**去哪网APP实战项目学习笔记（VUE+MOCK数据）**

**第一章 课程介绍**

课程内容如下：



课程技术点：



项目的课程安排：



学习前需要掌握的知识（.js的语法）：



学习课程后掌握的内容：

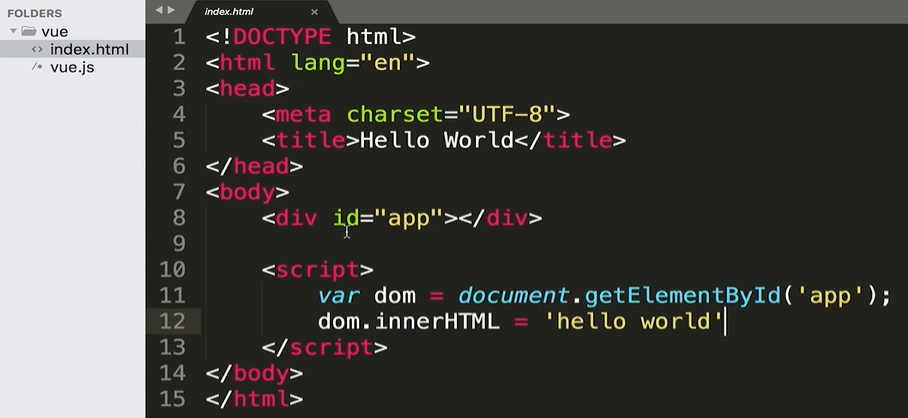


**第二章 Vue起步**

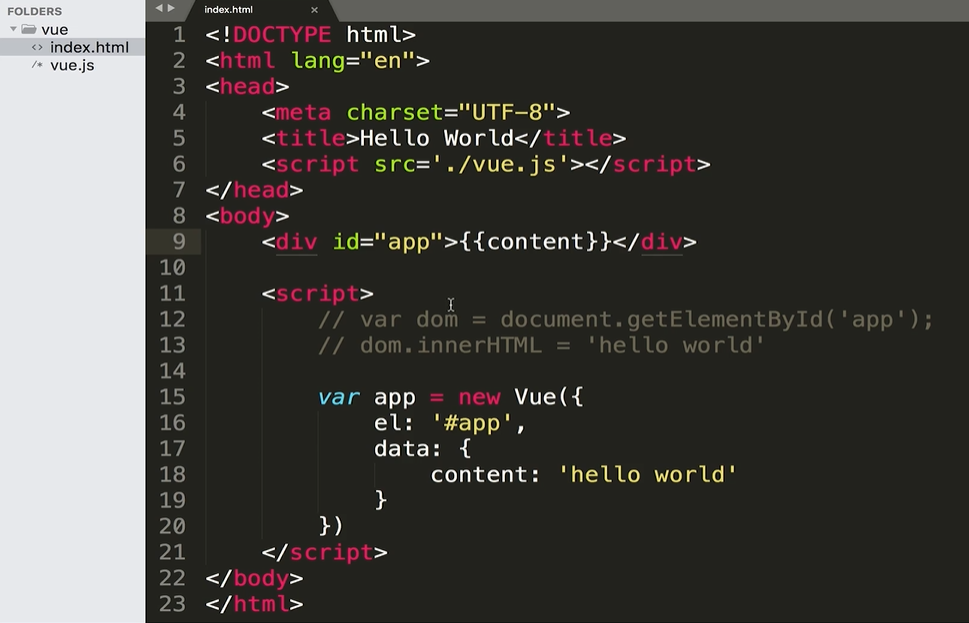
**Vue的学习方法：先学习培训视频，然后查看官网学习文档进行巩固**

原生js编写方式和vue编写方式的区别：

1、原生js编写方式：



2、Vue的编写方式：



创建一个vue实例，其中el表示实例管理的区域

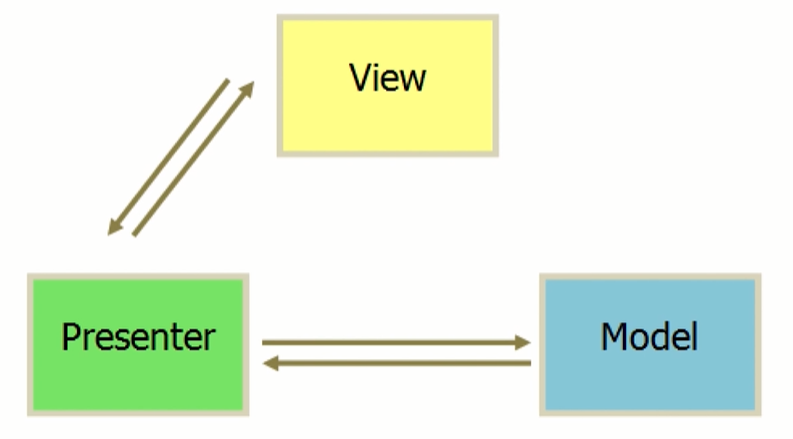
**Vue和原生js相比只关注数据，不关心网页dom**

V-model、v-for、v-on的vue命令使用方法：



传统前端的MVP设计模式

Model数据层、Presenter业务逻辑控制层、View视图层呈现结果



工作模式：

采用jquery编写todolist代码

M（模型层，代码范例暂时没有） V（视图，网页div显示） P（控制器script对逻辑进行处理，获取M层的数据，逻辑处理后，通过DOM操作改变视图）

