[笔记概要 5](#_Toc517336601)

[1.位置 5](#_Toc517336602)

[2.笔记步骤： 6](#_Toc517336603)

[3.笔记结构 6](#_Toc517336604)

[4.第3版本：学习方法 6](#_Toc517336605)

[5.word文档快捷键： 7](#_Toc517336606)

[6.格式修改进程备注： 7](#_Toc517336607)

[一、Html5 7](#_Toc517336608)

[WS&sublime常用快捷键 7](#_Toc517336609)

[H5 day01 8](#_Toc517336610)

[1.课程简介： 8](#_Toc517336611)

[1.1工具：webstorm 9](#_Toc517336612)

[2:html5新特性10 (网页漂亮,功能强,动画效果) 9](#_Toc517336613)

[2. html5新特性--增强型的表单 9](#_Toc517336614)

[(1)新input type? 9](#_Toc517336615)

[(2)新表单元素 10](#_Toc517336616)

[(3)新表单元素的属性 10](#_Toc517336617)

[3.1: datalist (数据列表) list="list3" 10](#_Toc517336618)

[3.2: progress 进度条 11](#_Toc517336619)

[3.3: meter 刻度尺 11](#_Toc517336620)

[3.4: output 输出，同span，语义标签 12](#_Toc517336621)

[3.5: <input>的表单新属性 13](#_Toc517336622)

[4. 学子商城(后台) 14](#_Toc517336623)

[4.1：后台管理员登录 14](#_Toc517336624)

[4.2: 项目目录结构 [c:/xampp/htdocs/h5/] 14](#_Toc517336625)

[1.在数据库建表 15](#_Toc517336626)

[2.建登录php 16](#_Toc517336627)

[2.1错误集锦: 16](#_Toc517336628)

[H5 day02 16](#_Toc517336629)

[H5 day03 21](#_Toc517336630)

[H5 day04 23](#_Toc517336631)

[1.第三方绘图工具库 echarts 23](#_Toc517336632)

[2.svg绘图 24](#_Toc517336633)

[1. h5新特性---第三方绘图工具库 echarts(canvas) 25](#_Toc517336634)

[二、Bootstrap 27](#_Toc517336635)

[三、Nodejs 27](#_Toc517336636)

[nodejs day1 27](#_Toc517336637)

[1.阿里面试题:用户在浏览器中输入 www.taobao.com 直到看到网页\_\_之间发生了什么?☆ 27](#_Toc517336638)

[2. 静态网页和动态网页 27](#_Toc517336639)

[3.node.js 基本概念/模块 28](#_Toc517336640)

[4. 淘宝镜像（node）下载安装 28](#_Toc517336641)

[5.Node.JS 两种运行模式及操作 29](#_Toc517336642)

[(1)交互模式---一般用于临时测试 29](#_Toc517336643)

[(2)脚本模式--正式项目中使用方式 29](#_Toc517336644)

[操作提示 : △用ws操作node时设置设置方法： 29](#_Toc517336645)

[6. Node.JS 如何自学一门语言-- (node.js) 30](#_Toc517336646)

[(1)了解背景 30](#_Toc517336647)

[(2)搭建开发环境:编写Hello World 30](#_Toc517336648)

[(3)数据类型-----重点 30](#_Toc517336649)

[(4)变量和常量 31](#_Toc517336650)

[(5)运算符 31](#_Toc517336651)

[(6)逻辑结构 31](#_Toc517336652)

[(7)通用小程序 九九乘法表 100以质数 数组排序 32](#_Toc517336653)

[(8)函数和对象 32](#_Toc517336654)

[(9)常用的组件，第三方工具，框架 32](#_Toc517336655)

[(10)实际小项目 32](#_Toc517336656)

[7.Node.JS 中特有概念--模块 32](#_Toc517336657)

[1.用户自定义模块与运用 33](#_Toc517336658)

[1.文件模块 （自加代码及文件模块间导出引用运用方法） 33](#_Toc517336659)

[导出的异同：exports和module.exports 前者导出对象失败 34](#_Toc517336660)

[综合练习：模块在url中的拼凑运用 35](#_Toc517336661)

[四、VUEJS 36](#_Toc517336662)

[今日目标： day 01 36](#_Toc517336663)

[1.三大框架前端 36](#_Toc517336664)

[1.1. 理解：框架、Model、view、绑定（了解）... 36](#_Toc517336665)

[2.概述Vue.js 37](#_Toc517336666)

[1.官网： 37](#_Toc517336667)

[2.手册： 37](#_Toc517336668)

[3. Vue.js的特点 37](#_Toc517336669)

[3.语法： 38](#_Toc517336670)

[1.Vue.js模板语法(Mustache大胡子):把Model绑定到View 38](#_Toc517336671)

[1.<ANY>{{ 模型数据 }}</ANY> 38](#_Toc517336672)

[2.Vue实例可能出现成员（完整写法）： 38](#_Toc517336673)

[2.1给webstorm添加vue基础模板 39](#_Toc517336674)

[2.模型数据可以是如下形式的数据： 39](#_Toc517336675)

[3.实现数据绑定的写法： 39](#_Toc517336676)

[(1)模板语法： <p v-cloak>{{ msg }}</p> 39](#_Toc517336677)

[(2)Vue.js指令： <p v-html="msg"></p> 39](#_Toc517336678)

[绑定练习1： 39](#_Toc517336679)

[绑定页面练习2——new Vue对象的结构： 40](#_Toc517336680)

[练习3绑定数据形式——三目： 41](#_Toc517336681)

[4..防止变量闪烁写法（隐身斗篷） 41](#_Toc517336682)

[(1)普通防抖动写法：v-cloak {{msg}} +[v-cloak]{display:none} 41](#_Toc517336683)

[(2)防止抖动写法：标签内写,v-html="msg" 41](#_Toc517336684)

[4.指令Vue.js中的 (Directive) 13个 41](#_Toc517336685)

[4 v-for 遍历形式 引用元素和下标： 42](#_Toc517336686)

[5 v-for 多级 对象[0]形式填充： 42](#_Toc517336687)

[6 v-if vs v-show形式显示、else if形式写法： 43](#_Toc517336688)

[7 v-for + v-if、对象.下标 填充： 43](#_Toc517336689)

[8 v-bind属性绑定变量值、剪切省略显示写法： 44](#_Toc517336690)

[9 v-bind属性绑定变量缩写、属性为超链接——超链接内有:字符串+变量时的写法： 44](#_Toc517336691)

[8. v-on 监听v-on 为当前元素绑定事件监听函数 45](#_Toc517336692)

[1.写法： 45](#_Toc517336693)

[练习1：事件绑定加方法（改属性值） v-on / @ 45](#_Toc517336694)

[练习2：事件绑定加方法（改数据变量） v-on / @ 46](#_Toc517336695)

[2.事件修饰符：https://cn.vuejs.org/v2/api/#v-on 46](#_Toc517336696)

[9. v-bind 47](#_Toc517336697)

[10. v-model 47](#_Toc517336698)

[练习1：data绑定多处——提交注册信息+同意条款绑定： 47](#_Toc517336699)

[练习2：data绑定——前后对应变动 1 47](#_Toc517336700)

[练习2.2：data绑定——前后对应变动（select——val） 48](#_Toc517336701)

[练习2.3：data绑定——前后对应图片变动（select——val） 49](#_Toc517336702)

[vuejs day02 51](#_Toc517336703)

[复习day01： 51](#_Toc517336704)

[5.绑定属性Class和Style： V-model—data—绑定属性 52](#_Toc517336705)

[练习3：style绑定3种 52](#_Toc517336706)

[练习4：class的绑定：3种方法 53](#_Toc517336707)

[6.指令自定义 Vue.js中的——了解（有过滤器了） 54](#_Toc517336708)

[1.指令写法： 55](#_Toc517336709)

[2.用户也可以自定义扩展指令： 55](#_Toc517336710)

[练习：自定义自动获得焦点 vuejs day2 12 55](#_Toc517336711)

[7.过滤器自定义Vue.js中的 —— 掌握 56](#_Toc517336712)

[1.可以在两种情形下使用过滤器： 56](#_Toc517336713)

[练习1过滤男女01 day2 14 57](#_Toc517336714)

[练习2 年月日过滤成yyyy-mm-dd hh:mm:ss day3 2（类似） 57](#_Toc517336715)

[2. 给过滤器 加参数 58](#_Toc517336716)

[练习：计算员工日期，返回年月日格式 day03 2 58](#_Toc517336717)

[vue.js day03 59](#_Toc517336718)

[8. axios get then异步请求Vue.js中发起 59](#_Toc517336719)

[练习异步示例代码：4步 day3 3 60](#_Toc517336720)

[9. Vue成员实例可能出现 6 60](#_Toc517336721)

[10. Vue.js实例的生命周期 —— 重点理论 61](#_Toc517336722)

[11.组件自定义Vue.js中的—— 重点 63](#_Toc517336723)

[(1)全局组件：可以在页面中任何一部分使用的组件 63](#_Toc517336724)

[练习：书籍组件 63](#_Toc517336725)

[(2)局部组件：只能用于特定的父组件内的组件 64](#_Toc517336726)

[练习：员工组件——+局部组件 day3 10 (可遍历) 64](#_Toc517336727)

[课后练习：待办事项列表——3组 Todo List day4 2 66](#_Toc517336728)

[vue.js day04 66](#_Toc517336729)

[12.HTML标签、属性、事件名称的大小写： 66](#_Toc517336730)

[1.HTML标签名： 66](#_Toc517336731)

[2.HTML标签属性名： 66](#_Toc517336732)

[3.HTML标签事件名： 67](#_Toc517336733)

[4.ES6中对象的简写方法： 67](#_Toc517336734)

[13.组件间通信 —— 难点 67](#_Toc517336735)

[1.Vue.js中组件直接传递数据 —— 重点&难点&面试题 68](#_Toc517336736)

[(1)父组件给子组件传递数据](#_Toc517336737)**[day04 4-5-6](#_Toc517336737)** [68](#_Toc517336737)

[(2)子组件给父组件传递数据](#_Toc517336738)**[day04 7-8](#_Toc517336738)** [68](#_Toc517336738)

[(3)兄弟组件间传递数据 —— 实际应用场景比较少](#_Toc517336739)**[day04 13](#_Toc517336739)** [69](#_Toc517336739)

[练习：在父组件内有模型数据：newsList: ['新闻标题1', ....] 69](#_Toc517336740)

[练习：使用父子组件实现商品列表：day4 5-6 69](#_Toc517336741)

[练习子传父：竞选总统day4 8 70](#_Toc517336742)

[练习子传父：购物车计算器day04 11-12 70](#_Toc517336743)

[2.小知识：Vue.js中的计算属性 day4 8 70](#_Toc517336744)

[▲总结：组件间通信的三种方法（面试） 70](#_Toc517336745)

[(1) Props Down, Event Up day5 1 70](#_Toc517336746)

[(2) $refs 和 $parent day5 2 71](#_Toc517336747)

[(3) 通信总线(Bus) day5 3-4 71](#_Toc517336748)

[综合练习：TODO List day5 9 72](#_Toc517336749)

[vue.js day05 72](#_Toc517336750)

[14. 模板的声明方法组件中的：用第三种 72](#_Toc517336751)

[(1)js中直接建模板 72](#_Toc517336752)

[(2) script引用模板 72](#_Toc517336753)

[(3)标签template模板 73](#_Toc517336754)

[15. 动画效果Vue.js中的 73](#_Toc517336755)

[方式1： CSS过渡(transition) (练习) 73](#_Toc517336756)

[练习：“不接受许可”则按钮向左滑动离开并渐隐；“接受许可”则按钮从右面滑动进入并渐显。 74](#_Toc517336757)

[午间自学：Animate.css —— 第三方动画效果库 74](#_Toc517336758)

[方式2：配合Animate.css动画库实现动画 75](#_Toc517336759)

[父子通讯综合练习：Todo List 经典分析 76](#_Toc517336760)

[16.Vue.js阶段项目 —— “学子商城后台管理系统”框架重构 77](#_Toc517336761)

[1.SPA应用 77](#_Toc517336762)

[1.SPA应用的实现原理简介： 78](#_Toc517336763)

[(1)定义“路由词典”： 78](#_Toc517336764)

[(2)定义唯一“完整”的HTML页面：index.html 78](#_Toc517336765)

[(3)客户端发起请求： 78](#_Toc517336766)

[1：分析url 78](#_Toc517336767)

[2.“页面跳转”其实只是在修改#/后的内容：（插件Vue-Router） 78](#_Toc517336768)

[3.Vue-Router插件的使用步骤：5步 78](#_Toc517336769)

[vue.js day06 000000 79](#_Toc517336770)

[4.VueRouter中实现“页面”切换 79](#_Toc517336771)

[(1)传统的超链接方式 79](#_Toc517336772)

[(2)VueRouter提供的超链接方式 79](#_Toc517336773)

[(3)使用JS跳转方式 80](#_Toc517336774)

[5.VueRouter中编写带参数的路由 80](#_Toc517336775)

[(1)声明路由词典时需要指定路由参数 80](#_Toc517336776)

[(2)模板页面跳转时必须为每个路由参数提供具体的值 80](#_Toc517336777)

[(3)组件中获取路由参数 80](#_Toc517336778)

[6.VueRouter中实现“嵌套式路由” 80](#_Toc517336779)

[午间任务：练习 80](#_Toc517336780)

[(1)部署“学子商城后台管理系统V1” 80](#_Toc517336781)

[(2)自学官方教程“单文件组件”相关内容 81](#_Toc517336782)

[7.Vue.js的两种使用方法 81](#_Toc517336783)

[(1)直接使用<script>引入 81](#_Toc517336784)

[(2)使用Vue-CLI工具 Command Line Interface 81](#_Toc517336785)

# 笔记概要

## 1.位置

|  |  |
| --- | --- |
| **块** | **位置** |
| 知识点、笔记、代码 | 笔记 截图代码放入笔记 |
| 习题 | 习题（截图+提示） |
| 笔记 (有必要才放笔记，必须进--习题) |
| 习题代码 | 习题（截图） |
| 代码文件 | teacher\_note |
| 小面试题 | 面试题 |
| 操作、常识 | 操作-专业知识 |

|  |  |
| --- | --- |
| **块** | **样式** |
| 练习题目（详情） | 标题2 |
| **\***习题 | 练习/案例 |
| 操作、常识 | 操作步骤： |

## 2.笔记步骤：

|  |  |
| --- | --- |
| 遍历笔记第一次：理清 | 复制，分级，复制代码解说，标注（+简记）， |
| 二次：实例 | 标需求，习题截图收录，面试题 |
| 三次：其他实例/作用 | 需求延伸联想 |

## 3.笔记结构

标题——提示语

思考线——总结

概念阐述——图解  
练习——步骤线

## 4.第3版本：学习方法

|  |
| --- |
| 1.看视频 截图重点（理解，代码，步骤）并标记回想提示放进笔记 2.边看边打代码 |
| 2.别被概念什么的禁锢脑袋，造成运转困难，要放松轻松运行。 |
| 3.学习一段代码：总结（写出）目的和步骤 |
| 4.先脑后兵（先弄全思路后做） |

4.先脑规划事件事件等（空找） 再搞详细逻辑（实际操作）

例：做逻辑题都是先审题——想着手地方——最后才动笔算（演示逻辑）——判断结果

例：先想要求（放大镜）——先触发中图事件——中图出阴影背景——阴影背景位置——再定大图位置（背景图位置偏移（left，top）

例：这个晚上学习这个知识点：代码整理好——笔记整理好——

## 5.word文档快捷键：

文件→选项→自定义功能区

选择键盘快捷方式 自定义

|  |  |
| --- | --- |
| 跳到之前目录 | ctrl+d |
| 跳到目前编辑的地方 | ctrl+g |

## 6.格式修改进程备注：

|  |  |  |
| --- | --- | --- |
| 例子格式修正 | HTML DOM API |  |
| 面试：其他好认的颜色 |  |  |
| 完整老师笔记代码 | 练习1：计算器点击事件 |  |
|  |  |  |
|  |  |  |

# 一、Html5

### WS&sublime常用快捷键

|  |  |
| --- | --- |
| ws | sublime |
|  | /\* |
|  | 窗口类 |
| ctrl+r | ctrl + shift + t 恢复窗口 |
| 1 | Ctrl+PageDown 向左切换当前窗口的标签页。 |
| 上一次编辑位置 | ctrl + q 打开 |
| 1 | ctrl + w 关闭 |
| 0 | ctrl +tab 流水切窗 |
|  |  |
|  | 选择类 |
| 0 | ctrl+D 选中文本 （复按：多光标） |
| 1 / ctrl+enter | shift +enter 后插入 |
|  | shift + ctrl + enter 前插入 |
| 1 | ctrl + ← 单位性地移动光标 |
| 1 | ctrl + → |
| 个体右边 | ctrl+m 在括号前后切换 |
| 1 | end / ctrl+i 光标行尾 |
|  |  |
|  | 编辑类 |
| ctrl+e | ctrl+e 删除整行 |
| ctrl+alt+L | ctrl + shift + v 格式粘贴 |
| 1 | home 光标行首 |
| 1 | ctrl+/ 注释 |
| 1 | Ctrl+Z 前进一步 |
| 1 | Ctrl+Y 后退一步 |
| 1 | ctrl+Shift+D 复制光标所在整行，插入到下一行。 |
| 查变量/0 | ctrl + shift + ↑/↓ 上移 |
| 0 | ctrl+K+K 从光标处开始删除代码至行尾。 |
| 0 | shift + ctrl + a 编辑该标签文本内容 |
| ctrl+shift+U | Ctrl+K+U 转换大写 |
|  | Ctrl+K+L 转换小写 |
|  |  |
| 1 | ctrl+Delete 右删除 |
|  |  |
|  | 搜索类 |
| 0 | ctrl + g 定行 |
|  | F6 单词检测拼写 |
|  | Ctrl+F2 设置书签 |
|  |  |
|  | 显示类 |
|  | \*/ |
| webstorm 快捷键 | |
| 全局查找 | ctrl+h |
| 高亮同词 | ctrl+f |
| 自动整理代码 | ctr+shift+f |
| 找文件 | 双击 shift |
| 选中当前词段 | 双击 |
| 备用 | ctrl+d |
| 备用 | ctrl+g |

H5 day01

## 1.课程简介：

(1)html5新特性--(8day) canvas 绘图技术[游戏/特效]

(2)Bootstrap css/js 框架6 day[ 响应式布局框架]

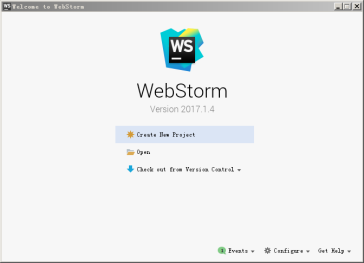
(3)Node.JS 运行于服务器开发平台(php)

**建议:复习** js基本语法 / js面向对象[原型/原型链/闭包] / dom

5小时[1小时总结 思维导图 4小时]

### 1.1工具：webstorm

1.放进hotdocs文件内运行：保证网页地址：127.0.0.1开头才能保证项目是经过后台在回来



## 2:html5新特性10 (网页漂亮,功能强,动画效果)

(1)html5新特性--新的语义标签 footer header...

