学习vue 项目开发的前提是熟读文档 不求全懂 但是要知道何处有何物

vue文档开篇介绍

vue是一套用于构建用户界面的渐进式框架。它是自底层向上逐层应用。

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结合。另一方面，当与现代化的工具链

以及各种支持类库结合使用时，vue也完全能够为复杂的单页面（spa single page web application）应用提供驱动。

构建vue应用 我们可以简单的利用cdn来引入 或者是vue-min.js 文件（这两种具体可以去bootcdn或者是https://cn.vuejs.org）

或者是webpack构建 或者是vue-cli生成

vue的核心允许采用简洁的模板语法来声明式的将数据渲染进DOM的系统

el

vue通过模板语法将数据渲染到DOM中 此时vue实例中有一个el属性 它是用来指定你将要在哪个dom元素中使用vue语法

这个属性在vue根容器中是必须绑定的；

data

vue通过模板语法渲染数据 data中我们一般用来初始数据 浅显的说就是我们可以通过ajax或者axios或者是fetch拿到数据

此时数据多为一个json数据或对象 我们可以通过在data中定义一个变量 然后通过this.\*\* == data(此处data为ajax的成功回调的参数)

实现数据对初始化数据的赋值

v-if

vue中支持使用指令来对dom进行操作 指令之说是继承于angularjs中的指令 angularjs中通过ng-XX vue中通过v-XX

v-if 用来动态控制DOM元素隐藏还是显示 对于v-if和v-show的区别 下面会说到

v-for

vue中我们会渲染各种类型的数据或对象 如果数据有很多条，他放在一个数组中 那么我们如果逐一赋值渲染并且再放到DOM中

那样就失去了简便的意义 在js中我们有for循环 forEach for in for of等等方法 在vue中 我们通过v-for 通过在标签中绑定v-for指令

<ol>

<li v-for='item in arr'></li>

</ol>

此处arr为我们之前data中定义的初始化数据变量 且我们已经通过ajax等拿到数据并且成功赋值 当然我们也可以在data中来自己定义数据

此处item为我们遍历出来的每一条数据 就像for循环

for(var i = 0;i<arr.length;i++){

console.log(arr[i])

}

此处的arr[i]就是item

v-on 简便的写法 @click

v-on指令用于在dom元素上添加事件监听器，借此调用vue实例中的各类方法 几乎所有的jq中的on可以绑定的事件 vue中都可以使用

methods

这是vue中的一个方法集 可以简单的理解为 我们通过v-on绑定事件后注定要调用某一个方法 但是在标签中写大量的js代码是不符合规范的，

并不符合我们的书写规范 所以我们只在事件监听器后面添加一个方法名 具体方法我们在methods中进行定义和实现

<a v-on：click='load'></a>

var VM = new Vue({

el:"#app",

data:{

message:"hello vue"

},

methods:{

load(){

console.log(1123)

}

}

})

我们此处点击a标签就会输出1123

v-model

vue中最值得谈到的就是数据的双向绑定 何为双向绑定 简单的理解就是vue中数据更新引发dom的变动，反之dom的改变触发数据的更新

vue数据双向绑定的原理是基于数据劫持结合发布者-订阅者模式的方式来实现的

1：实现一个监听器Observer，用来劫持用监听所有属性，如果有变动通知订阅者；

2：实现一个订阅者Watcher，可以收到属性的变化通知并执行相应的函数，从而更新视图；

3：实现一个解析器Compile，可以扫描和解析每一个节点的相关指令，并根据初始化模板数据以及初始化相应的订阅器。

简单就是通过v-model实现文本框值改变触发dom的更新

<div id='app'>

<p>{{mess}}</p>

<input type='text' v-mode='mess'

</div>

var app = new Vue({

el:"#app",

data:{

message:'hello vue'

}

})

关于vue初始暂时说到这里 对于vue组件的构建我暂时了解不是很透彻，先不误人子弟啦0.0