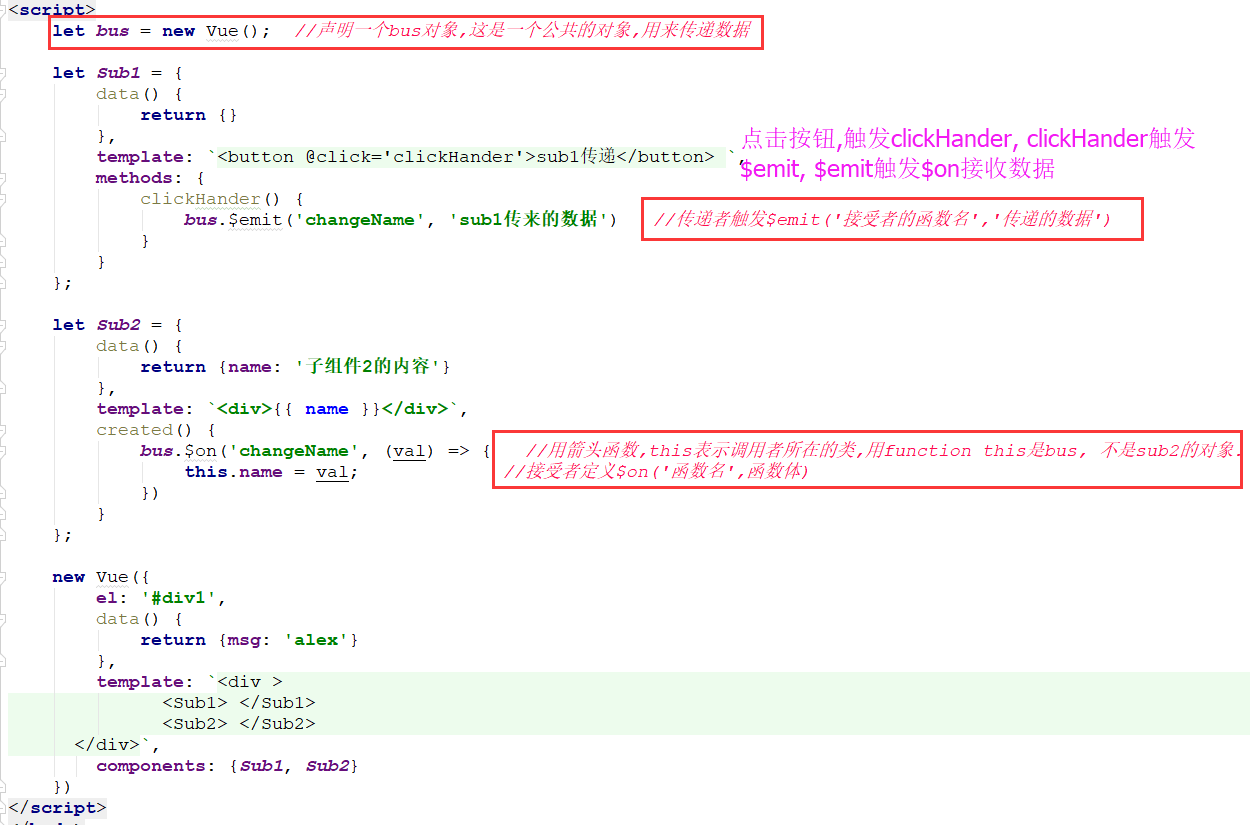
父组件向子组件传值:





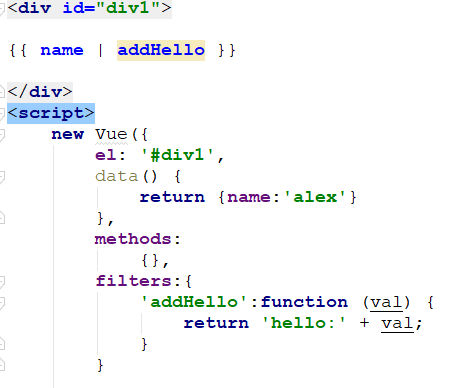
=============================================🡺