(2)html5新特性--增强型表单(2.0)

(3)html5新特性--音频和视频

(4)html5新特性--Canvas绘图(重点)-大鱼小鱼 (2day)

(5)html5新特性--SVG绘图(地图)

(6) html5新特性--地理定位

(7) html5新特性--拖放API

(8) html5新特性--Web Worker

(9) html5新特性--Web Storage 前端保存一些数据如cookie

(10) html5新特性--Web Socket 解决http的一些问题

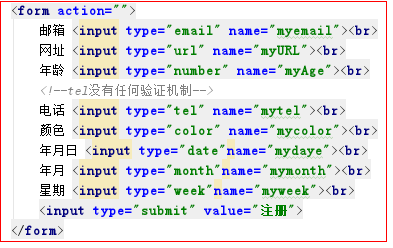
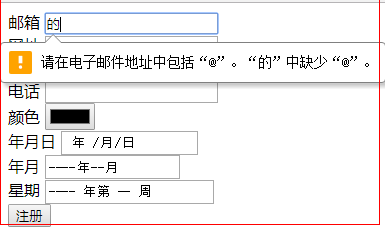
3-4:让网站更绚丽丰富

## 2. html5新特性--增强型的表单

### (1)新input type?

h4:text/password/button/submit/radio/checkbox/image/file/hidden/reset..

h5:email/url/number/tel/color/range/date/month/week



### (2)新表单元素

h4:input;button;select/option;textarea

h5:datalist/progress/meter/output

### (3)新表单元素的属性

#### 3.1: datalist (数据列表) list="list3"

<datalist id="list3"> datalist本身不可见

<option>xx</option>

<option>yy</option>

</datalist>

<input type="text" list="list3" />

datalist为input **提供建议**列表用户从中选取，也**可以手工输入**



#### 3.2: progress 进度条

显示一个进度条，有两种形式

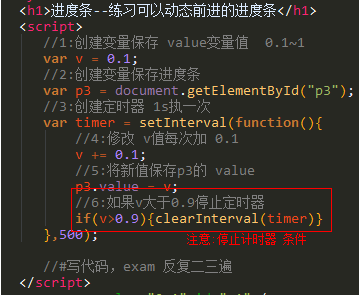
<progress></progress> 左右晃动进度条

<progress value="0.1" /> 具有指定进度值的进度条(0~1)

练习:

使用定时器+进度条实现一个可以动态前进的进度条

到 100% 时弹出“下载完成”



提示：#进度条外观有限,如果需要更好外观（复杂的进度条用cavnvas）

#### 3.3: meter 刻度尺

meter:用于标识一个值所处范围:

不可接受(红色) 可接受(黄色) 非常优秀(绿)

写法：

<meter min="最小值" max="最大值" low="下限" high="上限"

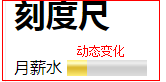
optimum="最佳值" value="当前值" >

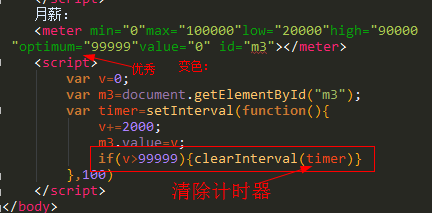
</meter>

例子：薪水级别

<meter min="0" max="100000" low="20000" high="90000"

optimum="99999" value="0" />





清除计时器

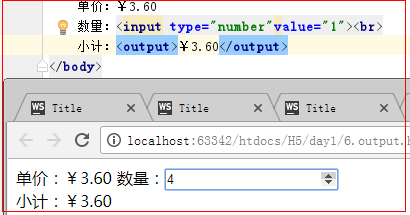
#### 3.4: output 输出，同span，语义标签

output:输出,语义标签，没有任何外观样式，样式等同span

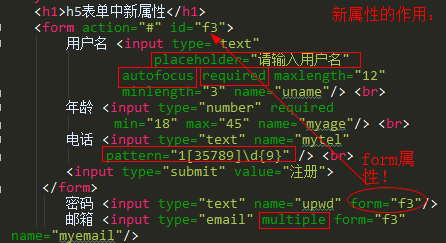
商品单价:￥3.50

商品数量:<input type="number" value="2"/>

小计:<output>￥3.50</output>



#### 3.5: <input>的表单新属性



h4:type;name;id;value;style;readonly;disabled;

h5:

(1)placehodler:占位符

<input type="text" placehodler="请输入用户名称"/>

(2)autofocus:自动获取焦点

<input type="text" autofocus/>

(3)multiple:可输多个值(使用逗号分隔) email

(4)form:用于把输入域放置form 外部（如下：↓）

==与验证相关属性

(5)rquired:必填项

(6)minlength:指定字符最小长度

(7)maxlength:指定字符最大长度

(8)min:数值最小值

(9)max:数值最大值

(10)pattern:正则表达式



### 4. 学子商城(后台)

(1)后台管理员登录

步骤：1. 项目结构admin

2. 数据库建表

3. 建php访问数据库登录

4. index.html+index.js访问

(2)产品分页显示

(3)删除指定产品

(4)更新指定产品价格

(5)显示指定产品详细信息

(6)搜索指定范围产品

(7)添加产品

--------------------------------------------------------------------------------------------------

（1）第一个功能:后台管理员登录（步骤分析）

学子商城:二类用户

(1)注册用户-->购买商品->浏览商品

(2)管理者->上架(下架,删除,更新)

操作流程

(1)创建一张表 管理员数据表 xz\_admin

(2)data/01\_admin\_login.php

(3)index.html

(4)index.js

常见错误

(1)php访问时直接下载

原因:header()内容拼写错误

(2) mysqli\_connect(): (HY000/2002

原因:数据库没有成功启动,数据库严重错误

(3)mysql 坑 SET NAMES UTF8 正确

SET NAMES UTF-8 错误

(4) Table 'xz.xz\_admin1' doesn't exist

原因:xz\_admin1 表不存在,通常表名拼写不正确

(5) Unknown column 'uid1' in 'field list

原因:uid1列不存在,通常列名拼写不正确

---------------------------------------------------------------------------------------------------

#### 4.1：后台管理员登录

#### 4.2: 项目目录结构 [c:/xampp/htdocs/h5/]

admin-------------------保存所有项目所需资源

db.sql-------------保存项目sql指令

data 保存所有php程序

img 图片

fonts 字体

css css文件

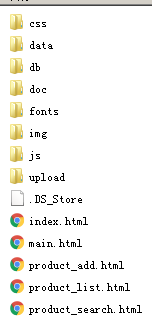
js js 文件

upload 上传文件目录

index.html 后台管理员登录(网页)

main.html 主页面

product\_list.html 产品主页面

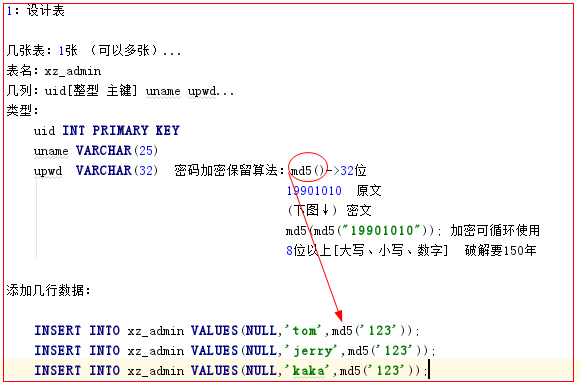


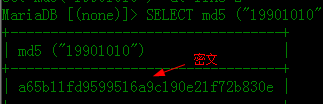
强烈建议：所有目录/文件/路径/ 不要有：中文、空格、特殊字符

规格：用全小写字母、下划线

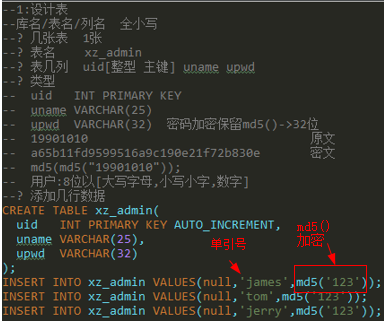
##### 1.在数据库建表

**加密实例**：

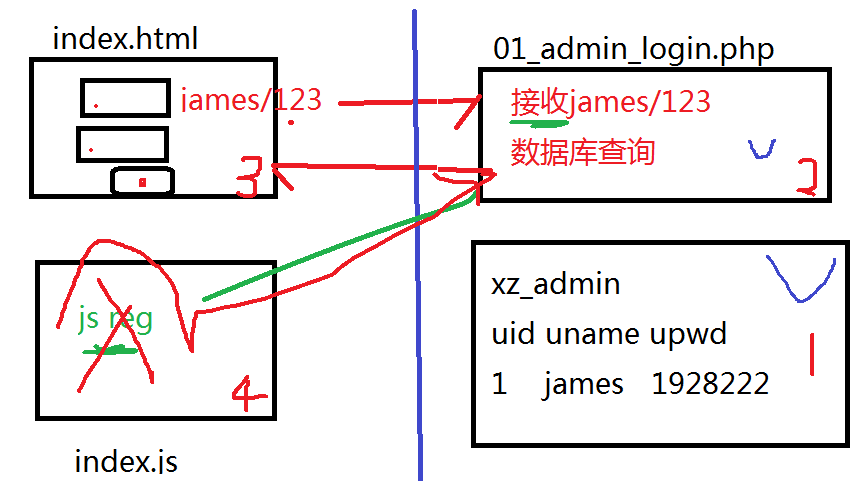




实例运用：



##### 2.建登录php



###### 2.1错误集锦:

(1) Parse error: syntax error line 17 语法错误

(2)下载错误 header内容写错

(3) mysql\_connect(): Access denied line 9访问被拒绝, 数据库

用户名或密码错

(4) mysql\_connect(): MySQL server has gone away in

mysql\_connect()-->升级-->mysqli\_connect();

2.2 验证码：解决攻击手段

H5 day02

今天学习的内容

1:复习昨天内容

2:作业:json格式

json - > AJAX

json:数据格式,通常是以字符串形式表示

对象 {"name":"james","age":19}

数组 [{"name":"tom"},{"name":"jerry"},{"name","james"}]

所有key 必须添加双引号

所有value 如果字符串类型加上双引号

key和value之间 : 分隔

对象与对象之间 ,

常用数据格式:xml json

<books>

<book>

<name>tom</name>

</book>

<book>

<name>jerry</name>

</book>

</books>

验证码:解决网络攻击手段(机器人攻击)

1:动态创建数字图片/字母数字角度/算术/

一组图片/拖动滑块

2:短信(金融)

如何实现功能:

1:创建php动态生成一张图片

2:用户输入图片中字母或数字

js/index.js

3:01\_admin\_login.php

先把用户输入字母和数字 a001

再把图片中数字获取 $\_SESSION["code"]

常见错误

1: syntax error,

<b>C:\xampp\htdocs\h5\admin\data\01\_admin\_login.php</b> on line <b>10</b><br />

原因:程序第10 行(附近)有语法错误...

2: {"code":1,"msg":"登录成功"}

应该对象 header("Conten....");

3:今天的目标

3.1:h5新特性--视频,音频

视频:网站卖点[腾讯/爱奇艺/优酷/tmooc/抖音/直播..]

3.2:Flash被H5 取代体现的哪些方面?

Flash 绘图 ==>Canvas/SVG

Flash 动画 ==>定时器+Canvas

Flash 视频和视频播放==>video/audio

Flash 客户端存储 ==>WebStorage

3.3:h5新特性--视频播放

h5提供一个新的标签用于播放视频

mp4/mov/webm/ogg

<video src="x.mp4">您的浏览器版本太低，请升级<video>

<video>

<source src="x.mp4" />

<source src="x.mov" />

<source src="x.webm" />

您的浏览器版本太低，请升级

</video>

它本身是一个300\*150的inline-block元素

video标签常用属性和成员

autoplay:false 是否自动播放

controls:false 是否显示控件

muted:true 是否静音

poster:"" 在播放第一帧之前显示海报

preload:"" 视频预加载策略

auto: 预加载视频元数据及缓冲一定时长视频

metadata: 只预加载元数据(尺寸,时长,第一帧内容)

none: 不预加载任何数据

###js对象属性###

currentTime: 当前播放时长

paused:true 当前视频是否处理暂停状态

volume:1 当前音量 0~1

playbackRate:1 回放速率 大于1快放,小于1慢播

成员方法:

play(); 播放视频

pause(); 暂停视频

成员事件:

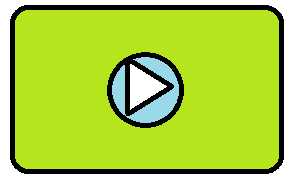
onplay 播放视频

onpause 暂停播放

练习1:不使用video自带controls，自定义播放/暂停按钮

移标移出视频区域，隐藏按钮,鼠标移入视频区域显

示按钮



练习2:只要视频暂停显示一副广告,只要播放，就隐藏广告

周一:弹幕...

3.4:h5新特性--音频播放

h5提供了一个新的标签用于播放音频

<audio src="bg.mp3"></audio>

<audio>

<source src="bg.wav">

<source src="bg.ogg">

<source src="bg.mp3">

</audio>

它默认是一个300\*30的inline-block,但若没有controls属性,不可见.

成员属性:事件/方法/属性

autoplay:false 是否自动播放

controls:false 是否显示控件

muted:true 是否静音

preload:"" 视频预加载策略

auto: 预加载视频元数据及缓冲一定时长视频

metadata: 只预加载元数据(尺寸,时长,第一帧内容)

none: 不预加载任何数据

###js对象属性###

currentTime: 当前播放时长

paused:true 当前视频是否处理暂停状态

volume:1 当前音量 0~1

playbackRate:1 回放速率 大于1快放,小于1慢播

成员方法:

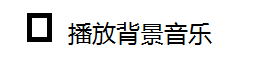
play(); 播放视频

pause(); 暂停视频

成员事件:

onplay 播放视频

onpause 暂停播放



练习:使用复选框控制网页背景音乐

cb.onchange = function(){

this.checked

}

3.5 h5新特性--canvas绘图(重点)

网页中走势图,统计图,网页游戏,特效,地图应用都要使用绘图技术。

(1)SVG绘图:2D矢量图绘图技术，2000出现，后纳入h5标

(2)Canvas绘图:2D位图绘图技术,h5 提出绘图技术

(3)WebGL绘图:3D绘图技术，尚未纳入标准

|  |  |
| --- | --- |
| Canvas绘图难点所在:  (1)坐标系  (2)单词比较多 |  |

3.6 h5新特性--canvas绘图技术

canvas:画布，是h5提供绘图技术

<canvas width="500" height="400">

您的浏览器版本太低，请升级!!!

