**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：

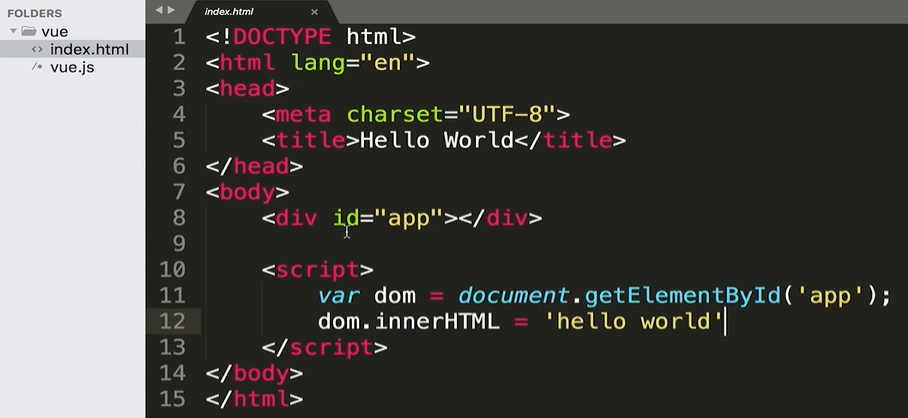


**第二章 Vue起步**

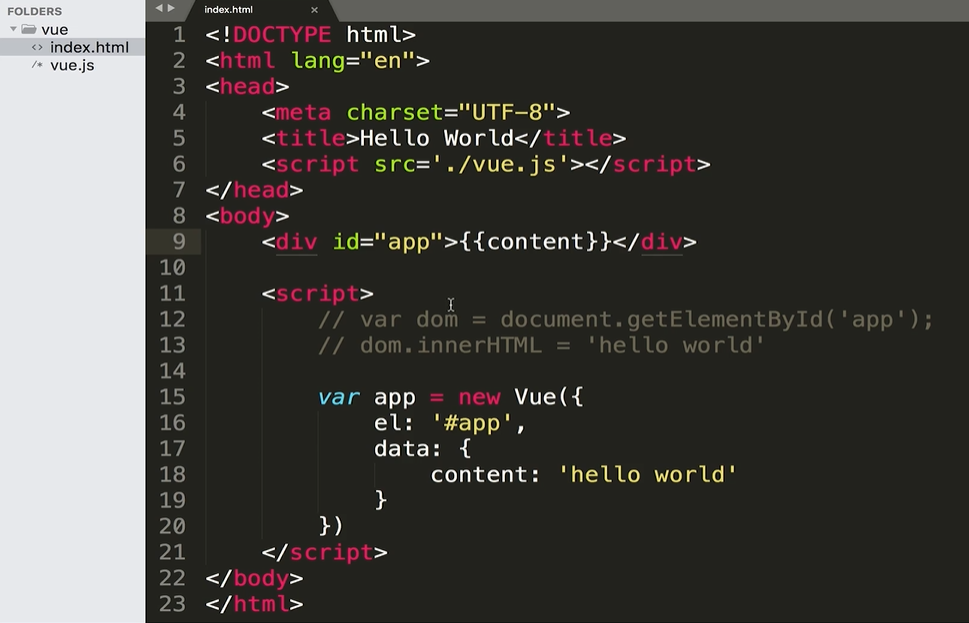
**Vue的学习方法：先学习培训视频，然后查看官网学习文档进行巩固**

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

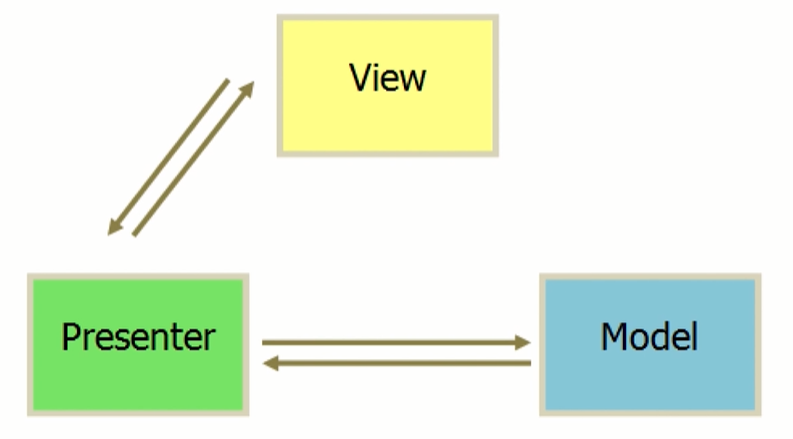
**Vue和原生js相比只关注数据，不关心网页dom**

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）

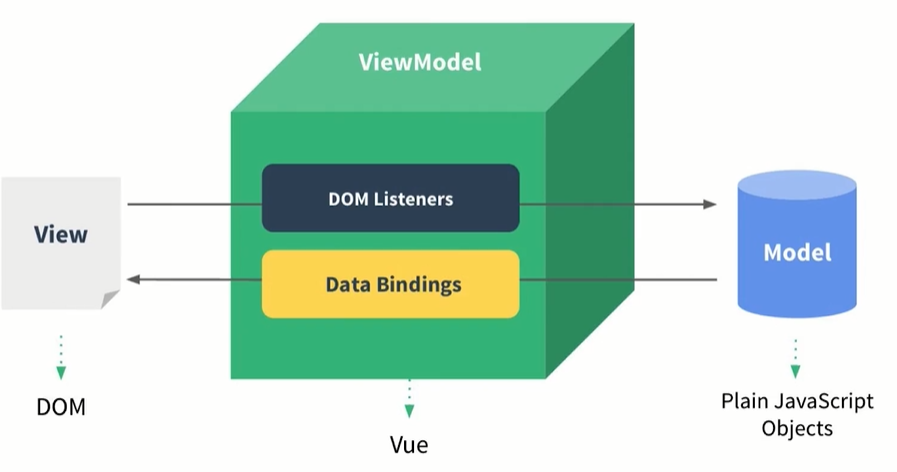
Model非常边缘化，Present er是核心层，主要是做DOM的操作



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



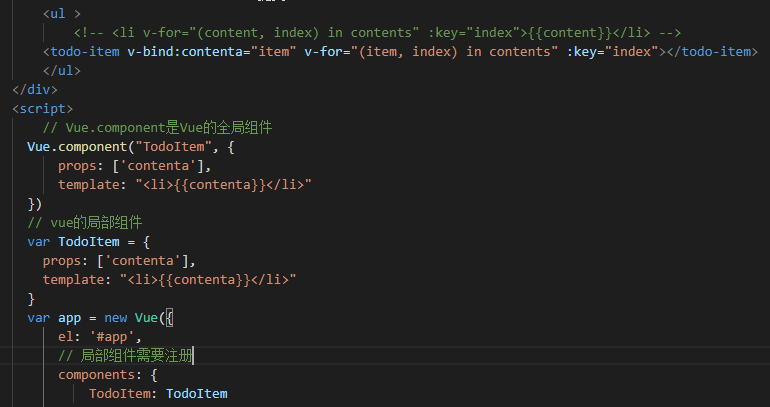