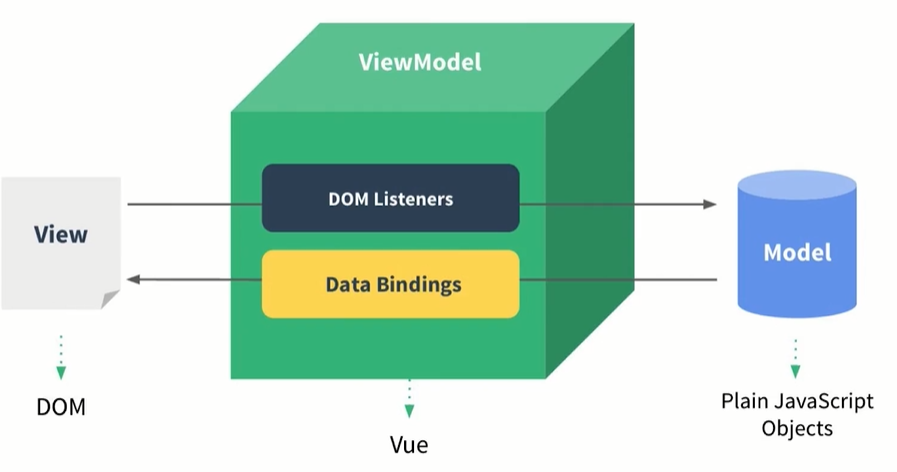
Model非常边缘化，Present er是核心层，主要是做DOM的操作



MVVM的前端设计模式

MVVM模式要是对M数据层进行操作，数据自动变化是VM层做的，VUE内置功能简化了DOM操作

主要是面向数据进行编程



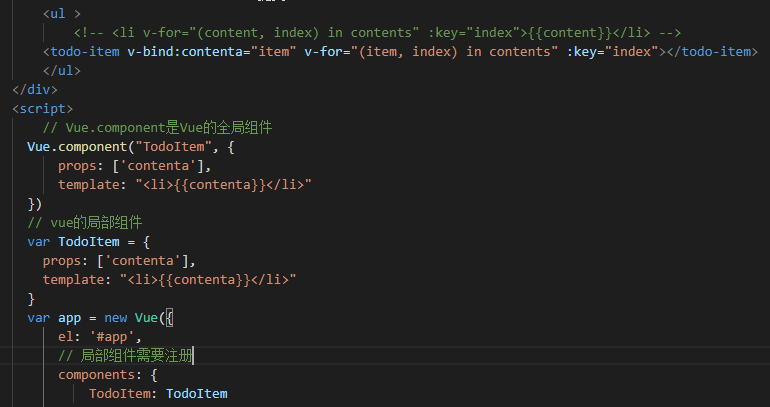
采用VUE的MVVM模式代码



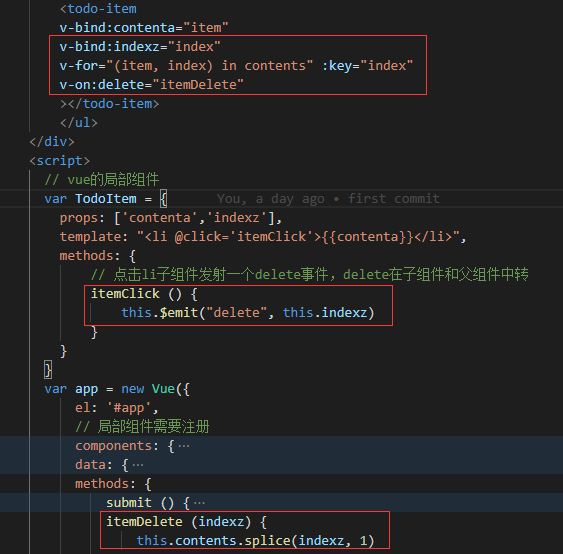
VUE的组件使用方法

分为全局组件和局部组件，包括v-bind的使用方法

**VUE中父组件向子组件传值**



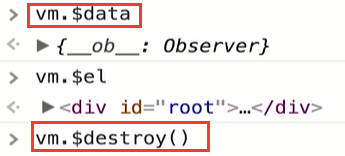
**VUE中子组件向父组件传值**



一个VUE项目是由多个VUE实例（VUE组件）组成的

VUE组件用$显示属性

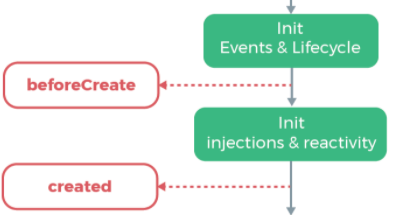
Vm.$destroy()将VUE组件销毁，销毁后相关数据绑定不会变化