</canvas>

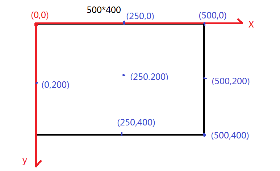
canvas标签在浏览器默认300\*150的inline-block

注意:画布宽和高只能html/js属性来赋值，

不能用css样式赋值

每个画布上有且只有一个"画笔"对象,

var ctx = canvas.getContext("2d");



(1)使用canvas绘制矩形

矩形的定位在自己左上角

ctx.lineWidth = 1; 描边宽度

ctx.strokeStyle = "#000"; 描边样式

ctx.fillStyle = "#000"; 填充样式

ctx.fillRect(x,y,w,h); 填充一个矩形(实心矩形)

ctx.strokeRect(x,y,w,h); 描边一个矩形

ctx.clearRect(x,y,w,h); 清除一个矩形范围内所有图形

练习1:在画布的左上角，右上角，左下角，右下角，

居中位置 绘制100\*80描边矩形(颜色不同)

练习2:在画布上描边一个可以左右移动100\*80描边矩形

提示:使用定时器，先清除画布上己有内容,

再重新绘制一个(x不停修改)

上下移动/右角45度..

(2)使用canvas绘制文本

ctx.textBaseline = "alphabetic"; 文本基线

ctx.font = "12px sans-serif"; 文本大小和字体

ctx.fillText(str,x,y); 填充一段文本

ctx.strokeText(str,x,y); 描边一段文本

ctx.measureText(str); 测量文本宽度{width:w}

作业:

1:创建web应用程序

salesdata.php

当客户发送请求给php时，返回JSON数据

[

{"label":"部门一","value":280},

{"label":"部门二","value":210},

{"label":"部门三","value":180},

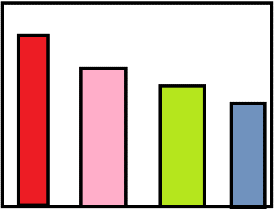
{"label":"部门四","value":170}

]

2:创建salesdata.html

发送ajax请求salesdata.php

依据返回数据，绘制统计图



H5 day03

今天学习的内容

1:复习昨天重点

2:作业

3:今天的目标

3.1:canvas绘图--(重点掌握:渐变对象.路径.图片.变形)

3.2:canvas绘图--渐变对象

线性渐变: linearGradient

径向渐变:

var g = ctx.createLinearGradient(x1,y1,x2,y2);

g.addStopColor(offset,color);

..

ctx.fillStyle = g;

ctx.strokeStyle = g;

3.3:canvas绘图--路径(最复杂)

Path:由多个坐标点组成任意形状,路径不见,可用于

"描边","填充","裁剪"...

ctx.beginPath(); 开始一条新路径

ctx.closePath(); 闭合当前路径

ctx.moveTo(x,y); 移动指定点

ctx.lineTo(x,y); 从当前点到指定点画直线

ctx.arc(cx,cy,r,start,end); 绘制圆拱形

cx cy 圆心

r 半径

start 开始角度 (弧度0~2\*PI)

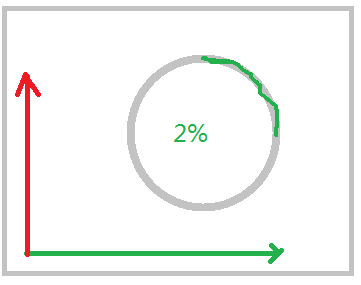
end 结束角度 角度\*Math.PI/180==弧度

ctx.stroke(); 描边

ctx.fill(); 填充

练习:使用描边绘制一个坐标轴

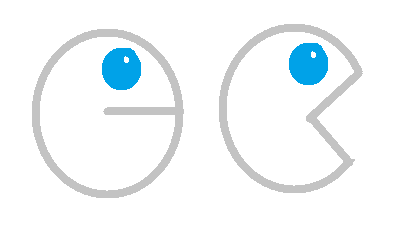
练习:使用圆拱形绘制可以前进的圆环进度条



练习:创建一个函数openMouth()/closeMouth();

创建定义时器，每隔1s交替调用

openMouth()/closeMouth();



3.4: canvas绘图--图片

canvas属于客户端技术，图片在服务器中，所以浏览器必须先下载要绘制的图片，且等待图片异步加载完成.

var p3 = new Image(); 创建图片对象

p3.src = "x.jpg"; 下载指定图片

p3.onload = function(){ 图片下载完成触发事件

console.log(p3.width);

ctx.drawImage(p3,x,y); 原始大小绘图

ctx.drawImage(p3,x,y,w,h); 拉伸绘图

}

练习:在画布四个角各画一架飞机

练习:在画布左上角画一个左右移动飞机

|  |
| --- |
| canvas绘图核心知识点--重占  绘制矩形:  ctx.fillRect();ctx.strokeRect();ctx.clearRect();  绘制文本  ctx.fillText();ctx.strokeText();ctx.measureText().width  绘制路径  ctx.beginPath();ctx.closePath();  ctx.moveTo();ctx.lineTo();  ctx.arc();ctx.stroke();ctx.fill()  绘制图像  ctx.drawImage(); |

3.4: canvas绘图--变形操作

canvas绘图可以针对于某一个图像/图形，的绘制过程

进行变形 rotate,translate

ctx.rotate(弧度); 旋转画笔，轴点画布的原点.

ctx.translate(x,y); 平移原点

ctx.save(); 保存画笔状态(存盘)

ctx.restore(); 恢复画笔状态到上一次保存(读取存盘)

练习:在画布左上角画一个绕自己为中心旋转飞机1

练习:在画布右上角画一个绕自己为中心旋转飞机2

飞机2旋转速度是飞机1 2倍svg绘图

练习:使用canvas绘制随机改变验证码图片!



var str = "ABC..abcdefghim...1234567890";

var char = str[字符串长度随机数] 17:45~17:57

要求:

画布背景随机颜色[浅色],ctx.fillRect();

文字内容随机，大小随机，颜色随机(深色),旋转角度随机

5条随机干扰线(深色), 贝赛尔曲线

100个干扰点(半径为1圆)

作业:弹幕->常见 视频+canvas综合功能.

作业:学子商品-->商品列表分页

H5 day04

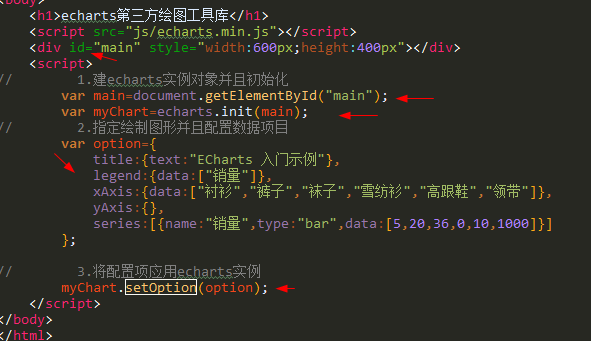
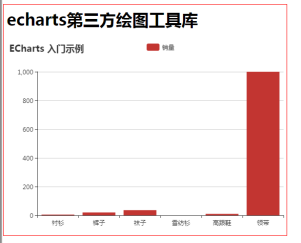
## 1.第三方绘图工具库 echarts

百度echarts ； d3 ; two; js ;....

网址：echarts.baidu.com （可在网址中的文档中编辑...再下载）

步骤：下载：echart.min.js放进js文件夹先，在按以下实例格式插入数据

1.条形图实例



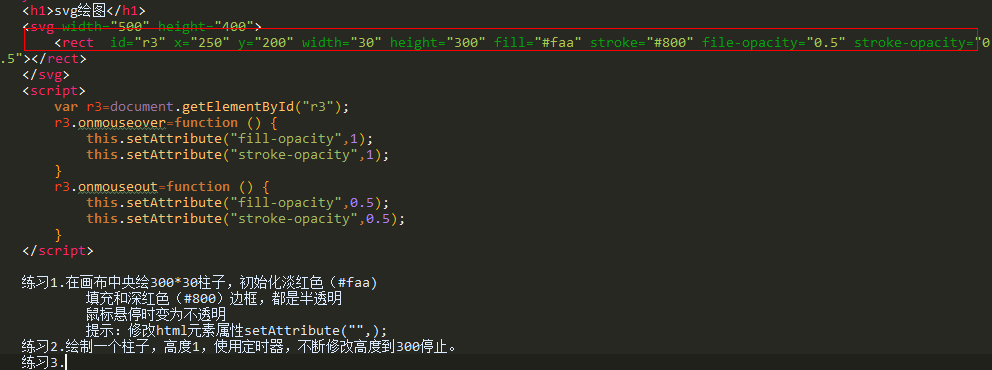
2. echart在异步中的应用格式



**3. 饼状图格式？**



## 2.svg绘图



异步数据变化

今天学习的内容

1:复习上周知识重点

2:作业:

[验证码;分页]

分页:商品分页管理

data/01\_productlist.php 查询数据库数据返回

(1)参数 pno页码 pageSize 页大小(一页几行)

1 10 默认值

(2)sql1:查询总记录数

SELECT count(lid) as c FROM xz\_laptop;

sql2:查询当前页内容

SELECT lid,lname,price FROM xz\_laptop

LIMIT (pno-1)\*pageSize,pageSize

(3)返回数据

{pno:3,pageSize:8,pageCount:6,data:[...]

product\_list.html 显示数据

product\_list.js 连接php获取数据，并且显示html

常见错误

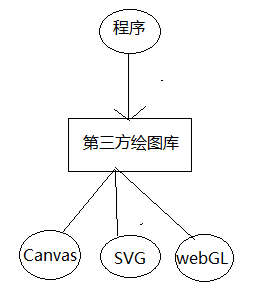
1: mysqli\_fetch\_row() expects parameter 1 to be mysqli\_result

原因:sql语句不正确

解决if(mysqli\_error($conn)){echo mysqli\_error($conn)}

3:今天学习的内容

## 1. h5新特性---第三方绘图工具库 echarts(canvas)



百度 echarts;d3;two.js;....

3.2:h5新特性---SVG绘图

3.2:h5新特性---SVG绘图--矩形

<svg id="" width="500" height="400">

<rect width="" height="" x="" y="" fill="" fill-opacity=""

stroke="" stroke-opacity=""></rect>

</svg>

练习1:在画布中央绘300\*30柱子，初始化淡红色(#faa)  
 填充和深红色(#800)边框，都是半透明  
 鼠标悬停时变为不透明,  
 提示:修改html元素属性setAttribute("",);  
 练习2:绘制一个柱子，高度1，使用定时器，不断修改

高度到300停止

练习3:使用ajax从服务器异步获取一段json数据

根据数据创建统计图

|  |
| --- |
| SVG 图形都是元素可以直接绑定事件监听  SVG 绘图特点  (1)所有图形默认只有填充色(黑色),没有描边色  (2)SVG图形的样式可以用元素属性声明，也可以用css 形式来声明，但是css声明只能使用svg专用样式，不能用css样式，如边框设置 stroke而不border  (3)图形可以用js对属性赋值,但不能使用HTML DOM形式，只能用核心 DOM操作  r3.x r3.width = 无效  r3.setAttribute("x",10);  (4)动态添加svg图形可以有两种方式  #HTML字符中拼接  var html = "<rect></rect>";  svg.innerHTML = html;  #创建元素  var rect = document.createElementNS(  "http://www.w3.org/2000/svg",  标签名);  svg.appendChild(rect); |

3.3: h5新特性---SVG绘图--圆形

<circle r="" cx="" cy="" fill="" fill-opacity="" ...>

练习1:在svg画布上填充五个圆形

四个角中央一个

练习2:在 svg画布上随机绘制30个实心圆形,

位置随机,填充颜色随机,透明度随机

点击某个圆形后，它慢慢变大、变淡直至消至消失

从DOM树删除

作业:验证码模仿二遍/分页/HTTP

滤镜：

# 二、Bootstrap

# 三、Nodejs

nodejs day1

今天学习的内容

1:复习上周的内容

2:作业:海葵/英特尔首页/学子商城商品列表

3:今天的目标

## 

## 1.阿里面试题:用户在浏览器中输入 www.taobao.com 直到看到网页\_\_之间发生了什么?☆

(1)操作系统(window/linux/macos)访问网络上DNS服务器，把域名转换IP地址

(2)浏览器通过ip向淘宝服务器发送HTTP请求

(3)淘宝服务器接收请求，解析请求，查找资源,

可能访数据库,构建响应消息，发送浏览器中

(4)浏览器接收并解析响应消息

(5)浏览器缓存接收响应主体，解析和渲染响应内容

## 2. 静态网页和动态网页

静态网页:网页内容任何人在任何时间访问都是不变的

HTML/CSS/JS/Flash/视频/音频/....

动态网页:网页内容不同人在不同时间访问可能是不同

JSP/PHP/ASP/Node.JS

JSP = html+java 功能强大适合大型或超大型项目{**卡车**}

2003=1个月[php+mysql];5w;50w-->mysql->oracle

500w->php->java[..]->1000w->服务器->小型机

oracle+小型机

集群+java+淘宝(mysql)-> 1亿/1day

PHP = HTML + PHP 简单易用;适合互联网项目[论坛] {家用轿车}

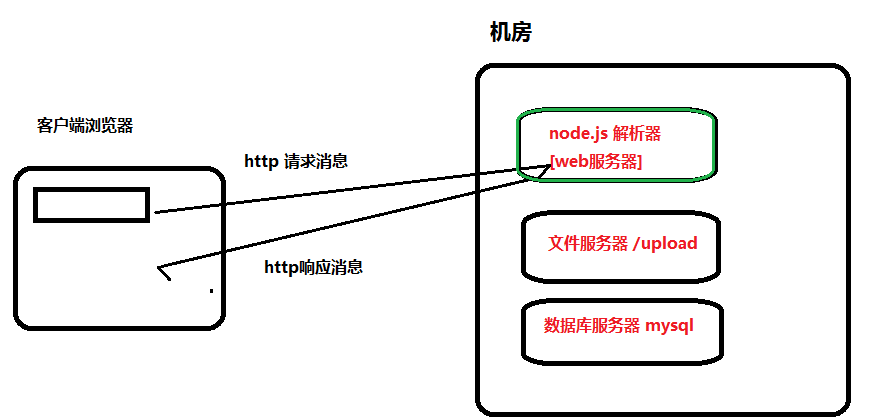
ASP = HTML+C# 微软产品,适全 windows  $$$$ {家用轿车} 300w

数据库 sql server 20w +开发工具+服务器

Node.js = HTML + "JS" 性能好,适合服务端IO密集型项目，不适合CPU

密集型项目 首页荐node.js {911}

|  |  |
| --- | --- |
| CPU密集型项目(计算复杂) | IO密集型(输入输出频繁) |
| 滴滴打车;高德导航;天气预报;石油预测分布 | I 输入 O 输出 数据获取，数据输出 |



## 3.node.js 基本概念/模块

Node.js不是JS。是一种运行于**服务器端开发平台**，它的开发语言使用 js

历史上第一个可以通吃前后台语言.

## 4. 淘宝镜像（node）下载安装

官方网站: www.nodejs.org

淘宝镜像: https://npm.taobao.org/mirrors/node

版本:0.12(16年初) 4.x(16年中)6.x(16年底)

node-v8.10.0-x64.msi windows下64位

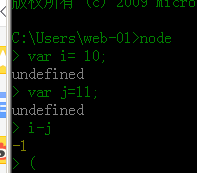
node-v8.10.0-x86.msi windows 下32位

node-v8.10.0.pkg 苹果电脑

## 5.Node.JS 两种运行模式及操作

### (1)交互模式---一般用于临时测试

node 回车

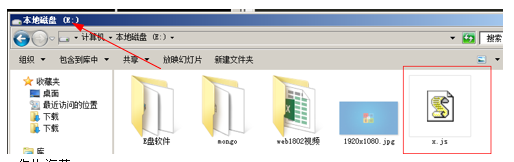
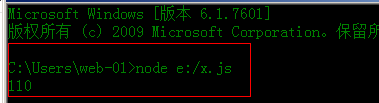


### (2)脚本模式--正式项目中使用方式

把要把的要执行语句编写一个文本文件(后缀任意，没有可行)

一次性提交node解释器执行

node x.js 回车



### 操作提示 : △用ws操作node时设置设置方法：

只要安装完node.js 重启一下webstorm，可以自动发现

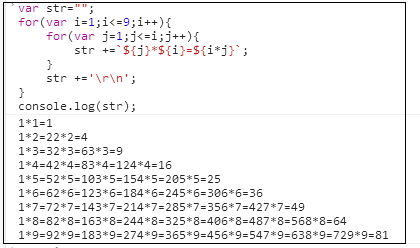
node.exe 解释程序,新建项目编号一定utf-8

File->Settings->Editor->File Encoding->

Global Encoding->UTF-8

Project Encoding->UTF-8

练习:创建02.js 打印九九乘法表, 以脚本模式来运行

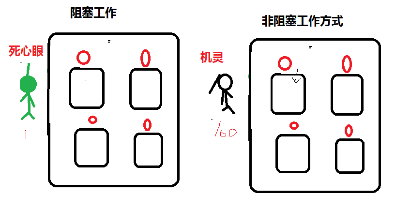


## 6. Node.JS 如何自学一门语言-- (node.js)

### (1)了解背景

--百度百科 node.js

[Node.js 使用事件驱动， 非阻塞I/O 模型]



### (2)搭建开发环境:编写Hello World

### (3)数据类型-----重点

红色的是对方没有的: 3 / 4

|  |
| --- |
| **前端js中数据类型**  1:基本/原生/值类型  string/number/boolean/undefined/null  2:引用/对象类型  ES对象类型:String/Number/Boolean/Math/Date/  RegExp/Object/Function/Error/...  BOM对象类型:window/document/location/history/..  DOM对象类型:Node/Element/Attr....  3:用户自定义:{} |
| **后端**nodejs中数据类型  1:基本/原生/值类型  string/number/boolean/undefined/null  2:引用/对象类型  ES对象类型:String/Number/Boolean/Math/Date/  RegExp/Object/Function/Error/...  3:用户自定义:{}  4:Node.js 十几万个第三方对象[https://www.npmjs.com/] |

### (4)变量和常量

var age = 20;

const PI = 3.14;

### (5)运算符

算术运算符

比较运算符

逻辑运算符

三目运算符

位运算符

赋值运算符 ( .= +=...)

