正课：

1. 什么是Vue

2. 如何使用Vue

3. MVVM

4. 绑定语法

5. 指令

一. 什么是vue:

1. Vue是第三方开发的渐进式的基于MVVM设计模式的纯前端js框架

(1). 渐进式: 可以在项目中逐步使用vue的各个组件和功能，可以和其它技术轻松混搭。

(2). 基于MVVM设计模式:

(3). 纯前端: 只需要浏览器就可以运行，无需任何后端技术的支持。

2. 何时: 以数据操作为主的项目都可用vue实现

3. 为什么: 原生DOM vs jQuery函数库 vs Vue框架

(1). 原生DOM: 优: 万能; 缺: 繁琐

(2). jQuery函数库: 优: 简答; 缺: 只简化了单个步骤，没有彻底简化流程。

(3). Vue框架: 优: 终极简化，自动化开发

缺: 需要转变做事儿的思路，难以接受

4. 兼容性: 所有框架不再支持旧浏览器(IE8)

二. 如何使用Vue：

1. 下载: 2种方式：

(1). 直接下载vue.js文件，引入网页使用 —— 前三天

vue 2.5

开发版: 未压缩的，拥有完备的注释，代码格式和见名知意的变量名。更重要的是，有极其友好的错误提示。但是体积大。——只适合开发和学习之用

生产版: 压缩后的，去掉注释和代码格式，极简化变量名。更重要的是，删除了所有错误提示！但是，体积小——适合生产环境使用。

(2). 脚手架 —— 后两天

vue 3.0

2. 我的第一个vue功能:

(1). 先引入vue.js:

<script src="js/vue.js"></script>

(2). 编写静态界面：

a. 整个静态页面只能包在一个唯一的父元素内，并起一个id。

<div id="app">

<button @click="doit(-1)">-</button>

<span>{{n}}</span>

<button @click="doit(+1)">+</button>

</div>

b. 页面所有可能发生变化的位置，都改成{{变量名}}

c. 页面中所有触发事件的元素都添加@事件名="处理函数(参数)"

(3). 定义一个专门保存页面所需数据的对象:

页面中有几处变化，需要几个变量保存数据，对象中就要保存几个属性变量来保存初始值:

var data={ n:0 };

(4). 创建一个vue实例对象,包含界面元素和数据对象，以及事件处理函数

new Vue({

el:"#app",//找到页面上唯一的父元素<div id="app">

data:data,//将保存数据的对象放入new Vue中，和界面放在一起。

methods:{//所有事件处理函数放在methods中

doit(i){//doit:function(){...}

this.n+=i;

this.n<0&&(this.n=0)

}

}

})

三. MVVM

1. 什么是MVVM: 是对前端项目各个组成部分的重新划分

2. 为什么: 旧的前端项目的划分方式已经不能满足新的需求

旧的前端项目分为三部分：

(1). HTML: 定义页面的结构和内容

(2). CSS: 定义页面中内容的样式

(3). JS: 为页面添加行为

问题：因为HTML和CSS是静态语言，缺少程序语言必须的要素。所有动态变化和行为都要依靠js来添加。导致js的负担非常重：重复代码和重复操作非常多！

3. MVVM如何重新划分前端项目：

也是分为三部分:

(1). 界面(View): HTML+CSS

增强版的: 支持变量，循环，分支等程序要素

(2). 数据模型(Model): 其实就是一个保存所有页面所需的变量的对象——data={...}

(3). 视图模型(ViewModel): 将界面(View)和数据模型(Model)绑定起来。结果，数据模型中的值如何变化，界面上就跟着自动变化。

4. MVVM的绑定原理:

从new Vue()起:

(1). 构建响应系统:

a. 将数据模型对象data，引入new Vue()中

b. 打散data对象，为多个属性

c. 为每个属性添加访问器属性保镖

1). 保镖负责获取或修改数据属性的值

2). 保镖还能在数据属性的值被修改时发出通知

(2). 构建虚拟DOM树:

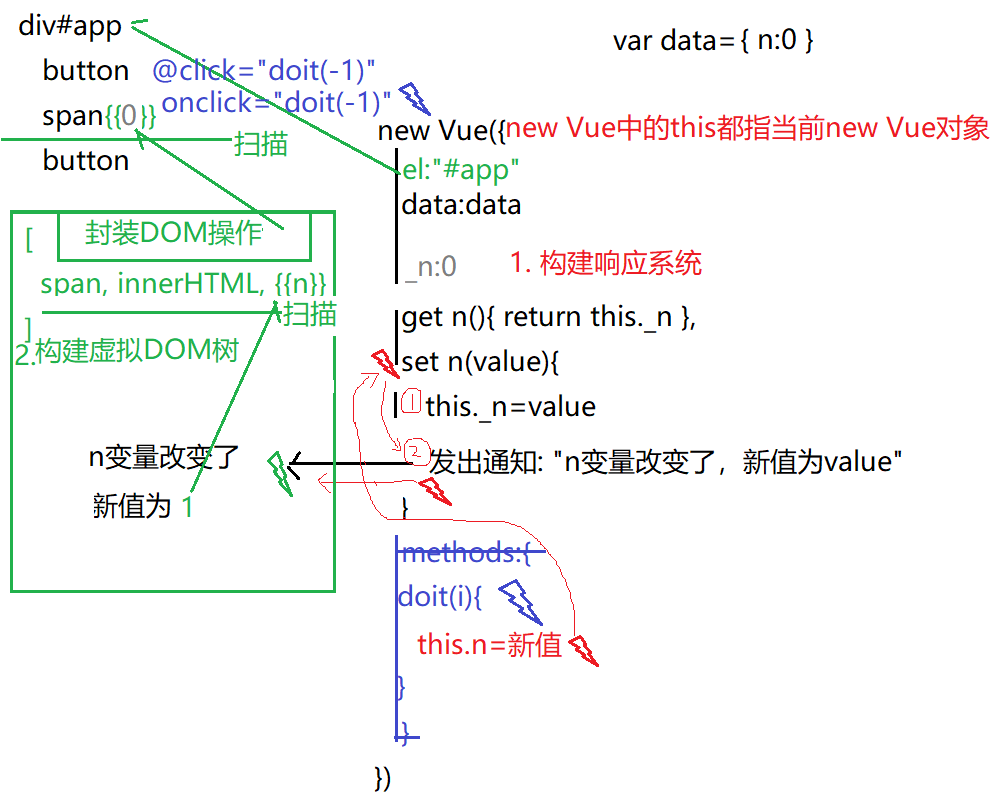
|  |
| --- |
| 虚拟DOM树: 只保存可能变化的元素的简化版DOM树 |