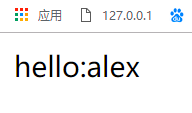
平行传值:



========================================🡺

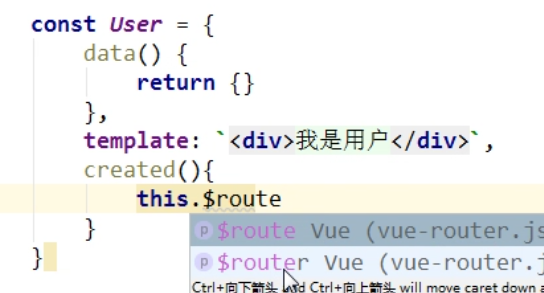
过滤器:





========================================🡺

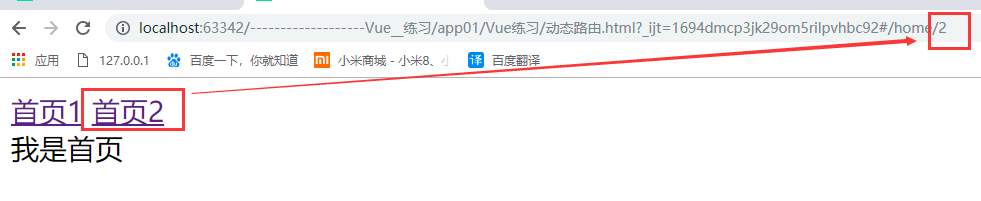


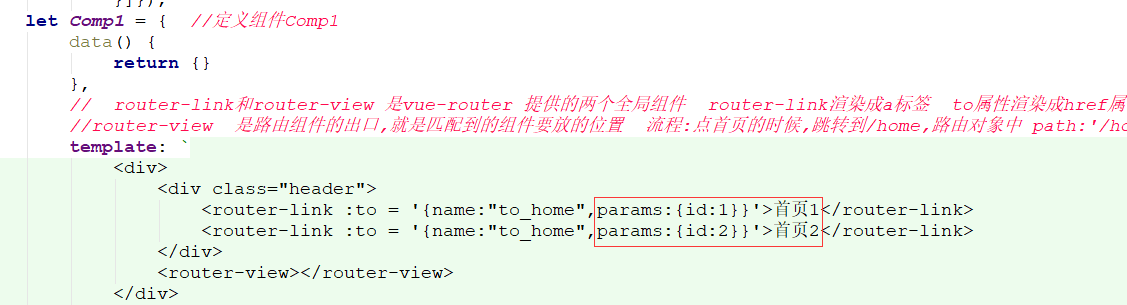


$router 路由对象 在组件内部用

=========================================🡺

带参数的路由:





============================================🡺

搭建路飞的首页和免费课程:

在create()方法中发ajax从后台获取菜单内容放到data的categoryList中,然后循环categoryList显示出来.