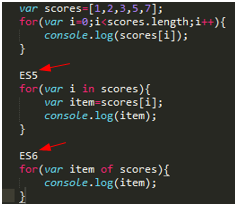
特效运算符 instanceof typeof .

### (6)逻辑结构

循环结构:for(;;); do{}while(); while(){}; for(in); for(..of..)

练习:03.js 声明一个保存5人学生成绩的数组,

使用三种for循环依次打印出每个成绩值



选择结构:if...else; switch...case

练习04.js 创建一个变量 var path = "/index"

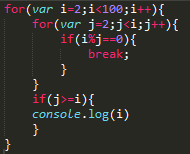
使用二种选择结构,判断path的值为哪种(/index;/search;/login)

调用不同的执行函数.



### (7)通用小程序 九九乘法表 100以质数 数组排序

练习:05.js 打印100以所有质数



### (8)函数和对象

### (9)常用的组件，第三方工具，框架

### (10)实际小项目

## 7.Node.JS 中特有概念--模块

|  |
| --- |
| Modal 模态框  Model 模型  Module 模块 |

### 1.用户自定义模块与运用

#### 1.文件模块 （自加代码及文件模块间导出引用运用方法）

一个web项目中可以分为多个不同"模块", 用户模块，订单模块 ，支付模块...

Node.js按照不同功能，将函数，对象保存在不同文件或者目录

这些文件或目录称 "模块"

Node.js中每个模块都是一个独立构造函数,解析器为每个.js

添加下代码.

(function(exports,require,module,\_\_dirname,\_\_filename){

exports:{} 用于声明向外部导出自己成员

require:fn 用于导入其它模块，创建指定模块对象

module:{} 指代当前模块对象!

\_\_dirname 二个下划线 当前模块所有属性目录绝对路径

\_\_filename 当前模块绝对路径

//自己编写代码

});

语法:

exports.属性名=值; #导出指定数据

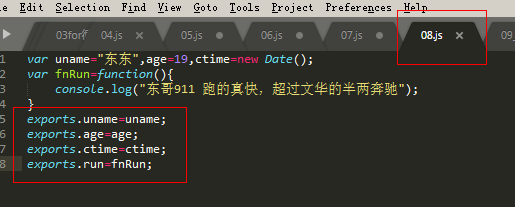
var m = require("./模块名称"); #导入指定模块并创建对象

练习:文件模块间 导出导入引用

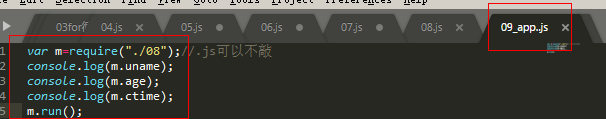
08.js 创建三个变量(uname/age/ctime)一个函数(fnRun)导出

09\_app.js 引入08.js 模块并且调用三个变量和一个函数

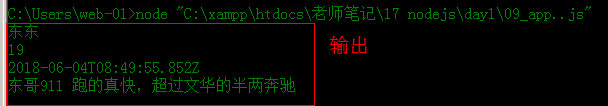
创建：导出



引用：



结果：



##### 导出的异同：exports和module.exports 前者导出对象失败

|  |
| --- |
| Node.js 模块中exports和module.exports的区别是什么?  二者都可以用于向外界导出自己的内部成员,但  module变量指代当前模块对象，真导出是 module.exports  node.js 底层 exports = module.exports  所以:若只是给exports添加新成员，则等价于module.exports添加新成员，但若修改了exports指向，则不会产生实质作用. |
| exports.age = 10; 等同于 module.exports.age = 10;  成功导出成员 |
| exports = {uname:"tom"} 导出失败  **module.exports** = {uname:"jerry"} 成功导出对象 |

练习: exports和module.exports 导出

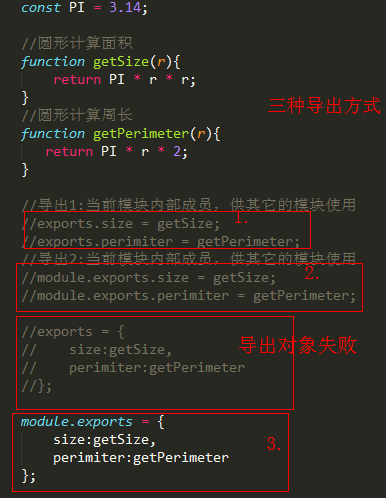
创建一个文件模块 10\_circle.js 其中定义常量PI，声明一个

方法 getSize(r) ,根据传入圆形的半径，返回其面积,

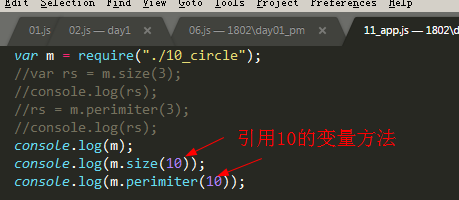
以及 getPerimiter(r); 返回其周长，再创建一个应用程序

11\_app.js 引入 10\_circle.js 模块，调用其公开二个成员方法

导出模块：



引用：



综合练习：模块在url中的拼凑运用

今写NODE.JS提供一个模块:URL

创建模块文件 myURL,向外导出一个方法resolve(url)

该方法接收的参数形式如下:

http://www.jd.com:443/ad/index?uname=tom&pno=3

返回对象:形如

{

protocol:"http",

servername:"www.jd.com",

port:443,

pathname:"/ad/index",

query:{uname:"tom",pno:3}

}

# 四、VUEJS

复习：箭头函数中this的指向问题

<button id="bt2"></button>

<script>

bt2.onclick = function(){

console.log(this); //BUTTON元素

}

bt2.onclick = ()=>{

console.log(this); //window

}

</script>

今日目标： day 01

(1)MVVM和双向绑定 —— 重点

(2)常用指令 —— 重点&难点

(3)组件 —— 掌握

## 1.三大框架前端

### 1.1. 理解：框架、Model、view、绑定（了解）...

框架的比喻：

|  |
| --- |
| DOM => jQuery  **Library**：库，是一套工具，本身不是项目，如jQuery，封装简化若干常用操作。  **Framework**：框架，是一套项目的半成品基本骨架，是前人优秀经验的结晶。先学会如何使用，反复实践，慢慢理解设计奥义。 |

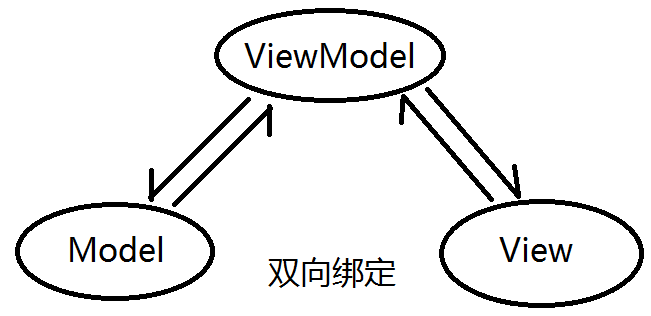
**框架**：Framework，是前人优秀开发经验的总结，是一个项目的半成品，便于后续的开发。

Vue.js/Angular/React是一个MVC/MVVM框架。

**Model**：模型，即页面中的数据，就是JS变量；

**View**：视图，即数据在页面中的呈现，就是**HTML/CSS**；

**ViewModel**：视图模型/控制器Controller，用于把模型和视图进行绑定，使得任何时候只要一方发生了改变另一方立即自动随之改变



var list = ['戴尔燃7000'，'联想大新', 'Apple Mac'];

ul>li\*3

table>tr\*3>td

div\*3

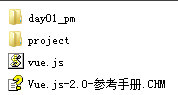
## 2.概述Vue.js

### 1.官网：

vuejs.org

中文官网：cn.vuejs.org （一般用中文官网就好：下载+学习）

### 2.手册：

day01 project js  

### 3. Vue.js的特点

是一个渐进式JavaScript 框架，可以在一个已有的大项目中一点点的慢慢渗透使用Vue.js。Vue.js的特点：简单易用，使用灵活，操作效率高。

## 3.语法：

### 1.Vue.js模板语法(Mustache大胡子):把Model绑定到View

#### 1.<ANY>{{ 模型数据 }}</ANY>



**注意：尽量绑定id：**



提示：{{username}}可以用方法{{username.toUpperCase}},

但推荐用过滤器{{username|uppercase}}

#### 2.Vue实例可能出现成员（完整写法）：

var vm = new Vue({

el: '#app',

data: {}, //数据

methods:{}, //方法，操作

watch:{}, //监听

computed:{},

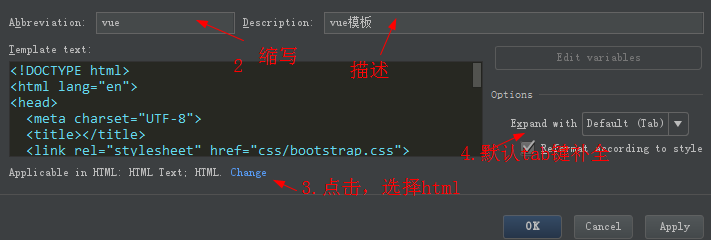
props:{}

});

console.log(vm); vm.$el vm.$data...

##### 2.1给webstorm添加vue基础模板





### 2.模型数据可以是如下形式的数据：

(1)string、number、boolean等原始值类型

(2)数组、对象、或者对象数组等复杂类型

(3)运算符表达式，如算术、比较、逻辑、三目等

### 3.实现数据绑定的写法：1 4

#### (1)模板语法： <p v-cloak>{{ msg }}</p>

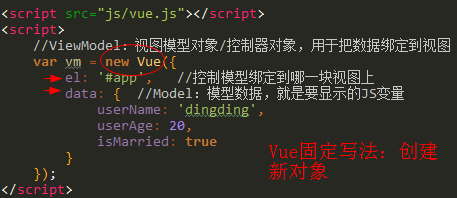
#### (2)Vue.js指令： <p v-html="msg"></p>

#### 绑定练习1：

把模型数据userAge绑定到一个<p></p>

把模型数据isMarried绑定到一个<span></span>





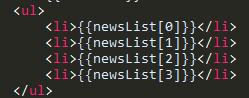
#### 绑定页面练习2——new Vue对象的结构：

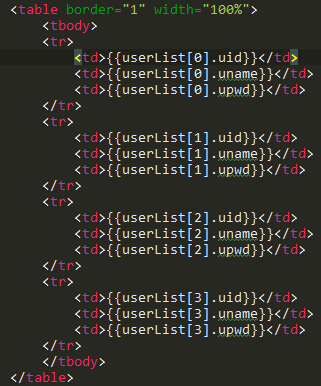
创建一个使用Vue.js的页面，

创建模型数据：newsList（字符串数组），包含4个新闻标题；绑定到视图中：ul>li；

创建模型数据: user（对象），包含uid、uname、upwd三个属性，绑定到视图中：span\*3

（可遍历）创建模型数据: userList（对象数组），每个对象包含uid、uname、upwd三个属性，共有4个对象，绑定到视图中：table>tr\*4>td\*3 1 2

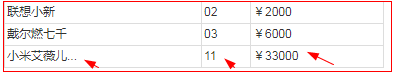
 



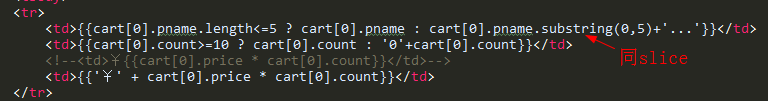


#### 练习3绑定数据形式——三目：

创建一个使用Vue.js的页面，创建模型数据： cart，保存三个商品信息，每个商品包含名称、单价、购买数量；把这些数据绑定在视图中：table>tr>td。提示：每行显示商品名称(最多显示5个字符，超过了就显示为...)、购买数量、价格小计(前面必须有￥提示符)



**提示：不用一个个tr写，可以用遍历指令，字符串用单引号引起来**



### 4..防止变量闪烁写法（隐身斗篷）1 4

#### (1)普通防抖动写法：v-cloak {{msg}} +[v-cloak]{display:none}

#### (2)防止抖动写法：标签内写,v-html="msg"

## 4.指令Vue.js中的 (Directive) 13个

**指令13**

1. v-text

2. v-html

3. v-show

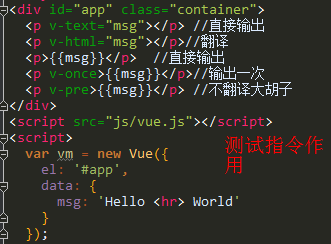
4. v-if v-else-if v-else

5. v-else

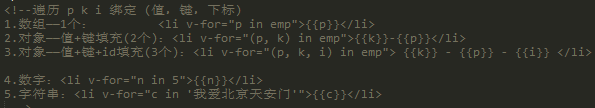
6. v-else-if

7. v-for ="(v,k) in arr/obj"

测试指令作用：防止混淆！

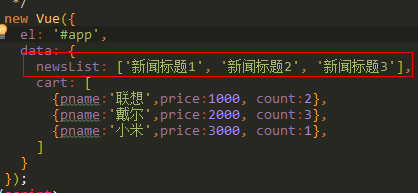
 

4 v-for 遍历形式 引用元素和下标：



创建模型数据：newsList，数组中保存4个新闻标题，使用循环绑定到ul>li中

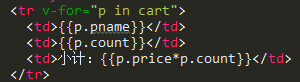




5 v-for 多级 对象[0]形式填充：

创建模型数据： cart，保存三个商品信息，每个商品包含名称、单价、购买数量；使用循环把这些数据绑定在视图中：table>tr>td

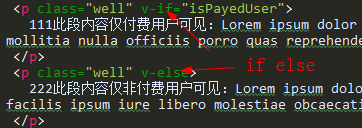
答案：创建对象上面！



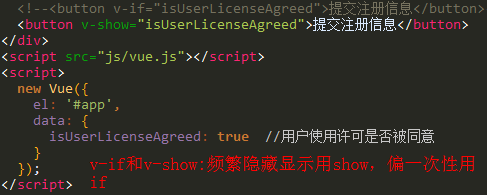
6 v-if vs v-show形式显示、else if形式写法：

创建模型数据： userScore，如果用户积分达到5000分，页面中显示<p>白金用户<p>；否则如果达到3000分，页面中显示<p>黄金用户<p>；否则如果达到1000分，页面中显示<p>白银用户<p>；否则显示<span>普通用户<span>

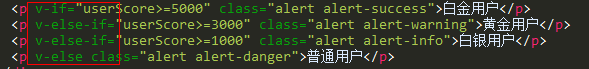
ps：内容可见设置：

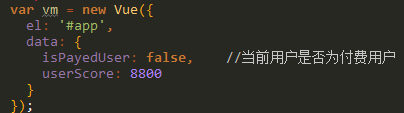


ps：if vs show （是否频繁：show 同意，显示按钮）



答案：

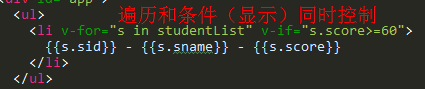




7 v-for + v-if、对象.下标 填充：

创建模型数据：studentList，保存4个学生的信息，包括学生编号、姓名、语文成绩。把学生信息绑定到一个ul>li，注意只显示成绩及格的学生信息

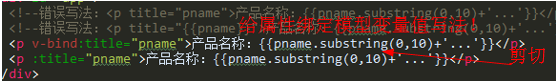
提示：<li v-for="" v-if=""> (数组对象)

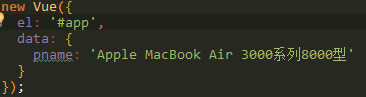




8 v-bind属性绑定变量值、剪切省略显示写法：

创建模型数据：imgList，保存4张图片的URL字符串，将上述数据绑定的ul>li>img上



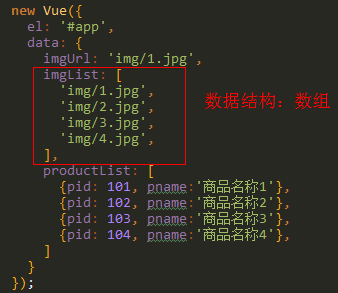


提示：同slice用法

9 v-bind属性绑定变量缩写、属性为超链接——超链接内有:字符串+变量时的写法：

创建模型数据：productList，保存4个商品的信息，每个商品都有编号和名称两个属性，将上述数据绑定到table>tr>td上，每个商品最后添加一列“删除该商品”，该超链接跳转到 'delete.php?pid=\*'





