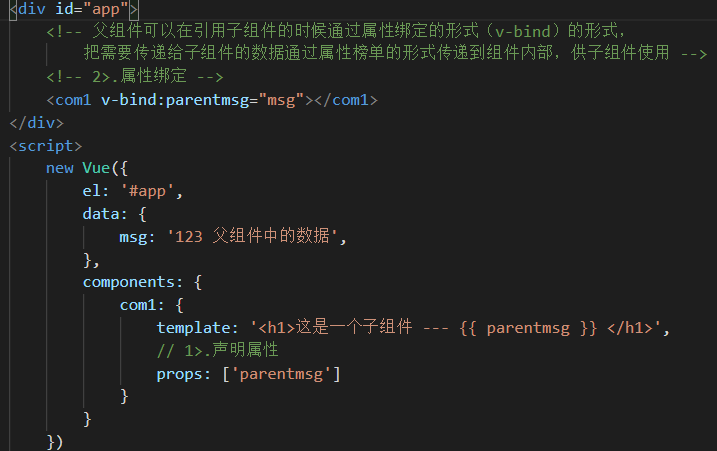
1. 父组件向子组件传值

默认情况下子组件无法访问父组件的data和methods(函数是子作用域能访问父级作用域的)

父组件可以在引用子组件的时候通过属性绑定的形式（v-bind）的形式，把需要传递给子组件的数据通过属性榜单的形式传递到组件内部，供子组件使用🡪



注意：data数据可读可写(自己身上的数据), props中的数据只读(传递的数据)

二．通过id创建的组件



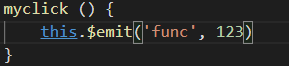
三．子组件调用父组件的方法



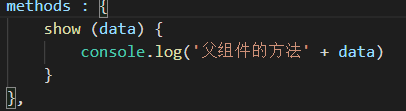
注意：子组件通过事件绑定先父组件传值通过$emit()的第二个和之后的参数进行传值，

父组件通过参数来接受(延伸可以实现子组件得data向父组件的data传值)🡪

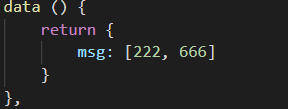
子组件:

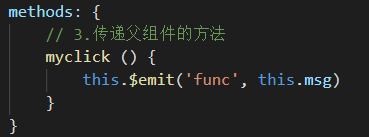


父组件:

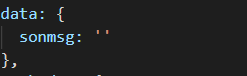


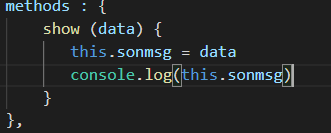
子组件中:





父组件:



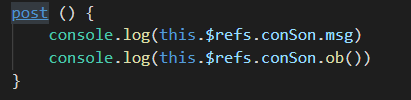


五．ref获取dom元素和组件的引用

引用哪里的在哪里加上ref=”引用标识(随便去引用时候用到)”-🡪



引用触发(一般绑定一个按钮触发)🡪



注意:1>.ref=””被引用使用和 this.$refs.引用标识.引用的东西。

2>.ref表示引用一个，refs表示引用多个。

六．前端路由和后端路由

1>.前端路由 哈希发(#/)🡪a标签差不多,不会刷新页面。

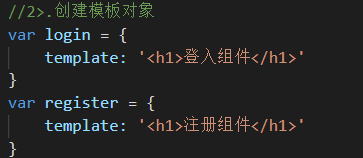
2>.后端路由 一个url对应一个服务器资源。

七.路由的基本使用步骤

1>.导入包



2.>创建模板对象



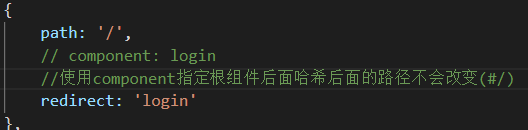
3>.创建一个router对象



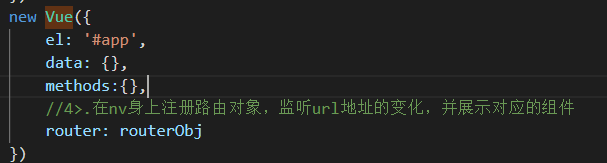
注意:1>.path后面是/引用地址不是./

2>.component值时组件的模板对象不是组件引用名称

3>.可以使用redirect指定默认路由--🡪



4>.在实例身上注册路由对象



5>.创建展示标签(vue自带的标签)



