# 01---课程安排

## 目的：Django（数据要展示）

1.+js 行为(ESMAScript(基础语言), JSdom bom(浏览器的对象))

## 前端框架：

react 🡪 Facebook公司开发的

angular 🡪 谷歌公司

vue 🡪 中国人（尤雨溪）

# 02 ---let和const

前置的准备学习：

ES6简单的语法

## let和const

var a = [];

// for (var i=0;i<10;i++) {

// a[i] = function () {

// console.log(i);

// }

// }

for (let i=0;i<10;i++) {

a[i] = function () {

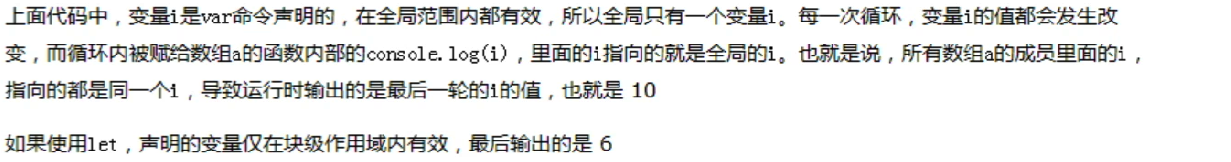
console.log(i);

}

}

a[2]() //var 声明的是全局变量 所以结果为10 无论a掺入的是几结果都是10

//let 声明的是 块儿级作用域只作用域for循环 所以输入的几 输出的就是几



## 模板字符串

使用：`` 反向单引号 和 ${变量名} 来替换要显示的变量

<script type="text/javascript">

var a =1;

var b=2;

// var str ="哈哈哈"+a+"嘿嘿嘿"+b;

// var str =`哈哈哈${a}嘿嘿嘿${b}` 模板语言 `${变量名称}`

var str =`\`哈哈哈${a}嘿嘿嘿${b}\`` //转义 输出格式为 `内容`

console.log(str);

</script>

## 箭头函数

1. function(){}====()=>{}
2. 箭头函数中 arguments 不能使用 原生的可以使用

3. 如果使用了箭头函数 this 指的是定义时使用的对象

指向了我们的window 全局

## 对象的单体模式

单体模式：function（）{}模式

内部调用的 this 为调用该方法的对象

<script type="text/javascript">

var person = {

name:"zhaichuang",

age:"18",

fav(){ //单体模式

console.log(this); //调用该方法的对象

console.log(this.age);

}

//等同于：

// fav:function () {

consle.log(this) //调用是 this指 全局变量window

// }

}

person.fav();

</script>

## es6的面向对象:

//ES6 构造函数

class Animal {

constructor(name,age) { //必须有这个属性 单体函数

this.name = name;

this.age = age;

// this.showName();

} //不要加逗号（，） 否则报错

showName(){

console.log(this.name);

}

}

var d = new Animal("张三");

d.showName()

## 模块儿化：



# 03 ---node 和npm

npm node的包管理工具

npm -v 查看版本

针对mac 用户

npm install npm@latest –g 升级npm版本

npm init -🡪 进入目录后运行该命令 初始化项目

npm run dev 🡪运行开发环境

npm run build –> 编译发布的包

生成 package.json 文件 ---🡪管理包

npm install jquery –save --save 必须加 默认最新版本

npm install jquery @version–save --save 必须加 默认最新版本

npm uninstall xxx –save 卸载包

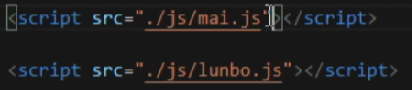
目录 node\_modules 下包含所有的包

上传到GitHub时 不能上传大于100m的项目 所以 node\_modules 目录无法上传上去 只有 package.json 文件 所以 下载项目后 运行 npm install 可以安装文件中的第三方插件

# 04 ---webpack bale和vue第一个程序

一个.js 文件就是一个模块儿

同步调用 按顺序加载



## webpack 打包工具 -🡪博客地址



babel 将es6代码编译成 es5

## 第一个VUE实例：

1. 引入vue.js
2. 创建一个vue实例

# 05 ---VUE 指令系统

{{插入表达式、三元运算符}}

v-if = ‘show’ 显示元素是否显示

v-else-if

v-else

v-show : 频繁切换

v-bind:title = “自定义的title”

v-bind:stc=””