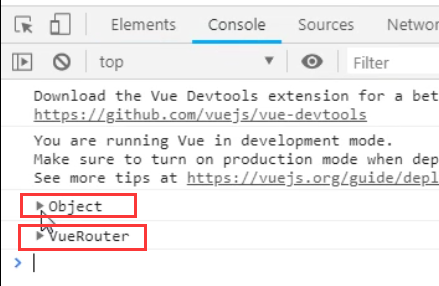


============================================🡺



============================================🡺

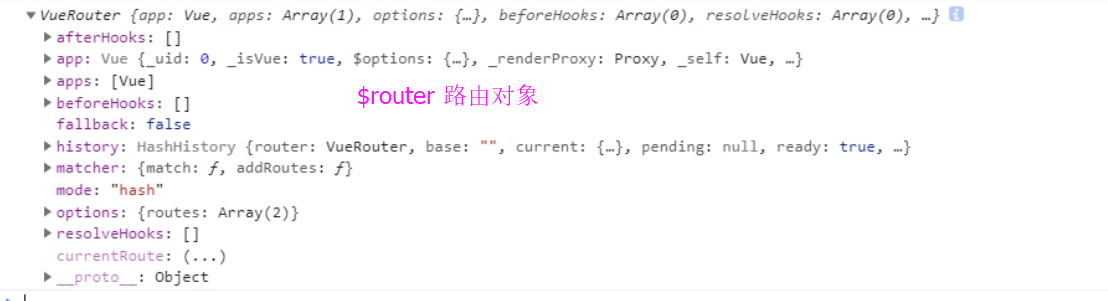
$route vs $router



$route: 路由的所有信息



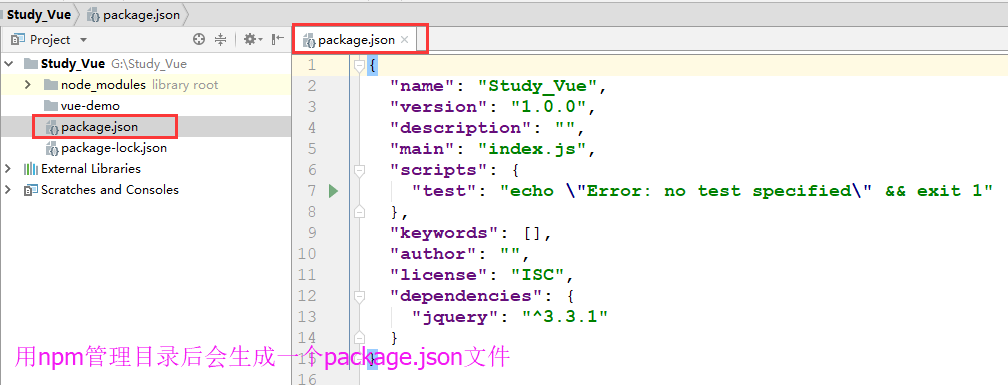
$router:路由对象:

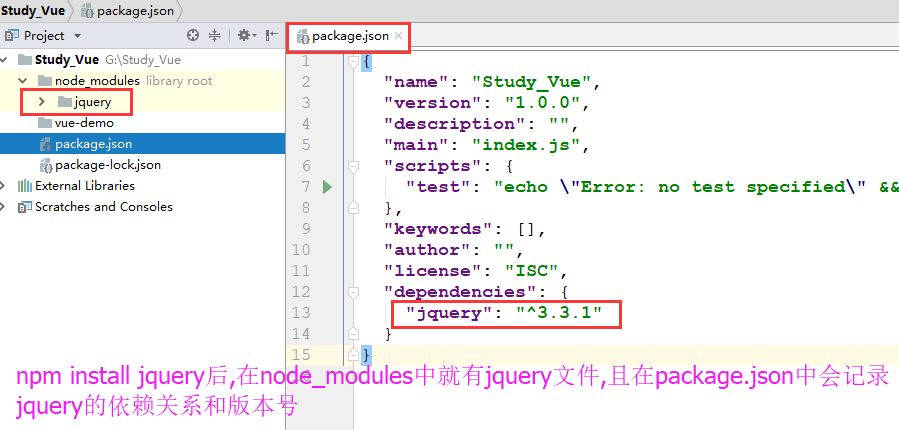


============================================🡺

使用npm:

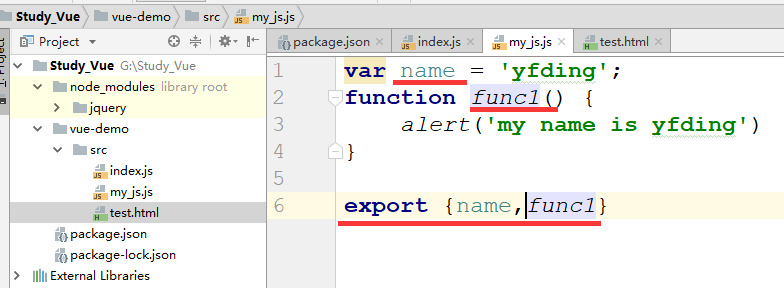


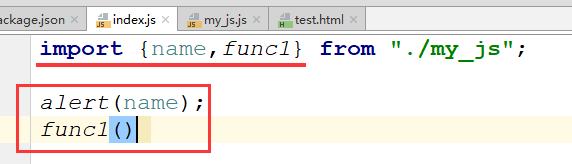




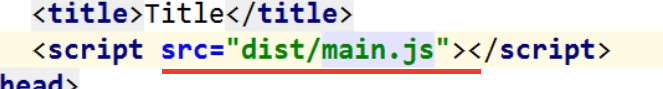
使用webpack:的导出和导入

1. 在一个js文件中写变量和函数名,然后导出



1. 在另一个js文件中导入:
2. 

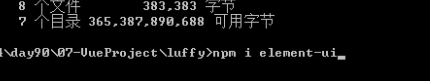
在html中导入这个dist下的main.js,以为打包后的js在dist的main.js下

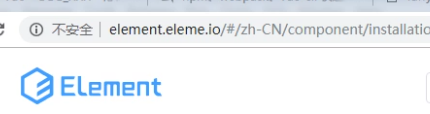


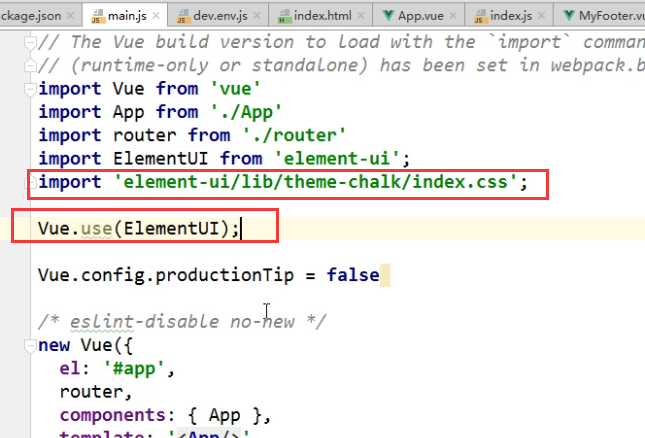
要webpack --mode development打包了才能使用.

===============================================🡺

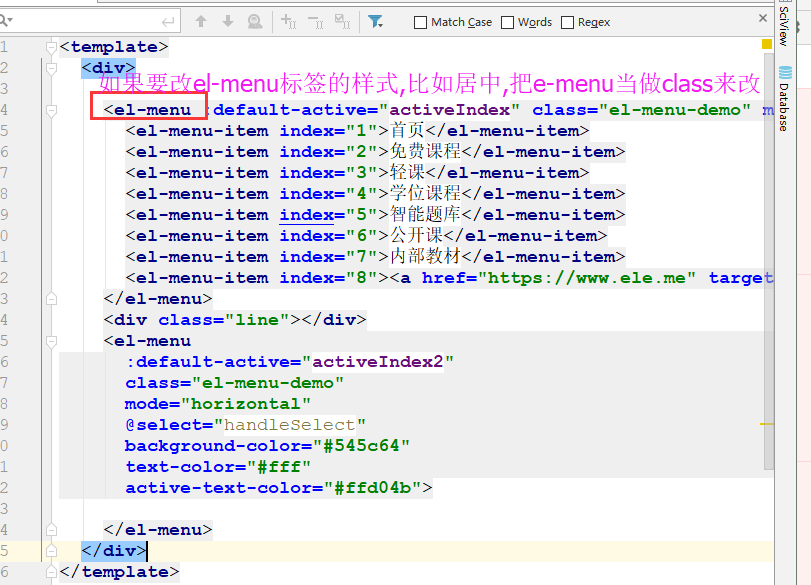
从这里开始没有练习:下载element-ui:







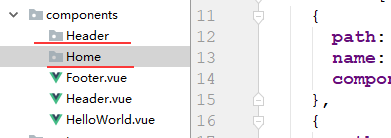
=============================================🡺





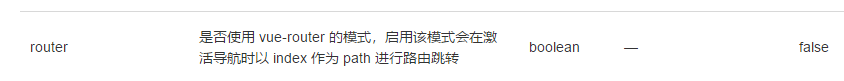
=============================================🡺

建2个文件夹,home中的组件放在Home文件夹下,Header中的组件放在header文件夹下.