课后练习：完成学子商城后台管理系统中的“商品列表功能”



思路：html——php返回的数据格式

### 8. v-on 监听v-on 为当前元素绑定事件监听函数

#### 1.写法：

<button v-on:click="f2"></button>

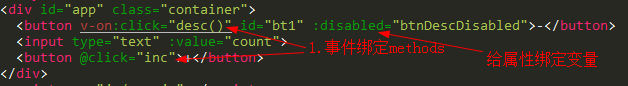
<button @click="f2"></button>

提示：v-on指令可用的修饰符：.prevent、.stop、.once、.13

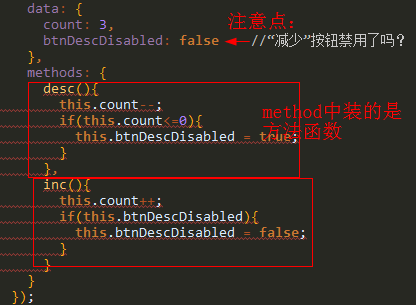
练习1：事件绑定加方法（改属性值） v-on / @

注意“-”按钮的disabled属性false/true情况

1.绑定



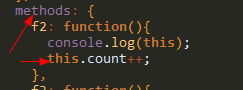
2.data中创建methods和属性默认值false



练习2：事件绑定加方法（改数据变量） v-on / @







注意：this指元素（除了创建的函数方法是箭头函数：会指向window）

#### 2.事件修饰符：https://cn.vuejs.org/v2/api/#v-on

如果不想在函数中自己改，可以用修饰词控制函数事件

.stop - 调用 event.stopPropagation()。

.prevent - 调用 event.preventDefault()。



.capture - 添加事件侦听器时使用 capture 模式。

.self - 只当事件是从侦听器绑定的元素本身触发时才触发回调。

.{keyCode | keyAlias} - 只当事件是从特定键触发时才触发回调。

.native - 监听组件根元素的原生事件。

.once - 只触发一次回调。

等价v-once

.left - (2.2.0) 只当点击鼠标左键时触发。

.right - (2.2.0) 只当点击鼠标右键时触发。

.middle - (2.2.0) 只当点击鼠标中键时触发。

.passive - (2.3.0) 以 { passive: true } 模式添加侦听器



键盘：

### 9. v-bind

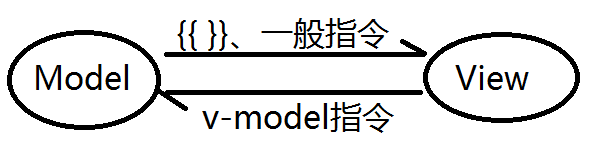
<a v-bind:title=“msg” :id=“uid”> </a> 绑定模型数据

<a v-bind:title="msg" :id="'link'+uid" :href="'detail.php?id='+uid"></a>

<a title="欢迎访问学子商城" id="link101" href="detail.php?id=101"></a>

### 10. v-model

v-model：用于实现View到Model方向的数据绑定

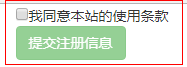
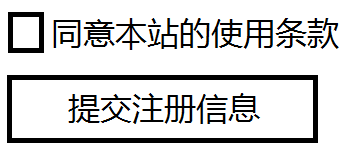


可能发生变化的View只有表单元素！也就是说只有表单元素可以声明v-model指令。

提示：1)v-model绑定其实是“双向绑定”，它既可以把模型数据绑定到视图，也可以把 视图上的数据改变绑定到模型 2)v-model绑定的模型变量必须在模型中声明

练习1：data绑定多处——提交注册信息+同意条款绑定：

（练习：勾选则可以点击，否则不能点击



**VIEW: checkbox =>MODEL: isAgreed =>VIEW: button:disabled**）

disabled=false + checked=false一变跟着变（用在true/false）





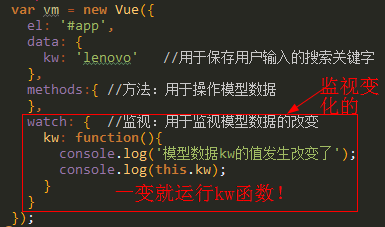
练习2：data绑定——前后对应变动 1



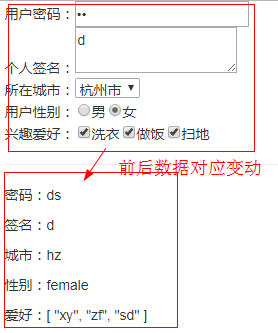
1.实现同变效果



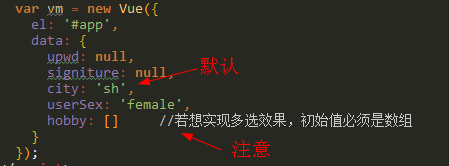
2.watch：监视变化——每变一次运行一次内部的函数（发ajax可以用到，每变一次就发ajax）



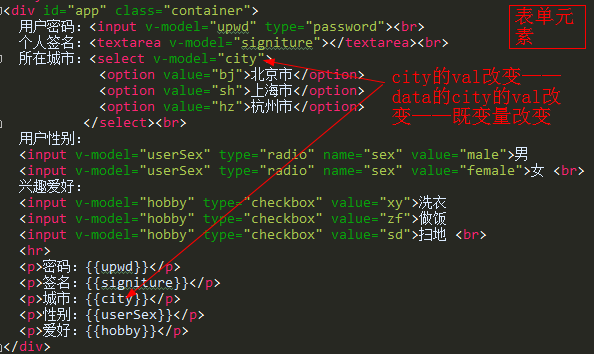
练习2.2：data绑定——前后对应变动（select——val）



1.绑定val（v-model）为模型变量



2.val改变——模型变量值改变（返回对应变动数据）——（后期的watch也会改变）

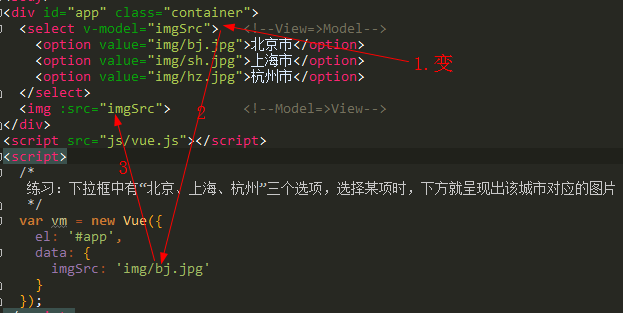


练习2.3：data绑定——前后对应图片变动（select——val）

(练习：下拉框中有“北京、上海、杭州”三个选项，选择某项时，下方就呈现出该城市对应的图片

**VIEW: select => MODEL: imgSrc => VIEW: img)**

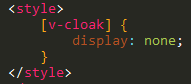




11. v-pre 完全不要编译，跳过

12. v-cloak +[v-cloak]{display:none}





13. v-once v-once只渲染一次，下次模型数据改变就不跟着变了



另：

为当前元素绑定事件监听函数，只监听一次

<button v-on:click.once="f2"></button>

等价于

<button v-once:click ="f2"></button>

Vue.js提供了十几个指令——用在HTML元素上，作为它们的扩展属性

|  |  |  |
| --- | --- | --- |
| 指令名 | 含义 | 说明 |
| <ANY v-html="模型数据">  </ANY> | 使用指定的模型数据替换当前元素的innerHTML |  |
| <ANY v-text="模型数据">  </ANY> | 使用指定的模型数据替换当前元素的innerText |  |
| <ANY v-cloak>  {{模型数据}}  </ANY> | 配合 [v-cloak]{  display: none  }使用 | 当前元素若还没有渲染出指定的模型数据，则具备v-cloak属性；否则就会删除该属性。 |
| <ANY v-pre>  {{模型数据}}  </ANY> | 元素中的模板语法将不被解析，直接显示为{{ }} |  |
| <ANY v-for="变量名 in 模型数据">  </ANY> | 使用指定的模型数据循环输出当前元素 | 此处的模型数据可能是数组、对象、数字、或字符串 |
| <ANY v-if="模型变量">  </ANY> | 如果指定的模型数据可以转换为true，则在当前DOM树上挂载指定元素；否则就从DOM上删除指定元素 | v-if和v-for若同时出现在同一个元素上，v-for的优先级更高，先执行 |
| <ANY v-else-if="模型变量">  </ANY> |  |  |
| <ANY v-else>  </ANY> |  |  |
| <ANY v-show="模型变量">  </ANY> | 如果指定的模型数据可以转换为true，则显示指定元素；否则隐藏指定元素 | v-if：挂载或删除元素，适合于不需要反复隐藏/显示的元素  v-show: display : none靠CSS隐藏或显示指定元素，适合于需要反复隐藏/显示的元素 |
| <ANY v-bind:标签属性名="模型变量">  </ANY> | 把模型变量的值绑定到指定的HTML元素属性上 | 注意：  可以简写为（直接用： 表示）  <ANY :标签属性名="模型变量">  </ANY> |

vuejs day02

#### 复习day01：

Vue.js是一个渐进式框架，采用了MVVM模型：

**Model**：模型，即数据，JS变量

**View**：视图，即数据展示，HTML/CSS

**ViewModel**：视图模型，用于把M和V绑定，代替DOM操作

<div id="app"> //View：呈现

<p>{{msg}}</p>

<p v-html="msg"></p>

</div>

<script>

var vm = new Vue({ //ViewModel：控制器

el: '#app',

data: { //Model：数据，js变量

msg: '欢迎访问学子商城',

uid: 101

}

});

</script>

今日目标：

(1)补充更多的Vue.js指令

(2)Vue.js应用中实现XHR异步请求

(3)自定义Vue.js指令

(4)自定义过滤器

(5)Vue.js组件

## 5.绑定属性Class和Style： V-model—data—绑定属性

口诀：V-model——data——绑定属性

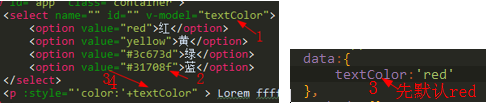
class3种style3种：

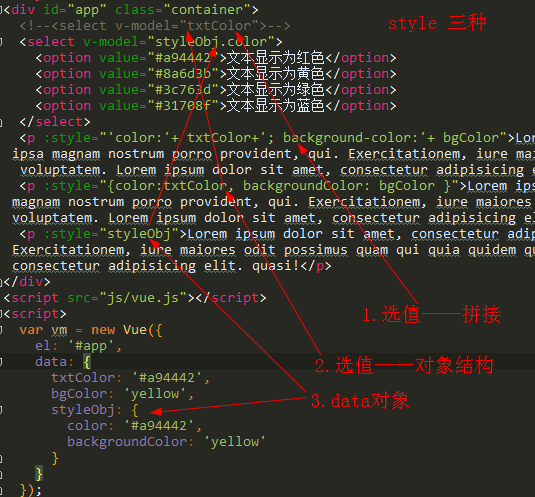
|  |
| --- |
| <p :style=" 'color:'+ txtColor "></p> 1.拼接  <p :style="{ color: txtColor }"></p> 2.对象选属性值  <p :style=" styleObj "></p> 3.data对象结构styleObj+data:{styleObj:{ }} |
| <p :class="(isTxtRed?'text-danger ':'') + (isBgRed?'bg-danger ':'')"></p> 1.三目添加  <p :class="{'text-danger': isTxtRed, 'bg-danger': isBgRed}"></p>  2. class="{'class名':true/false,'class名':true/false}"  <p :class="classObj"></p> 3.classObj: data:{classObj:{ }} |

练习3：style绑定3种

方法1：拼接方式

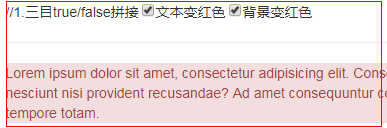


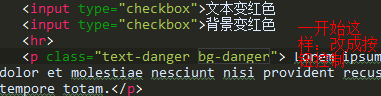


练习4：class的绑定：3种方法

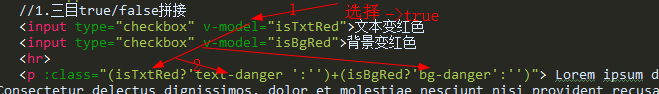
练习：使用两个复选框，分别控制“文本显示为红色(text-danger)”和“背景显示为红色(bg-danger)”

<p :class="classObj"></p>



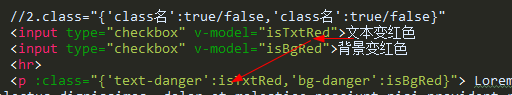


1.三目添加

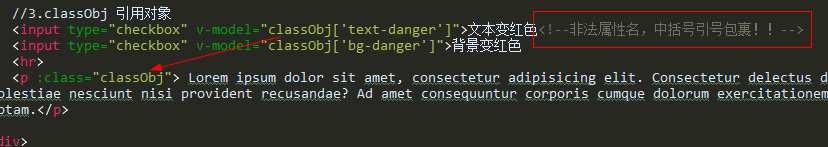


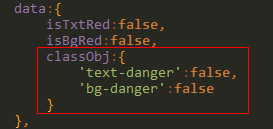


2：class="{'class名':true/false,'class名':true/false}" 写法方式



3.classObj class="对象"





## 6.指令自定义 Vue.js中的——了解（有过滤器了）

已有: 13个要熟！

### 1.指令写法：

<ANY v-指令名:参数.修饰符="参数值"></ANY>



Vue.js共提供了13个预定义指令。

### 2.用户也可以自定义扩展指令：

Vue.directive('指令名', {

bind: function(){ },

**inserted: function(){ },** //找到元素，运用方法

update: function(){ },

componentUpdate: function(){ },

unbind: function(){ }

})

1.自定义：教程<https://cn.vuejs.org/v2/guide/>

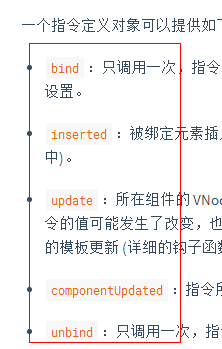


2.会用第二个钩子函数**inserted**就可以了：*（了解可以不理）*

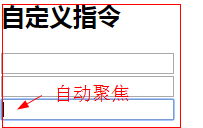
钩子函数会自动传过来一些参数



|  |
| --- |
| 钩子函数：Hook Function，指一个对象可以对外提供的多个内部函数（有些可以提供，也可以不提供），这些函数在特定的情况下会被自动调用。 |

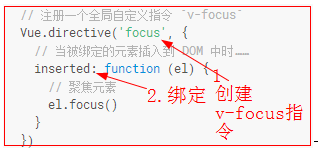
 <https://cn.vuejs.org/v2/guide/custom-directive.html>

练习：自定义自动获得焦点 vuejs day2 12



（基本自定义能力：会创建结构——会用inserted及事件函数创建）

创建指令——绑定到父——钩子函数传过来的el绑定聚焦事件



3使用指令+创建控制器





完整代码：



## 7.过滤器自定义Vue.js中的 —— 掌握

**（可以自己定义指令，相当于封装，了解）**

Filter：过滤器，用于将数据进行格式上的过滤(格式转换)。过滤器的本质就是字符串处理函数：接收原始的字符串，返回处理后的结果字符串。

### 1.可以在两种情形下使用过滤器：

(1) {{ 模型数据 | 过滤器1 | 过滤器2 | 过滤3 }}

(2) <p v-bind:title=" 模型数据 | 过滤器1 | 过滤器2 ">

注意：| 读作“管道”，通过管道把当前数据传递给下一个过滤器

声明一个自定义过滤器：

Vue.filter('过滤器名', function(val){

//对原始数据val进行处理

return 处理后的结果;

})

练习1过滤男女01 day2 14

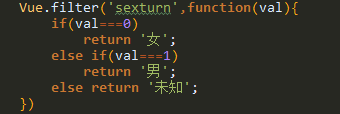
（练习：页面从后台获取的员工性别是0或1或其它，创建过滤器sex， 把0转换为“女”、1转换为“男”、其它转换为“未知”。

假设模型数据：sexList，其中有5个员工性别：[ 0, 1, 1, 0, 2]，绑定到ul>li中）

1.引用



2.filter结构



3.数据



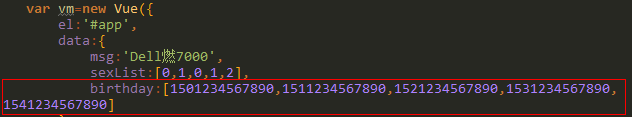
课后任务：

练习2 年月日过滤成yyyy-mm-dd hh:mm:ss day3 2（类似）

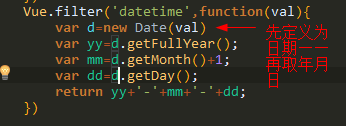
(1)数据库中保存日期时间都是长整型数字，如1501234567890。创建过滤器datetime，把传入的长整数转换为'yyyy-mm-dd hh:mm:ss'形式的字符串。

假设模型数据：birthdayList，其中有5个员工生日：[1501234567890， 1511234567890， 1521234567890， 1531234567890， 1541234567890]，绑定到ul>li中