**VUE实例的生命周期钩子（非常重要）**

1、初始化事件和生命周期相关的内容，执行完后会自动执行函数beforeCreate

2、处理注入和双向绑定相关的内容，执行完后会自动执行函数created





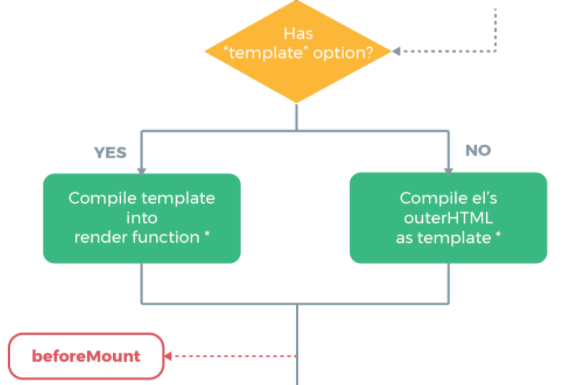
执行顺序如下所示：



判断是否有template模板

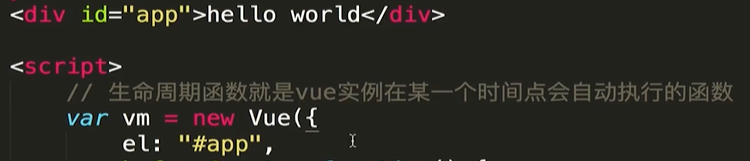
如果有模板，就会用模板来渲染页面

如果没有模板，会把el对应的html当做模板来渲染页面

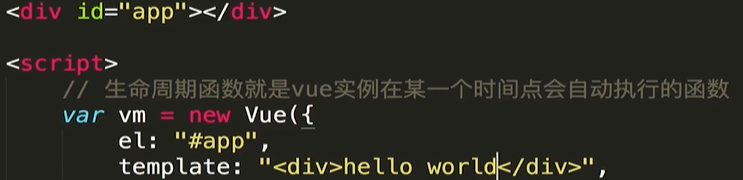


下面2种代码的显示方式相同：

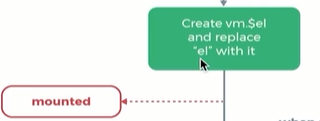
1、没有模板



2、采用模板方式

模板和页面相结合，即将加载到显示页面前的时间点自动执行beforeMount函数

模板和页面结合的元素加载到页面上，就是具体内容显示到页面上（页面有显示内容）后自动执行mounted函数



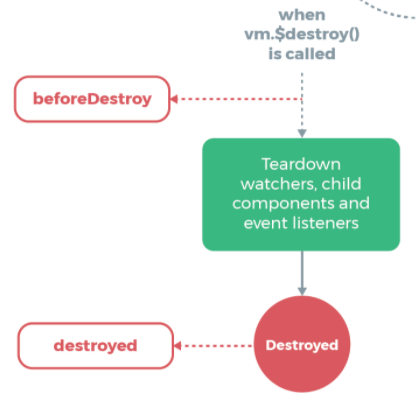
beforeMount页面还没有被渲染，mounted时候页面已经被渲染，可以看到页面内容