===============================================🡺

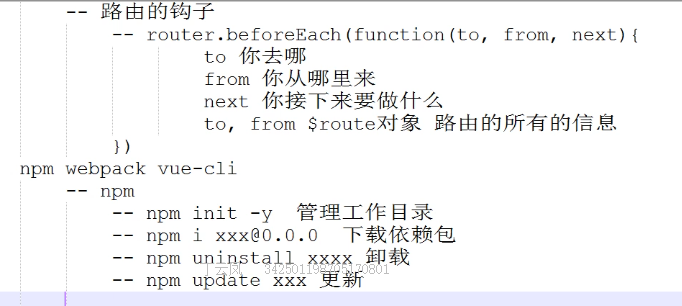
elemnt中的的标签加上:router=true表示启用router模式:

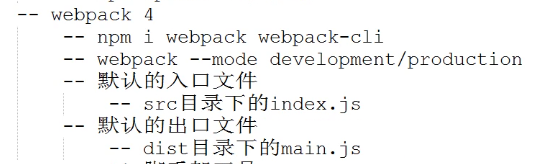


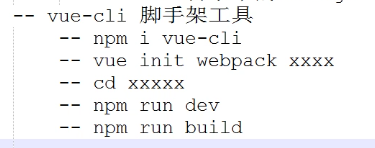


设置:router=true后,index就表示path

===============================================🡺

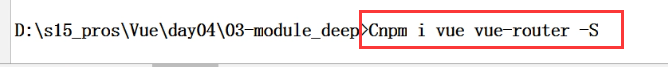




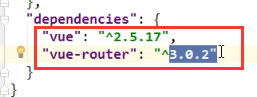


===============================================🡺

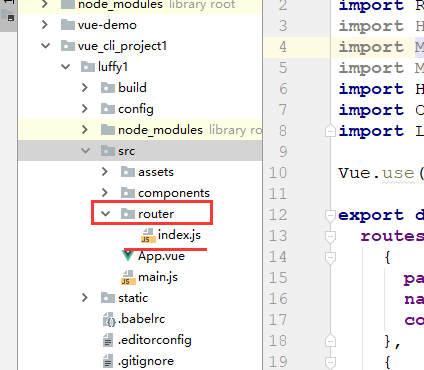
在pycharm中下载vue-router



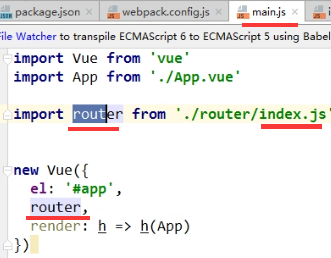
在package.json中可以看见已经下载的包:



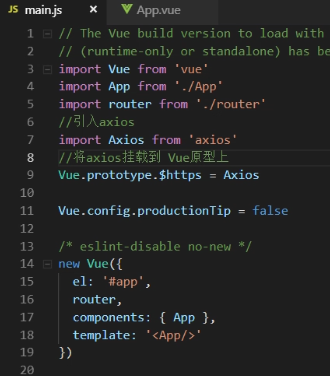
===============================================🡺