1.数据



2.过滤器



3.引用



(2)Vue.js发起异步请求有三种方法

1) 第三方库jQuery.ajax( ) —— 杀鸡用牛刀

2) 官方插件Vue-Resource —— 官方已经废弃

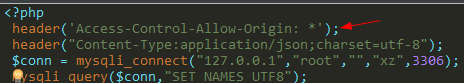
3) 第三方工具Axios —— 推荐方案

官方：https://www.npmjs.com/package/axios

自学Axios的使用，完成学子商城后台系统“商品管理功能点”

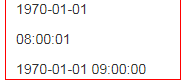


小提示：跨域异步请求的url：要加init.php的 header

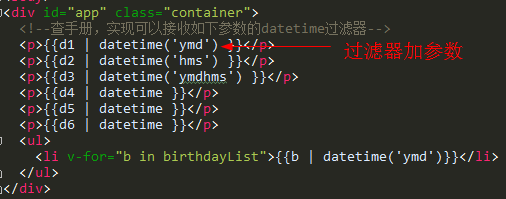


### 2. 给过滤器 加参数

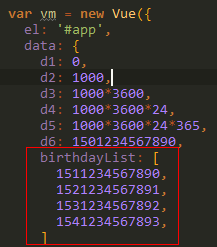
练习：计算员工日期，返回年月日格式 day03 2



1.引用+过滤器名+参数



2.做过滤器

 3数据

vue.js day03

复习：

见思维导图

今日目标：

(1)Vue.js实例、声明周期等小知识

(2)Vue.js中自定义组件 —— 重点&难点

## 8. axios get then异步请求Vue.js中发起

|  |
| --- |
| Vue.js生态系统（Ecosystem）：可以说包括很多组件吧~用Axios组件可以做做异步  Vue.js、Axios、Vue-Router、Vuex、MintUI、ElementUI、Weex.... |

1.如何在Vue.js中发起异步请求

(1)使用第三方工具库： jQuery.ajax( )

(2)使用官方插件：Vue-Resource

(3)使用第三方工具： Axios

var url = 'http://127.0.0.1/productlist.php?pno=2&pageSize=8';

axios.get(url).then(function(res){

console.log('异步请求成功');

console.log(res);

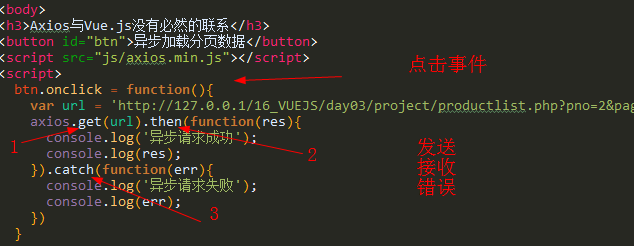
}).catch(function(err){

console.log('异步请求失败');

console.log(err);

})

练习异步示例代码：4步 day3 3



## 9. Vue成员实例可能出现 6

var vm = new Vue({

el: '#app',

data: { }, //数据

methods: { }， //方法，操作数据

watch: { }, //监听，监听数据的改变 add：监视变化

computed: { },

props: { },

beforeCreate(){ },

created(){ },

beforeMount(){ },

mounted(){ },

beforeUpdate(){ },

updated(){ },

beforeDestroy(){ },

destroyed(){ }  
 })

consoe.log(vm); => vm.$el vm.$data vm.$watch 可以这样引用到（不是window）

## 10. Vue.js实例的生命周期 —— 重点理论

面试要会说！



Vue.js实例的生命周期分为四个阶段：

(1)创建(create)阶段

(2)挂载(mount)阶段

(3)更新(update)阶段

(4)销毁(destroy)阶段

为了监听四个阶段，Vue.js提供了八个钩子函数

beforeCreate(){ } 实例创建之前自动调用

created(){ } 实例创建完成自动调用

beforeMount(){ } 实例挂载到DOM树之前调用

mounted(){ } 实例挂载到DOM树之后调用

beforeUpdate(){ } 实例中模型数据发生改变需要更新DOM之前调用

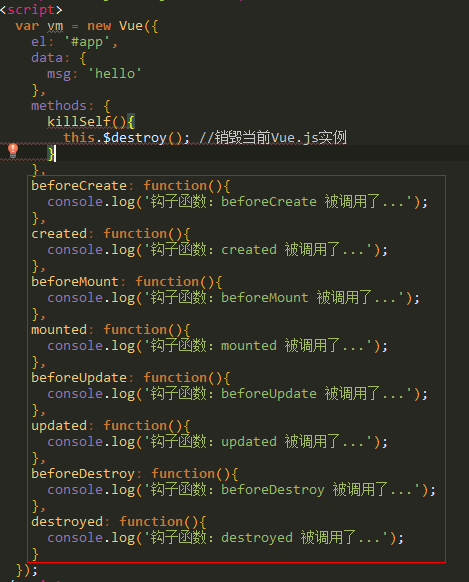
updated(){ } 实例中模型数据发生改变需要更新DOM之后调用

beforeDestroy(){ } 实例被从DOM上销毁之前调用

destroyed(){ } 实例被从DOM上销毁之后调用

var vm = new Vue({ }); —— 此处的vm是**根组件(root)**

测试vue钩子函数执行顺序：一开始是创建，挂载。msg发生改变时，



## 11.组件自定义Vue.js中的—— 重点

Component：组件，是可复用的Vue实例，在视图中呈现为用户自定义的扩展标签。项目中，组件是页面中的一块区域，该区域中对应专有的模型数据、HTML/CSS结构。

|  |
| --- |
| **面向组件式的开发：**把大网页划分为若干组件组成的区域，每个区域都有专属性数据、HTML元素、CSS样式。  Component Tree |

自定义组件有两种方式：

### (1)全局组件：可以在页面中任何一部分使用的组件

Vue.component('comp-name', {

template: ' ', //组件模板

data: function(){

return { } //专有的Model数据

},

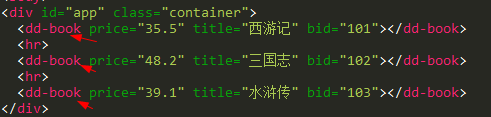
props: ['属性1', '属性2', ...] //自定义属性(也是Model)

})

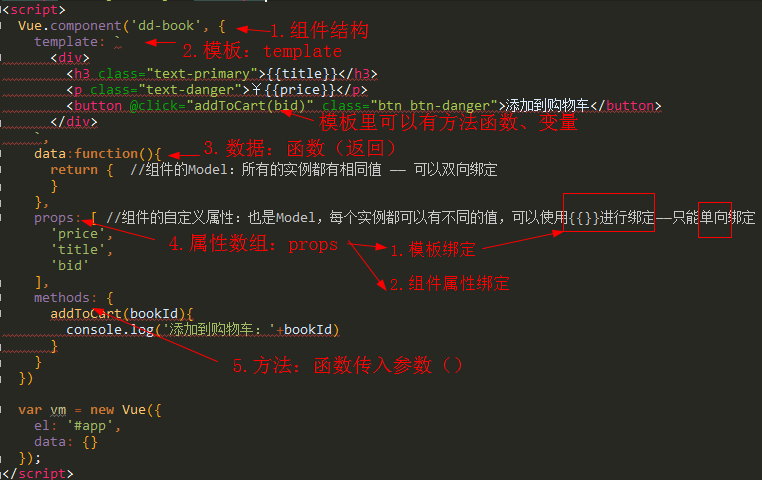
例子：1.引用（可多次）

练习：书籍组件





2.做组件 5要：模数属法建



### (2)局部组件：只能用于特定的父组件内的组件

Vue.component('comp-name', {

template: ' <child-comp-name/> ', //局部组件只能用在父组件内部

**components:** {

'child-comp-name': { //局部组件

template: '',

data: function(){return {}}

}

}

})

例子：1.引用

练习：员工组件——+局部组件 day3 10 (可遍历)





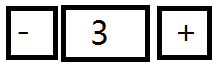
2.全局组件包局部组件——局部组件的模板导入全局。



做组件时可能遇到的错误：

|  |
| --- |
| [Vue warn]: Error compiling template:  <h3>戴尔燃7000</h3>  <span class="text-info">￥6800.00</span>  - Component template should contain exactly one root element.  每个组件的模板内只允许有一个根元素 |
| [Vue warn]: The "data" option should be a function that returns a per-instance value in component definitions.  组件的data属性必须是一个函数，返回一个对象，其中包含组件中的专属数据——这一点与根组件不同：根组件只有一个实例，而每个自定义组件都可能被实例化多次，每个这样的组件实例都必须有自己专有的模型数据 |

练习：创建一个自定义组件：<buy-count></buy-count>

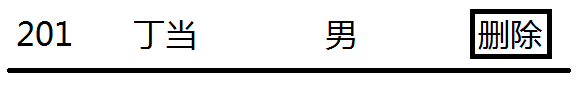


练习：创建一个自定义组件：<dd-book></dd-book>



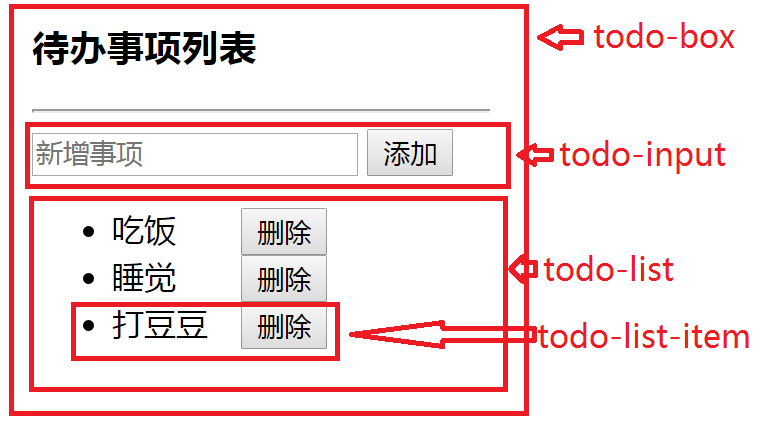
练习：创建一个自定义组件：<tedu-emp></tedu-emp>，

每个该组件实例都可以声明eid、ename、sex三个自定义属性



页面中实例化4个该组件对象

课后练习：待办事项列表——3组 Todo List day4 2



vue.js day04

复习：

见思维导图

## 12.HTML标签、属性、事件名称的大小写：

### 1.HTML标签名：

不分区大小写，推荐纯小写；①官方标签只包含**小写字母/数字**；②自定义标签**只包含小写字母/数字+至少一个减号**

<textarea></texterea>

<buy-count></buy-count>

### 2.HTML标签属性名：

不分区大小写，推荐纯小写；属性中只包含小写字母/数字，不能有减号！

<a href="" title="" style=""></a>

<buy-count productcount=""></buy-count>

### 3.HTML标签事件名：

不分区大小写，推荐纯小写；事件中只包含小写字母/数字，不能有减号！

<a onclick="" @mouseover=""></a>

<buy-count @userincrement=""></buy-count>

--------------------------------------------------------------------------------

### 4.ES6中对象的简写方法：

var eage = 30;

var ework = function(){}

var emp = {

age: uage,

work: ework

}

var age = 30;

var work = function(){}

var emp = {

age: age,

work: work

}

var age = 30;

var work = function(){}

var emp = {

age,

work

}

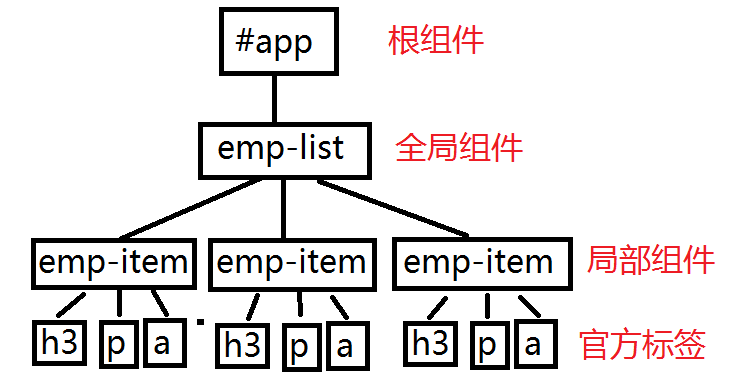
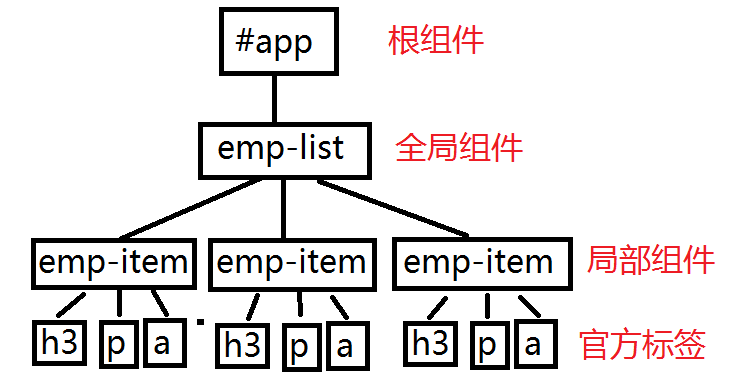
今日目标：

(1)组件间通信 —— 难点

(2)SPA和Vue-Router —— 重点

## 13.组件间通信 —— 难点

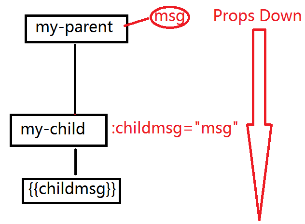
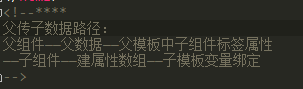
练习：根据下图写出全局组件和局部组件

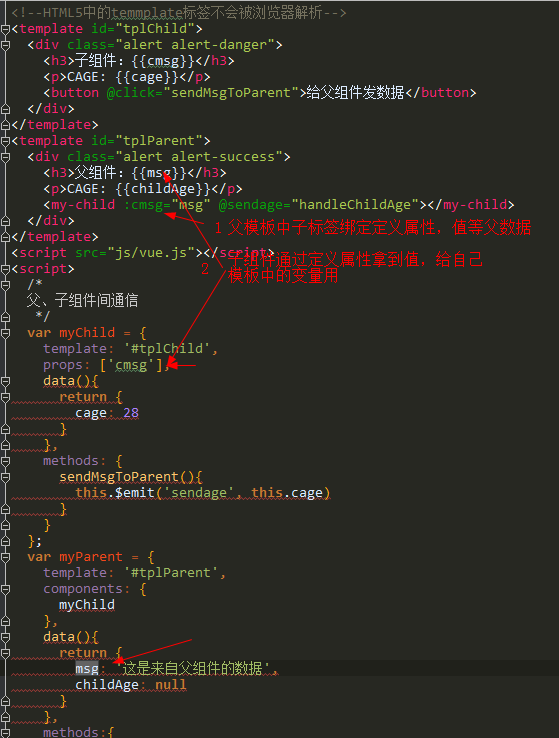


### 1.Vue.js中组件直接传递数据 —— 重点&难点&面试题

#### (1)父组件给子组件传递数据**day04 4-5-6**

"Props Down"：通过为子组件声明自定义属性，父组件中为该属性绑定模型数据，实现父=>子传数据效果。



#### (2)子组件给父组件传递数据**day04 7-8**

"Event Up"：子组件发射自定义事件，并携带数据；父组件监听子组件可能抛出事件，并提供处理函数

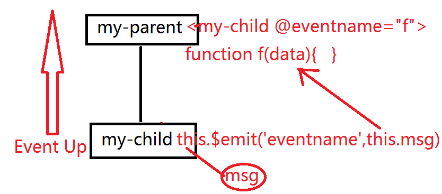
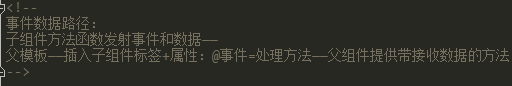
子组件：this.$emit('自定义事件名', **事件数据**);

父组件：<my-child @自定义事件名="func">

func(**事件数据**){

//接收到子组件发射的事件，并处理数据

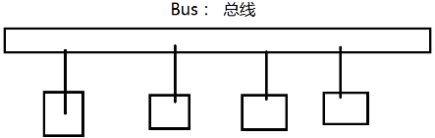
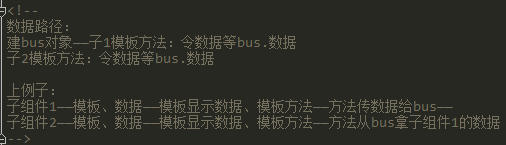
}

#### (3)兄弟组件间传递数据 —— 实际应用场景比较少**day04 13**

"Bus"：通信总线，可用于多个平级的兄弟间传递数据

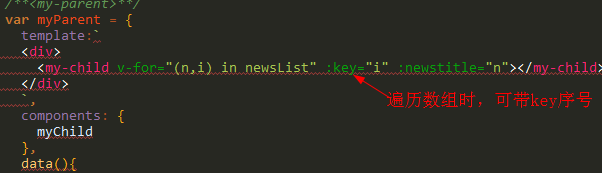
var bus = new Vue();