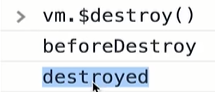
执行顺序和显示内容如下所示：



当VUE组件调用$destroy()方法销毁组件之前beforeDestory函数会被自动执行，当$destroy()方法执行后，VUE组件已被销毁，destoryed函数会被自动执行



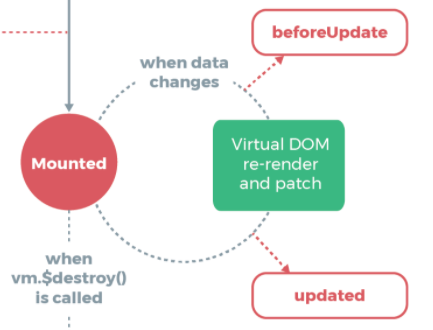
调用顺序如下所示：



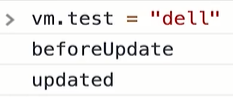
具体内容显示到页面上（页面有显示内容）后自动执行mounted函数，mouted函数执行后

当数据发生改变，页面渲染显示之前beforeUpdate函数会被执行

当数据发生改变，页面渲染显示完成之后updated函数会被执行

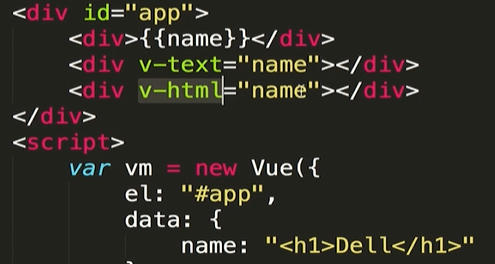


数据发生改变的时候函数执行的顺序如下所示：



模板语法

{{name}}为插值表达式，v-html解析为html的语法显示



页面显示效果如下所示：



**计算属性、方法和侦听器（非常重要）**

计算属性是内置缓存，当参与计算的值没有变化，就用上一次的缓存值，不会每次都重新计算（计算属性的性能更好）



采用方法也可以进行计算，但是没有计算属性的内置缓存机制，当参与计算的值没有变化时，每次都会重新计算



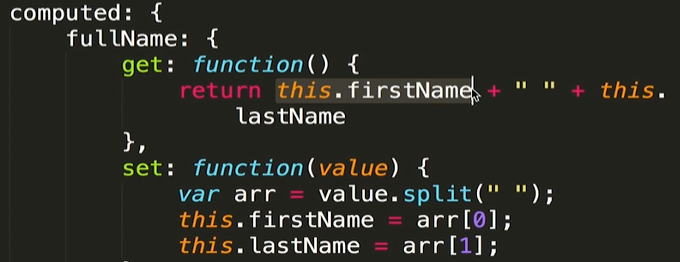
侦听器有缓存，当侦听的变量不发生变化的时候，就用上一次的缓存值，不会每次都重新计算（比计算属性语法复杂）



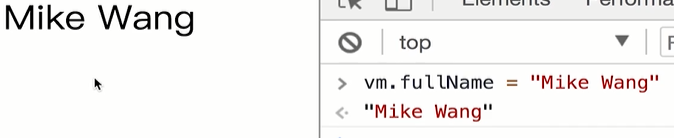
如果3种方式都可以实现，优先推荐计算属性

计算属性的get方法获取值

计算属性的set方法通过设置值来改变相关联的值，影响get方法的结果



fullName重新设置后通过set方法，影响了firstName和lastName值，同时firstName和lastName值变化影响了get方法里面的结果



如果没有设置set方法，那么fullName重新设置后，firstName和lastName值不会有变化，get方法里面的结果也不会有变化

