这里对vue文件的编写和查看进行特简易说明，三部分：vue文件结构，vue数据的交互，vue组件的查看

一、vue文件结构：



三部分：template，script，style

1.template是组件显示的内容，可完全按照html的内容编写，只有一个区别：template下一定是被一个元素包裹起来的，例如<template><div><span></span><span></span></div></template>，即只有在最大的div下的元素才能并列防止，template下是不能并列的。

2.script下就是当前组件的名字，数据和操作。name需要和文件名完全匹配，除了data还有一些基础的其他内容，排列如下：

<script>

import...(视情况添加)

export default {

name: ‘filename’,

components: {在其中包含template中需要使用的其他vue组件，使用前需要import},

data() {

return {

这里的值就相当于js里定义的那样，不过可以直接和template内的数据关联起来

dataname:’datavalue’,

}

},

activated: function() {

包含了当前组件加载完毕时进行的操作，即相当于js中的onload行为。

},

methods: {

functionname() {

functionbody

}

}

}

</script>

在每个方法中，this（类似java中的this）就是当前的vue对象，在方法中可直接调用data下的数据和methods下的函数的等等。

3.style

即唯一对应的css内容，一般不会使用（而是使用element ui后就可通过它自带的一些字段进行调整，因为自带的调整一般都有适配性）。

二、vue数据的交互：

1.使用data中的简单数据：在template中插入双花括号语法：{{msg}}，便会等于名字是msg变量的值

2.遍历数组数据：利用v-for进行，例如：

item就是数组内的一个个元素，index为元素的下标







3.使用methods中的方法：

通过方法调用指令便可调用对应方法，例如@click=”functionname”，点击后调用对应方法，其他方式调用可百度进行

4.表单和数据绑定变化：利用v-model

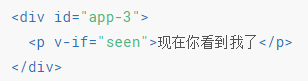
例如这个input和newTodoText绑定，此时input内输入的值就会和newT...这个变量同步变化了，其他的一些组件可能也有这一效果。



5.依据某个变量的值判断是否显示：v-if

判断boolean类型变量，true显示，false不显示

直接v-if = “msg”即可判断msg



此时这句话就会看到，如果seen为false就不会显示。

也可用条件判断实现：

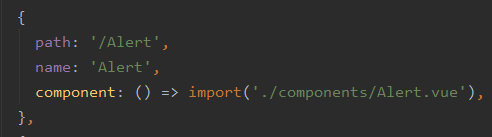
6.其他更多template和script的交互可在《vue基础语法学习文档》中的2.2.1 部分指令具体内容介绍看到，也可以百度。

三、如何显示vue组件

编写好vue组件后进入src下的router.js文件，它相当于vue项目的路由（路由器），负责配置各个组件的地址和内容：

在router常量内容下的routes数组中添加内容即可：

一个json对象就相当于一个页面，path是进入页面后通过什么路径能访问这个组件，name就和组件名字对应就好，component就是引用对应的组件。



添加好后，运行npm serve或者npm dev，这个和项目的构建方式有关，运行后从IP地址或本地就可直接访问。