6>.引用路由



注意:地址是#path

可以使用router-link标签代替触发跳转的标签



注意:地址是to=”path”, 其标签还有个tag属性(渲染成什么标签,无论什么标签都可以触发)

八．路由原理

1>.点击按钮触发url地址的改变

2>.路由监听到进行规则匹配（匹配对象的path）

3>.匹配到的路由放到router-view标签内展示

九．路由高亮(默认路由调转会自动添加class根据其class创建样式)

路由调转默认添加的class为🡪router-link-active



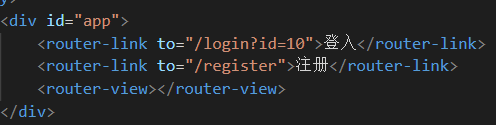
可以通过和routers平级的属性linkActiveClass属性修改其值🡪



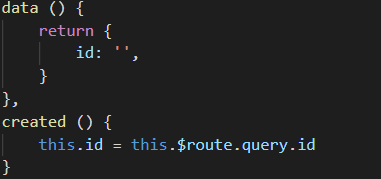


十．路由传参

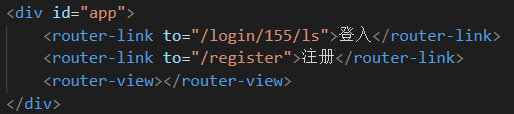
1>.利用query传参(在视图层中触发路由的标签上写传递参数的属性和属性值 ?id=10)



拿参数方式一(另一种方式看下一种传参方式params传惨)

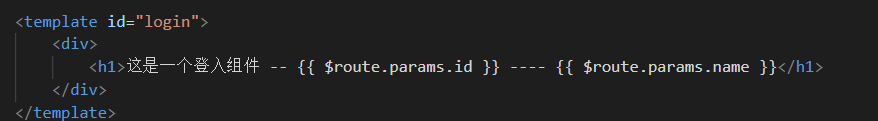


2>.利用params传参( 在视图层中触发路由的标签上写传递参数的属性值，同时在匹配规则中写上传递的是什么属性 /10, 匹配规则 /:id)





拿参数方式二(另一种方式看上一种传参方式query传参)



注意:传递的参数和匹配规则写上的属性要一一对应不然组件失效-🡪

/155 路由规则/:id/:name(失效)

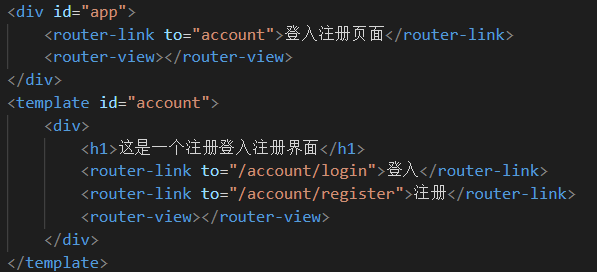
/155/ls 路由规则/:id (失效)

十一.子路由的嵌套children属性(routes里面的属性, 里面放子路由匹配规则 path不需要/)

子路由嵌套意思是在跳转路由的组件中在创建路由跳转

注意：没嵌套的是在根标签(#app)内创建跳转标签

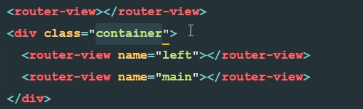
视图层中🡪

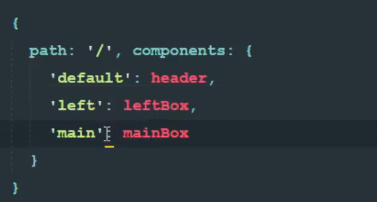


Vm层中（中间层）🡪



十二.命名视图(可以在router-view加上name属性指定下面的路由对象的components属性)🡪





注意：多个创建多个router-view标签且路由对象的component属性加了s 里面的defalt属性是默认指定的路由。components里面的属性要加’’(不加会默认加上)

Router-view就是指定的路由的展示区。