如果把所有的路由写是在main.js下代码太多. 在router目录下的index.js中写所有的路由.然后main中导入index.js中的router



===============================================🡺



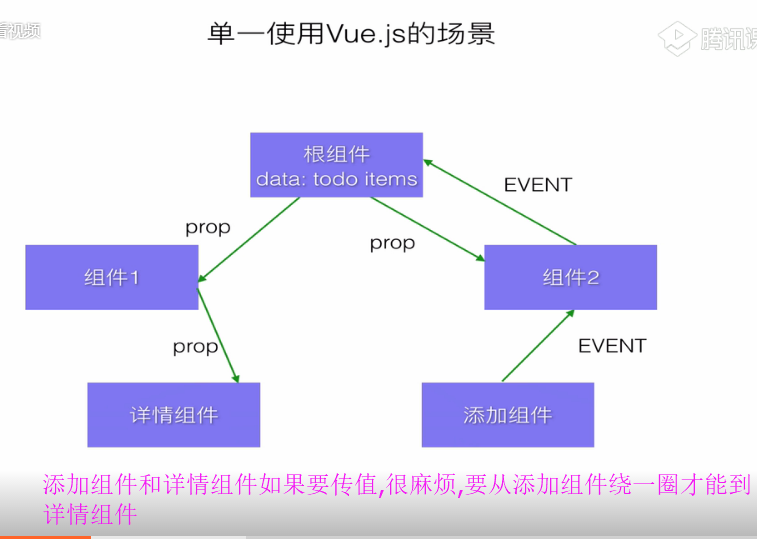
===============================================🡺

vuex是什么:

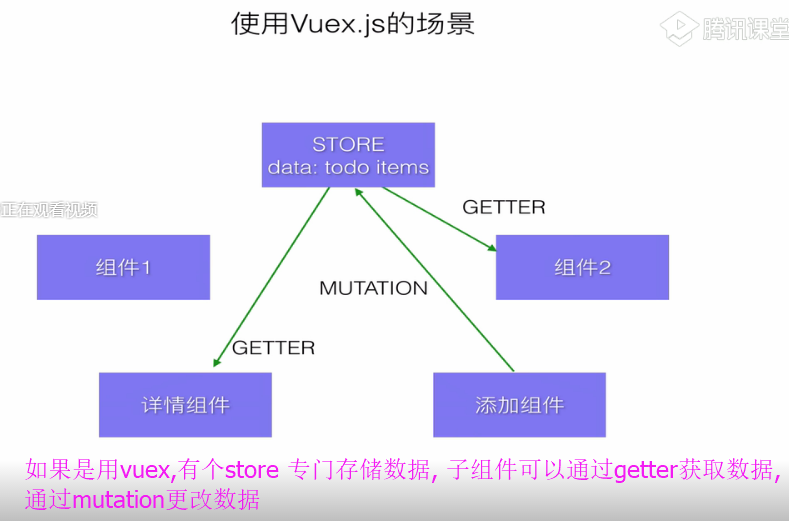


4个组件都可以使用store仓库中的东西

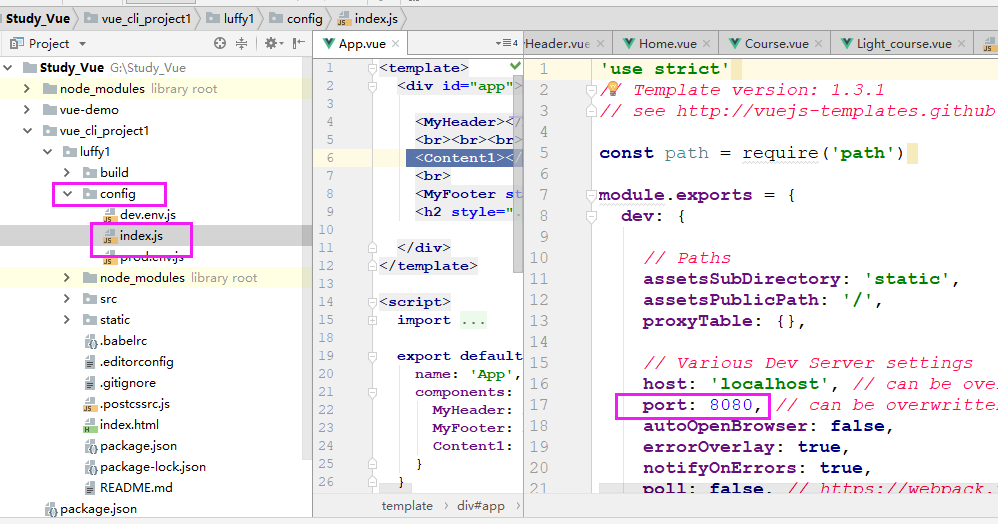
vuex是什么?vuex是vue中管理数据状态的一个库. 在store中的数据所有组件都可以使用.如果不用vuex,组件之间虽然也可以传值,但是比较麻烦..如下图.如果添加组件要传值到详情组件需要绕一个圈才行.



