# Vue上课笔记

Vue:

Script:直接引入文件；

模块化：import Vue from ‘vue’

Vue实例：

创建实例： new Vue({参数系列});

El:element，指定vue解析的节点范围。

Data：数据，没有出现在data里面的数据，是没有实时驱动效果；

Methods：函数系列，对象类型，里面定义很多我们用的函数。

Computed：计算属性，比如：根据生日计算出年龄。

Watch：侦听器，监听数据变化，进行比较复杂的操作。

生命周期：

页面初始化的时候：

beforeCreate：

created：el是undefine，data数据加载完成。

beforeMount：找到解析的el。

Mounted：数据挂载完成

数据更新的时候：

beforeUpdate：数据改变但是还没有渲染到页面；

updated：数据渲染到页面。

销毁实例：

beforeDestroy：销毁之前，最后通过数据来控制页面效果的机会。

destroyed：销毁之后，页面效果依然存在，但是页面不可控了。

模板引擎：

文本信息：{{varname}}

属性：:class=”clname”

v-html:该指令可以显示html代码。

条件渲染：

v-if v-else v-else-if

v-show:进来会初始化所有的节点，然后控制style里面的display；

列表渲染：

v-for:渲染列表，可以是数组，可以是对象，更多的时候我们拿来操作json格式的数据。

事件：

绑定事件：v-on:click=”clickMe” 简写 @click=”clickMe”

表单数据绑定：

双向数据驱动：v-model.

Input\radio\select

Checkbox:需要数组类型。

组件：类似于标签

作用域：全局组件和局部组件；

全局组件：

Vue.component(‘my-tag’, {

Template:`<div class=”mydiv”>展示的结构</div>`,

Data:function(){

return {}

},

Methods:{},

Watch:{},

Mounted:{}

})

template:必须有一个根节点。

data:必须是一个函数。

New Vue({

el:’’,

data:{},

components:{

myTag:{

和定义全局组件一致

}

}

});

单文件组件：

用于生产。

一个文件完成一个组件：包括结构、逻辑处理、样式。

后缀要求：.vue

Template:定义结构，必须要有一个根节点。

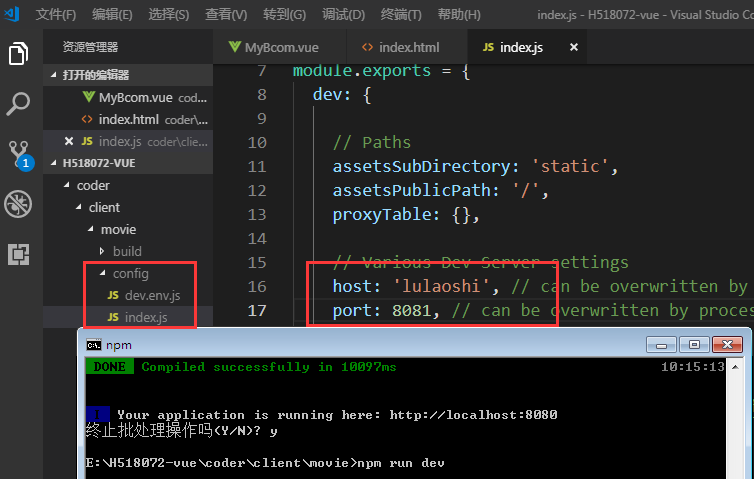
Script：定义逻辑处理以及模型。

Style：样式。Scoped表示当前样式只能作用域组件，规避了样式冲突问题。

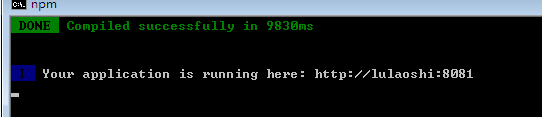
我们使用脚手架 vue-cli进行开发，一个页面对应的是一个vue组件。

修改脚手架项目的端口和访问地址：



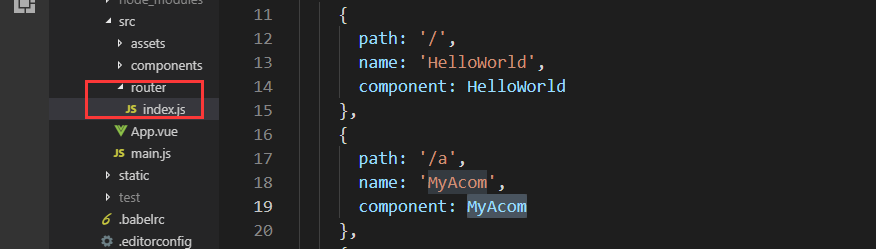


重启之后：



路由：

根据访问地址来匹配组件：路由配置文件



子组件：

第一步，引入子组件；

第二步，注册子组件到我们的父级组件。

Html不区分大小写，js严格区分大小写。

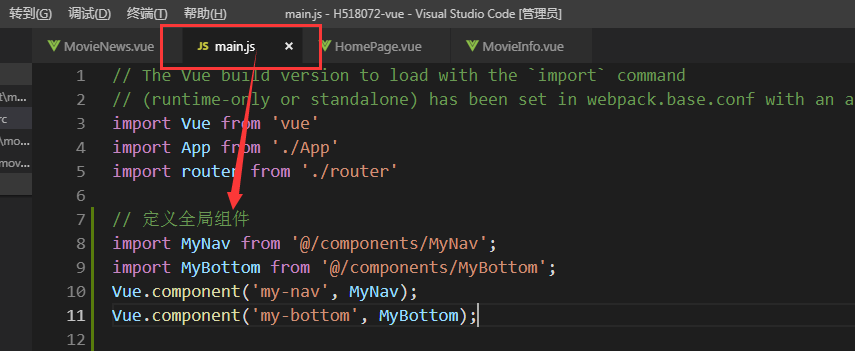
实现链接跳转：

Router-link

注册全局信息：

需要在入口文件里面完成。

入口文件：src/main.js



路由实现页面跳转：

Router-link 子组件 最总渲染出a标签

传参：

<router-link :to="'/movie?id=' + minfo.id">

Query：问号后面传值，get方式

接收query过来的值：route.query



Params:

路由配置：

{

path: '/movie/:id/:c/:d',

name: 'MyMovie',

component: MyMovie

}

<router-link :to="'/movie/' + minfo.id">

接收params传过来的值：route.params

使用UI框架实现我们的页面布局：

PC：elementUI、iviewUI等。

移动端：mintUI、vux等。

项目打包：

把我们写的组件（单文件组件）打包成浏览器认识的html、css、js代码。

配置打包路径：