a. 根据new Vue()中el:后的选择器找到要监控的区域的父元素

b. 扫描父元素及其所有子内容  
 c. 每扫描到一个可能发生变化的元素(包含{{n}})，就保存到简化版虚拟DOM树中

d. 在首次加载页面时，会立刻去new Vue中查找是否有需要的变量的值。立刻将变量的值替换到真实DOM树的指定位置。

e. 今后在new Vue中任何位置通过this.变量方式修改了变量值，都会给虚拟DOM树发通知。然后，虚拟DOM树只扫描受影响的元素，找到受影响的元素，用提前封装好的DOM操作，更新真实DOM树



|  |
| --- |
| 虚拟DOM树的优点：  1. 小，只包含可能变化的元素，遍历极快！  2. 效率高，因为只更新受影响的元素。不受影响的元素保持不变！  3. 已经封装了DOM增删改查操作，避免大量重复编码 |

5. MVVM vs MVC

(1). MVC: Model View Controller

模型 界面 控制器

其中控制器的作用，也是将Model中的数据，填充到View界面中

(2). VM vs Controller

a. vm可自动将Model中的数据，通过虚拟DOM树更新到View中指定位置

b. Controller需要手动编写将Model中数据填充到View中的代码。

四. 绑定语法: 学名: Interpolation插值语法

1. 什么是: 在HTML中动态生成内容的一种特殊的语法

2. 何时: 只要HTML中有一处内容可能发生变化，就要用绑定语法

3. 如何: <元素>{{变量}}</元素>

4. 结果: 只要new Vue中变量变化，则此地内容自动跟随变化。

5. {{}}中和模板字符串的${}要求是一模一样的：

(1). 能放: 一切有返回值的js表达式

(2). 不能放: 程序结构: if else if else for while ...

6. {{}}只能绑定元素的内容，不能绑定元素的属性值和样式

7. 且{{}}中不能执行修改变量的语句

五. 指令(directive)

|  |
| --- |
| 什么是: 专门给HTML添加新功能的特殊属性符号  何时: 只要想给html元素添加特殊功能，就用不同的指令 |

包括：

1. 绑定元素的属性值时: v-bind

<元素 v-bind:属性="变量/表达式"

结果: 这个属性的值也可以通过变量和表达式动态生成

强调: 绑定属性值时，虽然""中写的是变量或表达式，也不用加{{}}，""就当做{{}}用了！

简写: 其实可以省略v-bind，只写 :属性="变量/表达式"

2. 控制显示隐藏:

(1). 一个元素，控制是否显示：v-show

<元素 v-show="判断条件">

结果：每次new Vue扫描带有v-show的元素时，都会动态判断条件。如果条件返回true，就正常显示。如果条件返回false，就自动翻译为display:none，隐藏该元素

(2). 两个元素，二选一显示: v-if v-else

<元素1 v-if="条件">

<元素2 v-else>

结果: 在扫描时，如果条件为true，就显示v-if的元素，如条件为false，就显示v-else的元素

强调: v-if和v-else两个元素之间禁止插入其他元素。两个元素必须紧邻

|  |
| --- |
| v-show vs v-if  v-show使用display:none隐藏元素 —— 效率高  v-if是通过删除元素方式隐藏元素 —— 效率低 |

(3). 多个元素，多选一显示: v-if v-else-if v-else

<元素1 v-if="条件1"> //如果条件1返回true，就只显示第一个元素

<元素2 v-else-if="条件2"> //否则如果条件2返回true，就只显示第二个元素

<元素3 v-else>//否则如果之前条件都不满足，就只显示最后一个元素

|  |
| --- |
| 将data中的数据绑定到页面显示:  1. 绑定元素内容: {{}}  2. 绑定属性值: :  3. 一个元素控制显示隐藏: v-show  4. 两个元素二选一显示: v-if v-else  5. 多个元素多选一显示: v-else-if |

作业：

1. 学子商城: index.html中一楼第2个和第3个商品

2. 详情页右上角四个位置，要绑定