如果使用vuex,通过getter就可以获取数据.

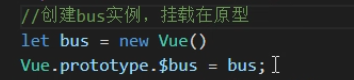


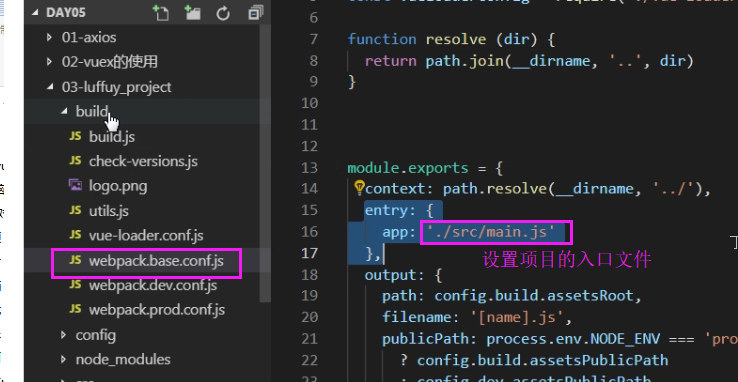
==================================================🡺



改服务器端口

=============================================🡺

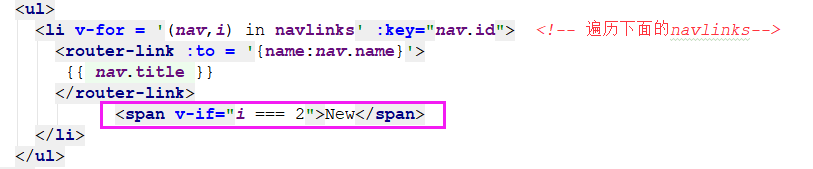


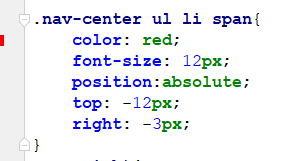




=========================================🡺





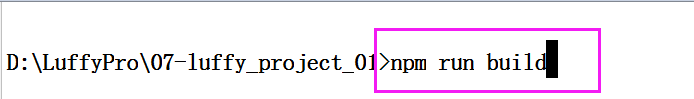


=========================================🡺

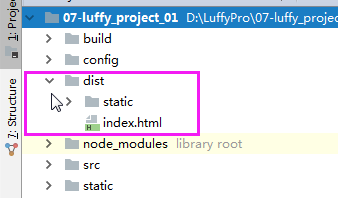


=========================================🡺

打包



打包后生成dist目录

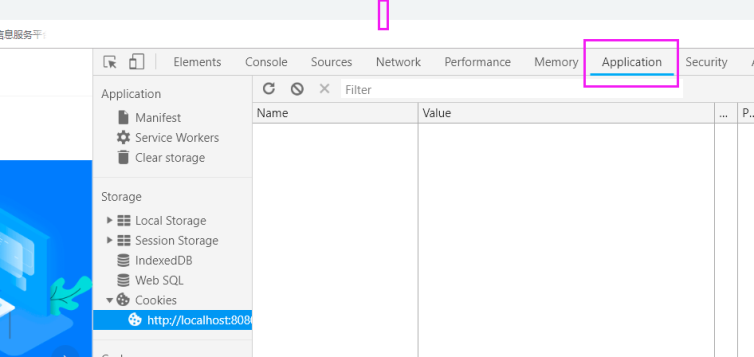


dist内容放在linx的nginx上.

实际工作会发布在2个服务器上,一个是后台的,一个是linx的nginx上.

=========================================🡺

前端cookie在application中:



local storage

session storage

=========================================🡺