# 06 ---VUE 指令系统2

v-for : (item,index ) in list

v-html = ‘str’ 渲染标签

v-model=”msg” -----数据的双向绑定 ----只适用在form表单控件里面

v-model.lazy 鼠标离开输入框后显示输入的内容

v-model.number 只能输入数字

v-model.trim 去除输入框两边的空格

{{msg}} 双向绑定

e.target.value

@mouseover 鼠标滑到到元素之上

@mouseleave 鼠标离开

@input input事件对象

@submit.prevent 阻止提交form

只要是与后台交互的技术都叫ajax技术

# 07----第三天

1.{{}} 模板语法，插值 简单的运算

2.指令系统

v-if

v-show 频繁使用

v-bind 绑定属性 简写： ：

v-on 绑定事件 简写： @

v-for 遍历 -🡪优先级高

v-html 解析html标签

v-model 只适用在表单控件中 双向数据绑定

双向数据绑定 = 单项数据绑定+UI事件监听

23 中设计模式

MVC MVT MVVM

Model View ViewModel

Ulr =🡺视图函数=🡺 Templeate

computed 计算属性，计算数据的属性 在data中保存的数据

默认只有 get方法 需要时 可以实现set方法

数据驱动视图

# ：：：：：：：Vue组件的使用：：：：：：

## 组件创建

在同一个页面创建一个组件 ，关键字 component

组件中 没有 el el 只用于根组件

组件创建:

Vue.component(“name”,{

//选项：

//data 必须是一个函数，必须返回一个函数

data:funciton(){

return {

//必须填写 哪怕是返回空

}

}

})

vue init webpack-simple myProject01

一个 .vue 文件 就是一个组件

webpack+node.js 将我们的项目更加方便化

一个组件有三个部分组成:

1. template
2. script

export default{

name:’App’,

data(){

return{

msg

}

}

}

1. style

4.

cd到当前目录

vue init webpack-simple 01

接下来根据提示操作

cd 01

npm install

npm run dev

在main.js 中写入import $ from “jquery”

## 父子传值

### 父组件--🡪子组件传值

步骤：父组件界面定义数据--🡪组件标签绑定自定义属性指向数据---🡪子组件页面中 使用 porpos关键字验证传来值类型-🡪使用传过来的值

1. 绑定自定义属性
   1. 在父组件中定义属性 citys：；[“上海”，“北京”]
   2. 子父组件页面引用的 子组件中绑定自定义组件 :mycitys=”citys”
   3. 在子组件页面中 使用关键字 props: 接收父组件传过来的值并且必须通过验证