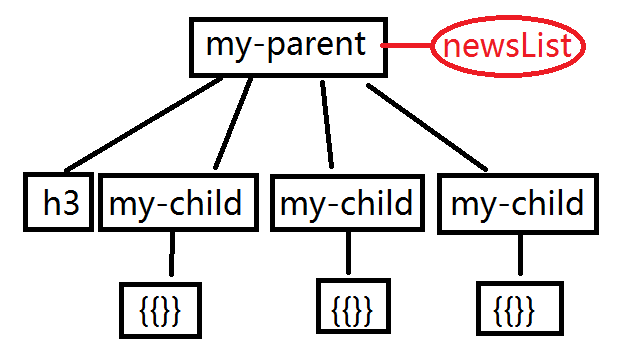
练习：在父组件内有模型数据：newsList: ['新闻标题1', ....]

day4 4

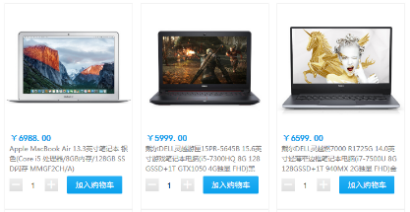
注意：

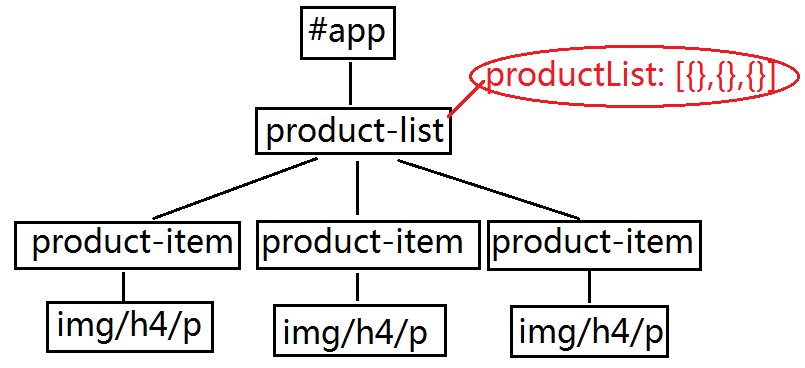


根据此数据，渲染出多个子组件，其模板如下：

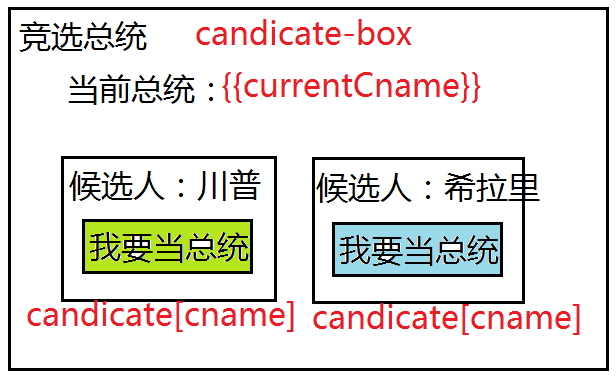


练习：使用父子组件实现商品列表：day4 5-6

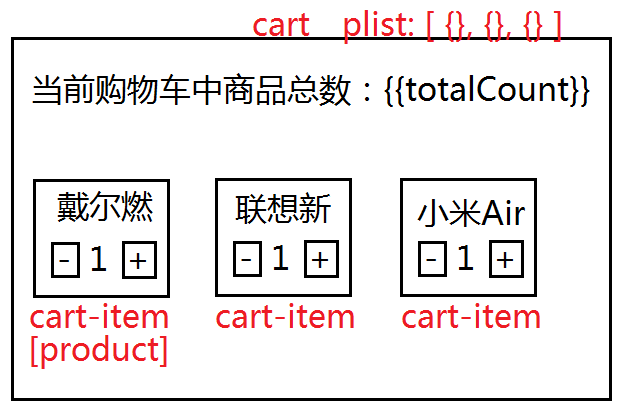
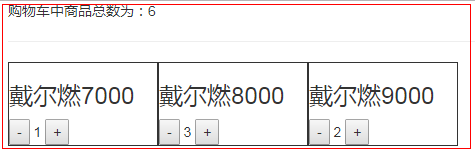




练习子传父：竞选总统day4 8



练习子传父：购物车计算器day04 11-12

问题day4 11-12区别?

### 2.小知识：Vue.js中的计算属性 day4 8

<p>{{totalCount}}</p>

var component = {

computed: {

totalCount: function(){

return 数据;

}

}

}

注意：计算属性的本质是函数，但比methods中的函数优越在：多次访问同一个计算属性仅会调用一次。

### ▲总结：组件间通信的三种方法（面试）

#### (1) Props Down, Event Up day5 1

通过自定义属性把父组件的数据传递给子组件；

通过自定义事件把子组件的数据传递给父组件；

数据传递范围可控，推荐使用。

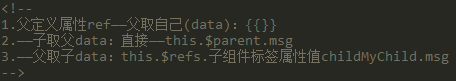
#### (2) $refs 和 $parent day5 2

<my-parent ref="childRef">

父组件获得子组件的引用： this.$refs.childRef

子组件获得父组件的引用： this.$parent

数据传递范围不可控，不推荐使用。



#### (3) 通信总线(Bus) day5 3-4

var bus = new Vue()

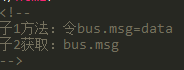
bus.$emit('事件', 数据)

bus.$on('事件, fn)

用于在兄弟组件间传递数据。

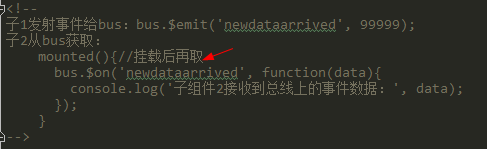
一种：





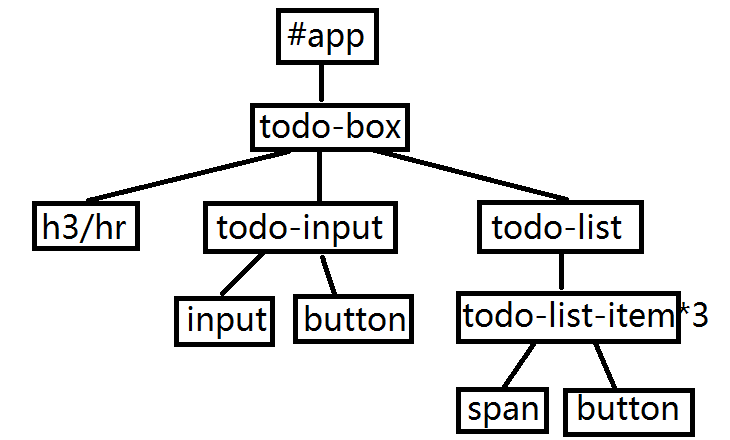
二种：





### 综合练习：TODO List day5 9

(1)分析组件结构



(2)显示已有列表项：

Props Down todo-box(msgList)=>todo-item

(3)添加新的列表项

Event Up input(userInput)=>todo-box

(4)删除已有列表项

Event Up todo-list-item=>todo-list=>todo-box

课后练习：

(1)完整实现Todo List

(2)完整实现“学子商城后台产品列表功能点”

vue.js day05

复习：

见思维导图

## 14. 模板的声明方法组件中的：用第三种

常用第三种

### (1)js中直接建模板

<script>

var c = {

template: `<div>innerHTML...</div>`

}

</script>

### (2) script引用模板

<script type="text/x-template" id="myTpl2">

<div>innerHTML</div>

</script>

<script>

var c = {

template: '#myTpl2'

}

</script>

### (3)标签template模板

<template id="myTpl3">

<div>innerHTML</div>

</template>

<script>

var c = {

template: '#myTpl3'

}

</script>

## 15. 动画效果Vue.js中的

Vue 在插入、更新或者移除 DOM 时(v-if、v-show、v-for、组件间切换等)，提供多种不同方式的应用过渡效果。

包括以下工具：

(1)在 CSS 过渡和动画中自动应用 class

(2)可以配合使用第三方 CSS 动画库，如 Animate.css

(3)在过渡钩子函数中使用 JavaScript 直接操作 DOM

(3)可以配合使用第三方 JavaScript 动画库，如 Velocity.js

### 方式1： CSS过渡(transition) (练习)

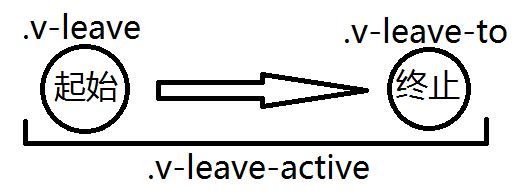
**口诀：消失 显现 +name ：.name-leave-active**

Vue.js为添加在<transtion>标签中的即将消失的DOM元素提供了三个class：day5 5

.v-leave-active：离开动画进行中的class

.v-leave：即将离开时的class，动画的第一帧

.v-leave-to：离开动画终止时的class，动画的最后一帧



Vue.js为添加在<transtion>标签中的即将显现的DOM元素提供了三个class：day5 5

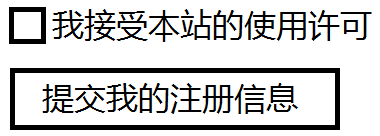
.v-enter-active：进入动画进行中的class

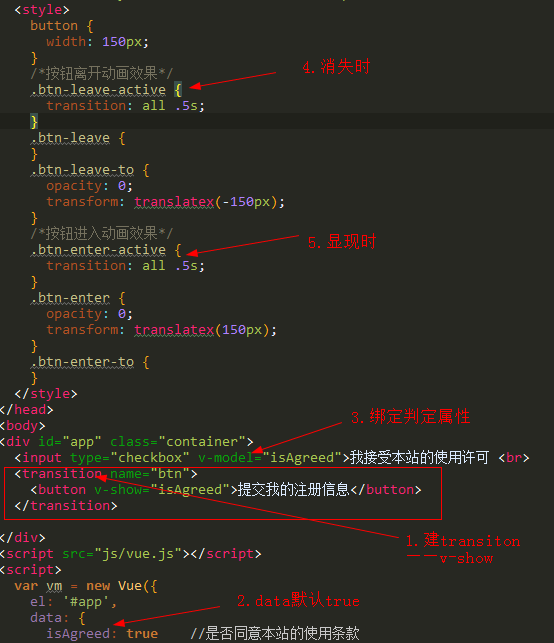
.v-enter：即将进入时的class，动画的第一帧

.v-enter-to：进入动画终止时的class，动画的最后一帧

提示：若页面中有多个需要实现不同动画效果的元素，只需要给<transtion>添加不同的name属性即可，上述六个class的前缀不再是v-，变为name值。

练习：“不接受许可”则按钮向左滑动离开并渐隐；“接受许可”则按钮从右面滑动进入并渐显。

day5 6



#### 午间自学：Animate.css —— 第三方动画效果库

下载Animate.css文件：<https://daneden.github.io/animate.css/> day5 css



|  |
| --- |
| Animate.css是一个使用CSS3 Keyframes实现的动画效果库，提供了几十个常用的动画效果，使用非常简单：day5 7  <ANY class="animated 动画效果类">  <ANY class="animated infinite 动画效果类"> |
|  |

动画效果关键词：

bounce flash pulse rubberBand

shake headShake swing tada

wobble jello bounceIn bounceInDown

bounceInLeft bounceInRight bounceInUp bounceOut

bounceOutDown bounceOutLeft bounceOutRight bounceOutUp

fadeIn fadeInDown fadeInDownBig fadeInLeft

fadeInLeftBig fadeInRight fadeInRightBig fadeInUp

fadeInUpBig fadeOut fadeOutDown fadeOutDownBig

fadeOutLeft fadeOutLeftBig fadeOutRight fadeOutRightBig

fadeOutUp fadeOutUpBig flipInX flipInY

flipOutX flipOutY lightSpeedIn lightSpeedOut

rotateIn rotateInDownLeft rotateInDownRight rotateInUpLeft

rotateInUpRight rotateOut rotateOutDownLeft rotateOutDownRight

rotateOutUpLeft rotateOutUpRight hinge jackInTheBox

rollIn rollOut zoomIn zoomInDown

zoomInLeft zoomInRight zoomInUp zoomOut

zoomOutDown zoomOutLeft zoomOutRight zoomOutUp

slideInDown slideInLeft slideInRight slideInUp

slideOutDown slideOutLeft slideOutRight slideOutUp

### 方式2：配合Animate.css动画库实现动画

<transition

enter-active-class="元素进入的动画效果类"

leave-active-class="元素离开的动画效果类"

>

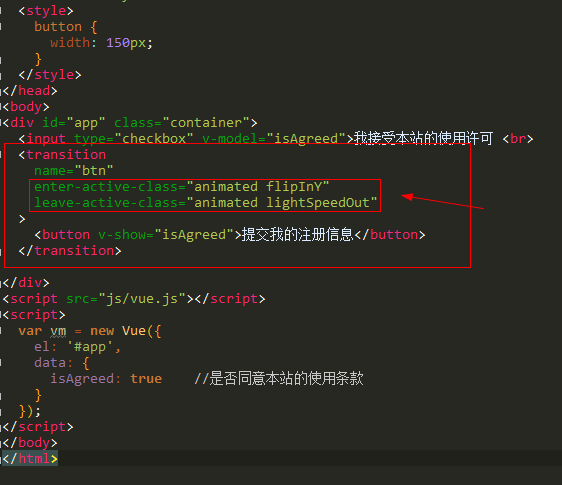
<ANY></ANY>

</transition>

提示：自己不用写class了（day5 6）:

.v-leave-active {all 1s linear}...——> leave-active-class="animated lightSpeedOut"

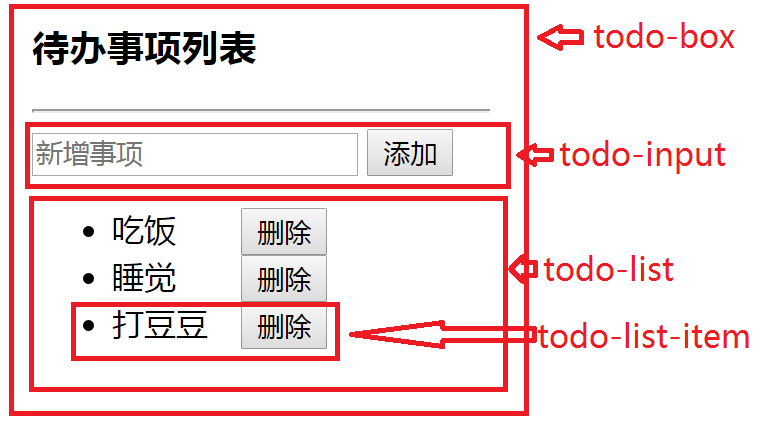
例子：day5 8



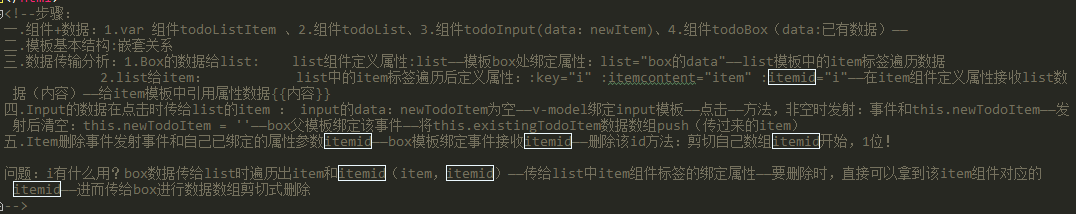
## 父子通讯综合练习：Todo List 经典分析

4步分析：day5 9





可以直接看代码分分析解决：



**(1)设计基本的组件包含关系**

**(2)显示已有待办项(Props Down x 2)传了两级路径：**

todoBox:existingTodoItem =>

todoList:list /(item in list) =>

todoListItem:itemcontent

**(3)添加新的待办项(Event Up)**

v-model + @click => $emit('useradditem', newTodoItem)

todoBox: @useradditem=fn

**(4)删除已有的待办项(Event Up x 2)**

todoListItem: this.$emit('userdeleteitem', itemId)

todoList: this.$emit('listdeleteitem', itemId)

todoBox: this.existingTodoItem.splice(itemId, 1) 从数组中删除1个元素

## 16.Vue.js阶段项目 —— “学子商城后台管理系统”框架重构

重构：Refactor，指开发中对于已有代码使用更好的方式“重新实现”一遍，体现设计和实现方面的优势，以更好的适应未来的应用场景。

除了Vue.js还需要两方面的辅助技术：

(1)SPA技术

(2)单文件式组件技术

### 1.SPA应用

Single Page Application：单页面应用，指整个项目有且仅有一个页面具备完整的HTML DOM树（引入外部各种资源）；不同的内容可以放在不同的“模板页面”中(只是一段HTML DIV片段而已)。

|  |  |
| --- | --- |
| **传统的多页应用** | **SPA应用** |
| 项目中的多个网页，每个都是完整的HTML内容 | 整个项目只有一个页面具有完整的HTML结构，其它来回切换的“页面”其实只是一段DIV片段 |
|  |  |
| 每次浏览器都要加载完整的DOM结构 | 每次所谓的“页面切换”其实只需要改变DOM树的部分内容，加载和渲染速度更快——尤其适用于移动端应用 |
| 页面间切换就是旧DOM树的删除和新DOM树的加载和重建——不可能有“过场动画”效果 | “页面切换”其实只是两个DIV在同一棵DOM树上切换，可以实现任意的过场动画 |

#### 1.SPA应用的实现原理简介：

#### (1)定义“路由词典”：

[

{path: '/login', component: {template:'<div>...'}},

{path:'/main', component:{template:'<div>...'}},

...