props：{

mycitys: 类型（Array，String，Number，Boolean,Function,Object）

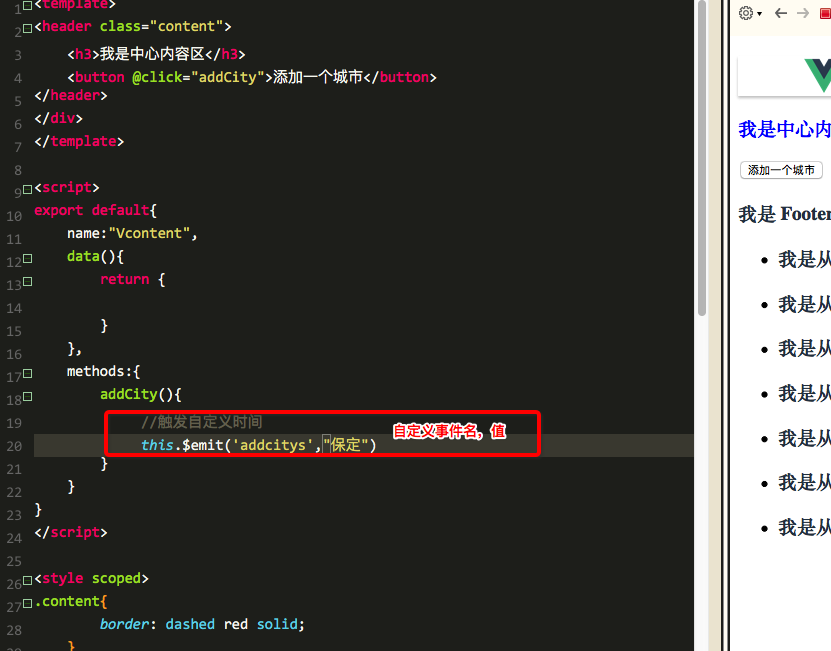
}

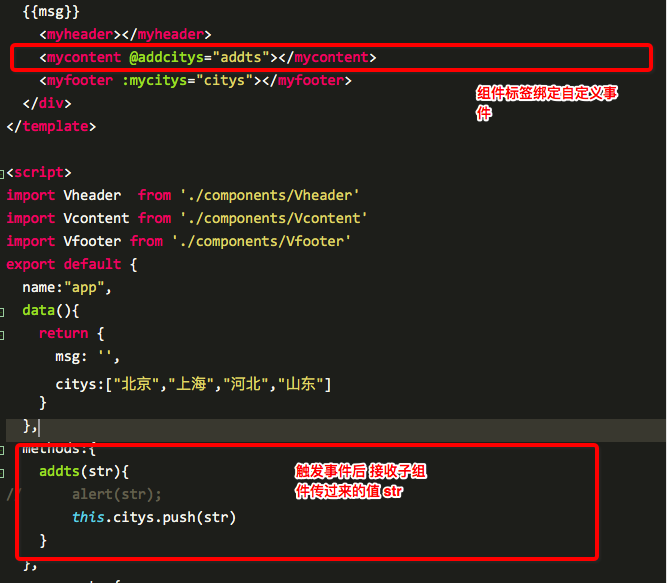


1. 直接在子组件中使用穿过来的值

### 子组件--🡪父组件传值

步骤： 子组件页面内定义时间 -🡪触发组件便签绑定的自定义事件（this.$emit(“组件标签绑定的自定义事件”)）--🡪修改父组件中的属性值





### 小练习：实现markdown效果

下载插件 marked

命令：npm install marked –save

## Vue-router 核心插件：：：：：：：

总：在vue项目中如果使用插件必须在main.js中配置

### 安装

npm install vue-router –save

### 使用

在 main.js中必须使用 Vue.use() 明确的安装了路由功能

improt VueRouter from “vue-router”

import home from ‘xxxxx/home’

Vue.use(VueRouter)

实例化VueRouter

const router = new VueRouter({

routes:[ //这里必须是单词 routes 一个字母都不能错 否则 地址栏能跳转 但是内容却不显示 ---大坑 深坑

{path:”/”,component:hone} 一个地址对应一个vue

]

})