]

#### (2)定义唯一“完整”的HTML页面：index.html

<html>

<head>js/css...</head>

<body>

....

<特殊的接口></特殊的接口>

....

</body>

</html>

#### (3)客户端发起请求：

##### 1：分析url

http://127.0.0.1/index.html#/login

浏览器会先请求index.html创建DOM树，然后解析location.hash，若发现以'#/'开头，查找路由词典，动态异步加载指定的模板片段即可。

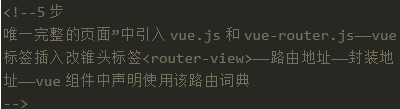
##### 2.“页面跳转”其实只是在修改#/后的内容：（插件Vue-Router）

<a href="http://127.0.0.1/index.html#/main"，浏览器根本无需删除/重建DOM树，只需要再次查找路由词典，异步请求对应的组件模板，替换掉“特殊的接口”中已有的内容。

|  |
| --- |
| 说明：Vue.js本身并不具备SPA应用功能，必须依赖于官方提供的“插件”——Vue-Router |

##### 3.Vue-Router插件的使用步骤：5步

day5 10（未加地址链接）看截图步骤（简化）



(1)在“唯一完整的页面”中引入vue.js和vue-router.js

(2)在“唯一完整的页面”中指定用于挂载各种“模板页面(各种改锥头)”的接口：

<router-view></router-view>

(3)在JS中定义完整的“路由词典”——就是各种组件与请求地址的对应关系

var routes = [

{ path: '/', component: xx }, //默认路由地址

{ path: '/login', component: xx },

{ path: '/main', component: yy },

]

(4)定义路由器，封装路由词典：

var router = new VueRouter({

routes

})

(5)在根组件中声明路由器

var vm = new Vue(){

el: '#app',

router

}

课后练习：

(1)再次重现Todo List练习。

(2)自学VueRouter手册，通过“超链接”和“JS”两种方式实现模板页面的跳转，编写出类似学子商城“管理员登录页”和“管理员监控主页面”的页面跳转效果。

vue.js day06

复习：

见思维导图

|  |
| --- |
| 提示：如何给Vue.js添加一个扩展插件：  Vue.use( 插件对象 )  例如：Vue.use( VueRouter )、Vue.use(VueResource)... |

##### 4.VueRouter中实现“页面”切换 2标签 1js

###### (1)传统的超链接方式

<a href="#/路由地址">

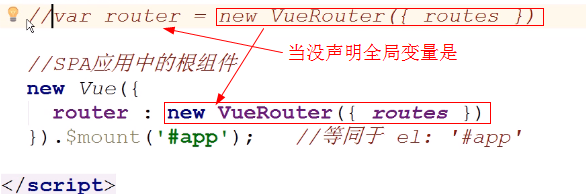
###### (2)VueRouter提供的超链接方式

<router-link to="/路由地址"></router-link>

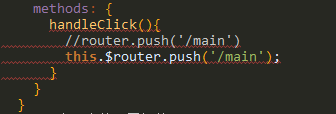
###### (3)使用JS跳转方式（推荐第二种）

1.种：使用全局变量：router.push('/路由地址')（有声明全局变量的写法时才能用）

2种：在组件内部使用：this.$router.push('/路由地址')





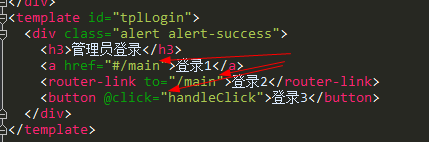


多种写法：不用this，直接写标签事件中：



练习：VueRoter链接3种方式：day6 1

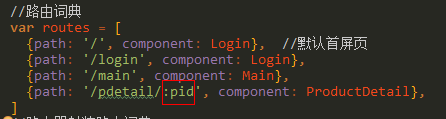




##### 5.VueRouter中编写带参数的路由 3步

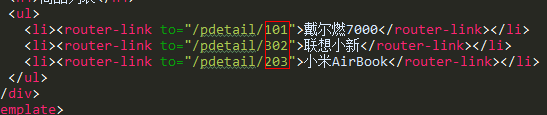
###### (1)声明路由词典时需要指定路由参数

{ path: '/weatcher/**:cityId**/**:date**', component: ... }



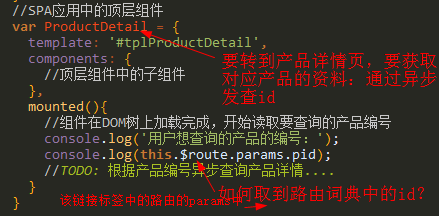
###### (2)模板页面跳转时必须为每个路由参数提供具体的值

<router-link to="/weatcher/**1011**/**20181015**">



###### (3)组件中获取路由参数

this.$route.params: {cityId: '1011', date: '20181015'}



练习：使用VueRouter和路由参数实现：

创建组件 NewsList，包含模型数据 nlist: [{nid:, title:},...]，对应的模板中显示ul>li>a，显示出所有的新闻。

创建组件 NewsDetail，根据路由地址中的参数nid，显示出该新闻的编号即可： 101号新闻详情为:....

<!--  
vuerouter跳转页面步骤：  
1.词典绑定定路由参数——2每个标签连接中要有具体参数值  
——3.拿id发异步请求：在路由词典中拿id方法：this.$route.params.id  
-->

##### 6.VueRouter中实现“嵌套式路由”

在一个路由组件中继续嵌套更多的子路由组件

var routes = [

{path: '/login', component: Login},

{path: '/main', component: Main, children: [

{path: 'productlist', component: ...},

{path: 'productadd', component: ...},

{path: 'userlist', component: ...},

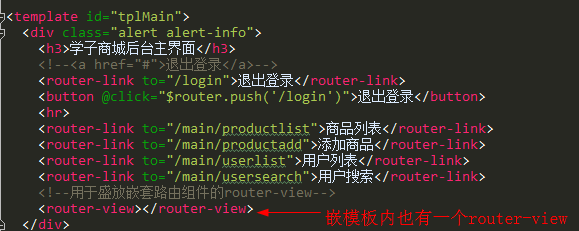
{path: 'usersearch', component: ...},

]}

]

注意：1.所有声明了嵌套路由的组件，其模板内部必须有 <router-view></router-view>

2.子路由路径不用加 / 了





##### 午间任务：练习

###### (1)部署“学子商城后台管理系统V1”

管理员登录验证PHP页面：

http://127.0.0.1/16\_VUEJS/day06/xz\_admin\_v1/data/admin\_login.php?aname=admin&apwd=123456

商品列表PHP页面：

http://127.0.0.1/16\_VUEJS/day06/xz\_admin\_v1/data/product\_list.php?pageSize=8&pno=2

###### (2)自学官方教程“单文件组件”相关内容

##### 7.Vue.js的两种使用方法 17还是7？？？？

###### (1)直接使用<script>引入

简单直接；但是所有的组件都声明在一个文件中，不便于维护和分工协作。

###### (2)使用Vue-CLI工具 Command Line Interface 命令行插入

|  |
| --- |
| Vue CLI工具是Vue官方提供的项目构建工具，运行于命令行中，运行此命令可以在当前目录下创建一个基本的Vue项目，其中还包含了Vue.js/VueRouter + Node.js/Express + Babel + Webpack + 辅助工具.... —— 该项目的运行不依赖于Apache，且该项目支持“单文件式组件”:  **Login.vue = template + script + style** |
| 提示：Vue-CLI工具V2.x和V3.x的区别很大！  https://cli.vuejs.org/ |

使用步骤：

(1)下载全局工具@vue/cli

npm i -g @vue/cli

默认下载到c:/users/当前用户名/AppData/Roaming/npm下

(2)运行@vue/cli工具，在当前目录下创建一个空白项目

vue create 项目名

(3)进入项目目录，运行该项目

cd 项目名

npm run serve //执行当前目录下package.json文件中scripts/serve指令

Vue-CLI工具创建的项目模板目录结构 —— 重点

xz\_admin\_v2

|\_node\_modules 项目所依赖的第三方模块

|\_package.json 项目的基本配置文件 scripts/seve

|\_public 放置可以被客户端直接访问的静态资源

|\_index.html SPA项目唯一完整的HTML文件

|\_favicon.ico 可以被index.html直接引用静态资源文件

|\_src 项目的源代码

|\_assets 可以被.js/.vue引用的静态资源文件(图片/css/js)

|\_components 用户自定义组件(单文件组件)，公共子组件

|\_views 用户自定义组件(单文件组件)，SPA单页组件

|\_App.vue 项目的顶级组件

|\_main.js 项目的入口文件，声明了根组件 new Vue({})

|\_router.js 项目的路由器配置文件

|  |  |  |
| --- | --- | --- |
|  | **Node.js** | **ES6** |
| 向模块外部导出一个对象 | module.exports = {  } | export default {  } |
| 引入另一个模块对象 | var m2 = require('./xx.js') | import m2 from './xx.js' |

如何在Vue组件文件中引入外部CSS/JS文件？

<script>

import '../assets/css/app.css'

import '../assets/js/jquery.js'

</script>

课后练习：“学子商城后台管理系统”Vue.js重构

仿照Login.vue的实现步骤，实现Main.vue的内容(只需要实现统计信息展示页面即可)

angular day3

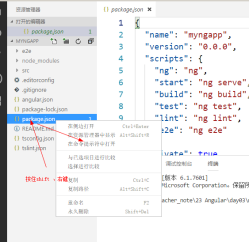
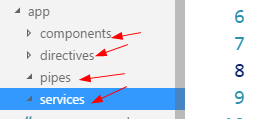
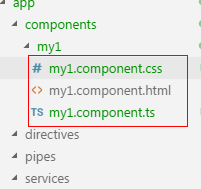
|  |
| --- |
| Visual Studio Code下载：code.visual.studio.com |

读：VS Code

##### 1.9:40 vs code使用

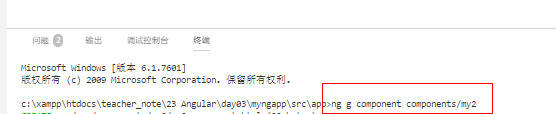
|  |  |
| --- | --- |
| 快捷键 | |
| ctrl+b | 文件栏隐藏/打开 |
|  |  |

1.打开空模板——按shift+右键在window中运行：npm run start——VScode中删除git类文件，后期速度快一些——（改颜色主题）——（字体大小：ctrl&+ ）——搜索插件（angular v6）安装

2. ——打开空白的项目：浏览器：输入127.0.0.1:4200——app下建文件夹4个——component下建my1文件夹——my1下建文件——模块自动补全，输入：a-c，会有提示

3.在html中加点div，内容等——创建、声明、使用组件10:13——

查看浏览器？？？

4.自动创建方式10:32：

##### 2.TypeScript中的特有语法

2.1.强类型

2.2声明属性成员的3种方式

##### 3.Angular自定义指令3步 day03 app directive

<input >

##### 4.Angular 过滤器/管道 day03 app app my4 uppercase 14.26

管道API及说明：https://angular.cn/api?type=pipe

14.40 （4）date 格式截图：my4

my5

##### 5.16.25前？

##### 6.服务

快捷补全：a-s

ctrl+小 ` 号打开cmd——建模块：ng g module XzCart

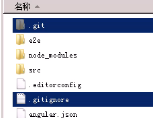
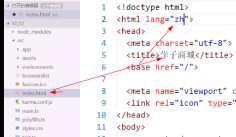
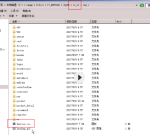
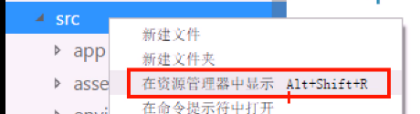
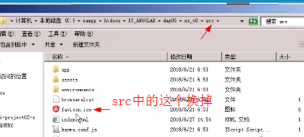
observable异步创建方法：day04 视频15.56

1.直接在vs运行后就可以在浏览器打开，不要dom？ctrl+` ——ng run start

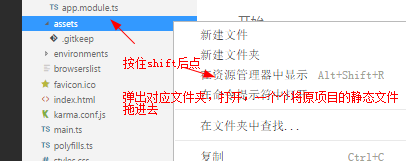
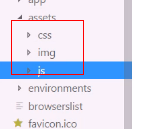
项目重构步骤：day5pm

步骤：

1.用脚手架创空白项目ng new xz\_v2（或直接文件夹粘贴完成后vscode打开）

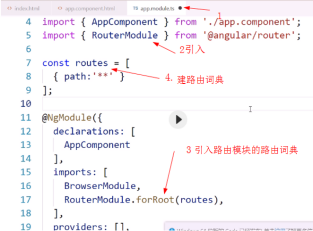
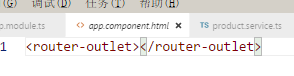
——名字改成——.git文件删掉——用vscode打开——改index.html的东西——换网title图片（在原项目包的img中有）放在中的

2.静态资源移植到assets目录（三个：css，js，img）

（矫正：不用按shift）————了解index.html中的路径base：指定页面中所有相对路径的基础值——了解angular.json中的一些东西：一般都别动

3.改路由功能：

步骤：

app.module.ts中引入路由模块方法见图——app组件html加总路由出口

4.启动项目npm start/ng serve -o ——创建组件：

notfound、

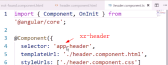
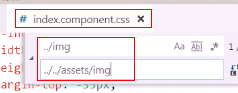
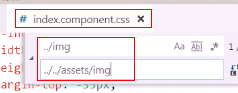
header（无需路由地址）、

footer（无需路由地址）、

index

步骤：

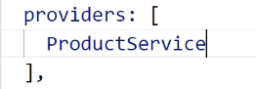
——1.建notfound组件——在app.module.ts的路由词典中配置该模块——项目启动npm start——在notfound组件html中写好404页面——在该组件的.css中设置一下样式

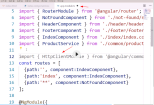
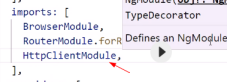
——.2建组件：header、footer组件（css文件分析：1.给index组件的html中引入<xz-header> <xz-footer>（header和footer没有单独样式文件样式已经合并在base.css中了）2.index中的css文件有些是公共的如（）放在公共index.html中移入，有些是index页面自己用的，复制粘贴在index页面的.css中（注意有些要改成Angular的路径））——建header、footer组件——两个组件的html代码粘贴上去——（footer同理）header模块的selector组件名改成<xz-header>——在index组件html中引入header组件（footer同理）——（footer同理）header.html路径更换——在公共index.html中移入公共css样式——在index组件.css文件中导入主页css（方法2种：1复制进去2导入文件）——css中如有路径可能会报错，改成Angular中的组件路径如index.component.css中的路径（全部替换）

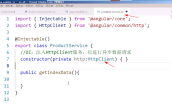
5.建service提供getIndexData()方法：

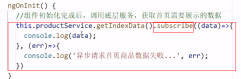
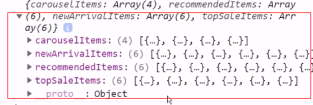
步骤：

**——1**.ng g service common/Product （解析：建product服务放在common文件夹中）

**——2**.找到app.module.ts模块在providers中声明服务

**——3**. 提供getIndexData()方法，分析：方法要异步就要用到httpclient服务——app.module.ts导入模块HttpClientModule  ——并在app.module.ts的imports中声明HttpClientModule（跟路由同类型）

**——4.**找到product.service.ts注入服务HttpClient服务（异步请求用）——ProductService中创建方法getIndexData（）{}方法通过http拿到数据返回数据给服务模块

**——5**. 为index提供数据：index.component.ts首页组件注入productService方法获得数据——在index.component.ts的钩子函数ngOnInit中监测首页数据方法是否返回结果（subscribe（（）=>{}））——返回一个大对象要绑定到html页面中

**——6**解开对象**（**一个大对象绑定麻烦，在组件中用几个属性接收对象中的几大块）——声明几个属性接收数据——接收——

##### 填数据步骤：

0.找到代码片段放入模块：可能在html中，可能在js中

1.src、href属性要绑定，且不加{{}}



2.img路径加：'../../assets/'+ （框架中的路径和原项目在服务器中的路径）



3.￥ 用管道直接实现



4.用包裹内容：在有数据时才显示

ng 项目day2am

改index.component.css中css 的所有background渐变：把-webkit-linear..放到background的集群的最后（angular才能读出来）

——轮播：——加载中标签+\*ngIf="!carouselItems"——有轮播广告时<ng-container \*ngIf="carouselItem">+li标签遍历（\*ngFor="let c of carouselItems;let i=index"）——li里嵌套a（绑定属性[href]=c.href）</~>——轮播显示最后一个？——li加行内样式[ngStyle]="{display:i>0?'none':'auto'}"........

——公共index引入公共css slid.CCS...

（day2am10.35改vscode快捷键）

——