挂载到Vuede实例化中

去掉url中的 # 号 使用 mode：history

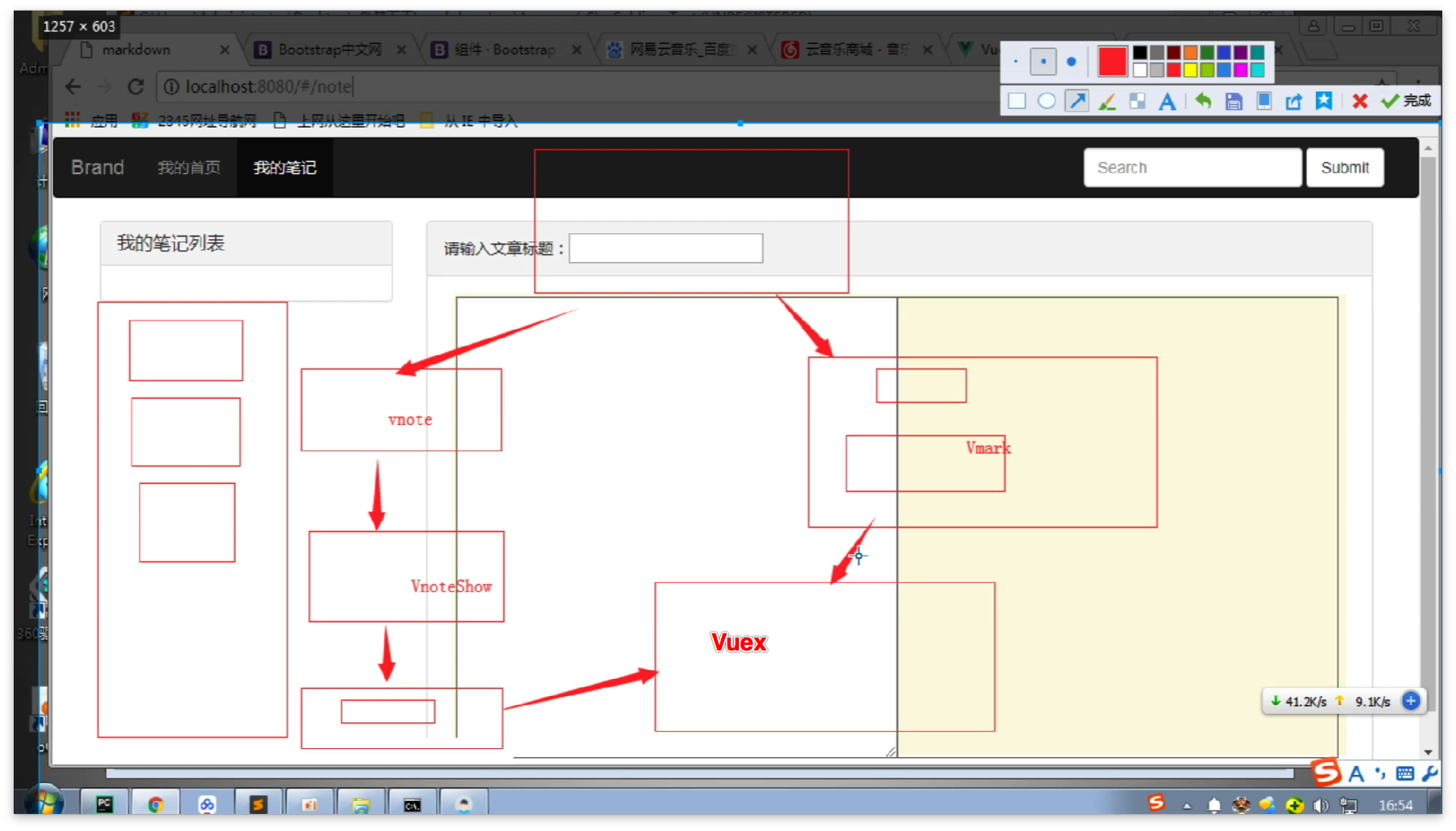
路由切换保持状态：刷新后仍然在当前页

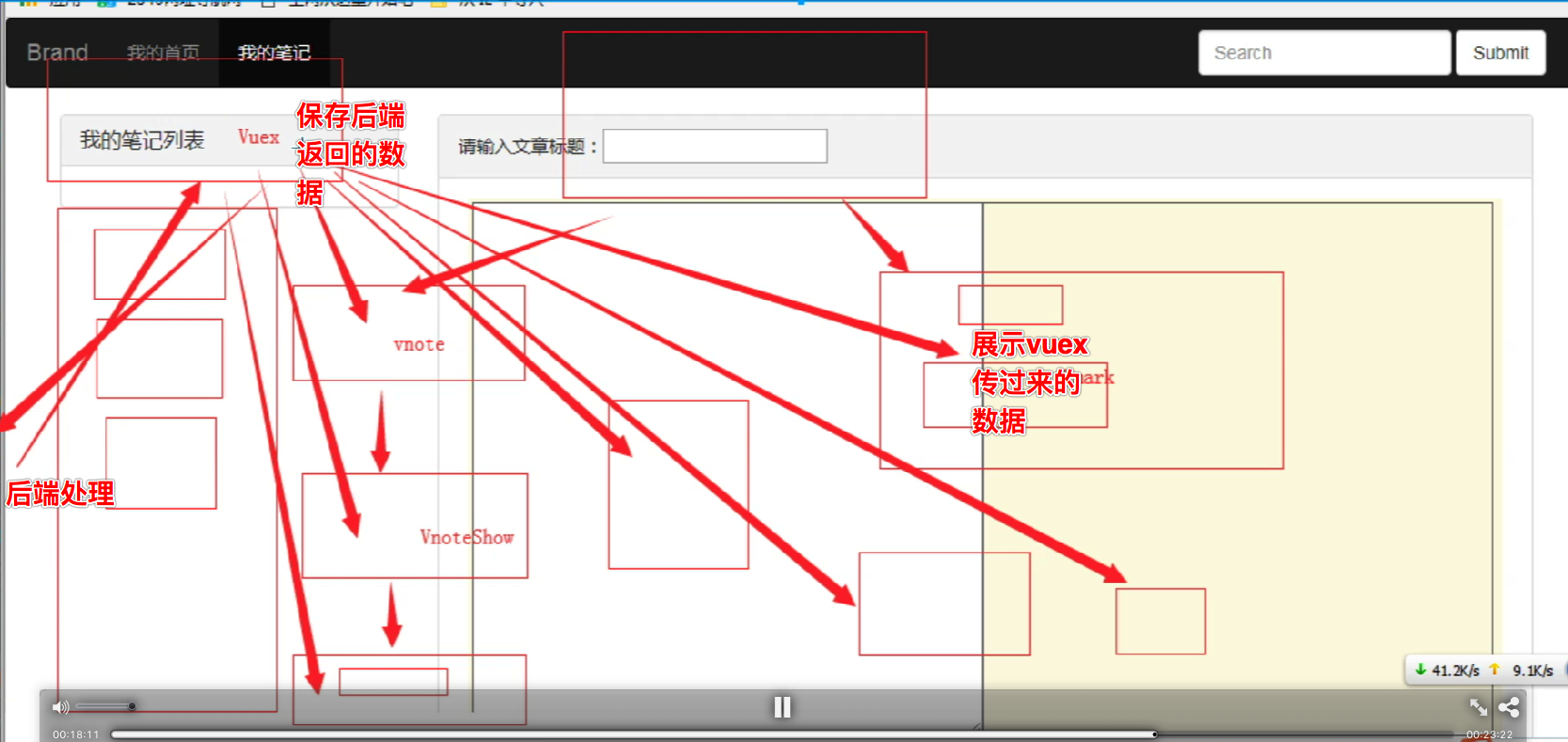
created:{

this.currenturl = this.$route.path

}

### VUEX





### 安装VUEX

npm install vuex --save

### 使用VUEX

1.导入VUEX : import Vuex from ‘vuex

2.声明使用： Vue.use(Vuex)

mutation 更改vuex中的 state状态的唯一方式



state getter mutation action module

组件 --🡪dispatch -----action ---commit ----mutation ---render---组件