采用VUE的MVVM模式代码



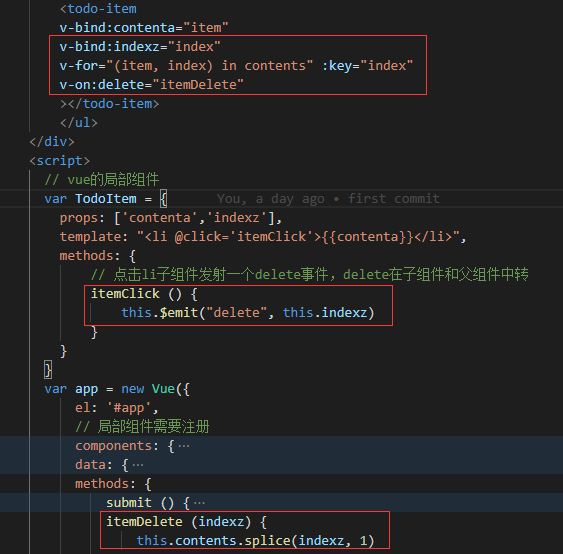
VUE的组件使用方法

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**



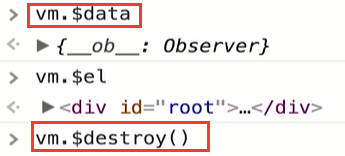
**VUE中子组件向父组件传值**



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

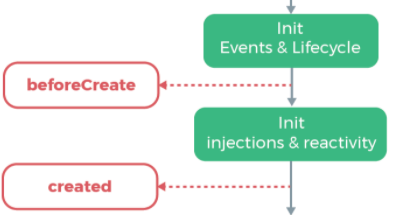
Vm.$destroy()将VUE组件销毁，销毁后相关数据绑定不会变化



**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





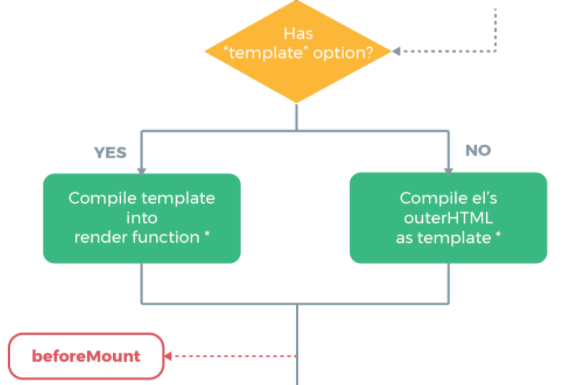
执行顺序如下所示：



判断是否有template模板

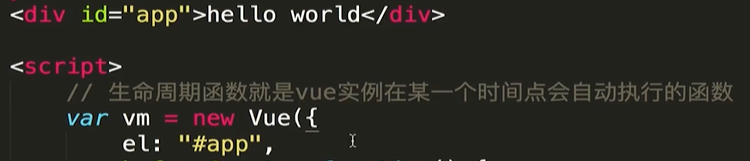
如果有模板，就会用模板来渲染页面

如果没有模板，会把el对应的html当做模板来渲染页面

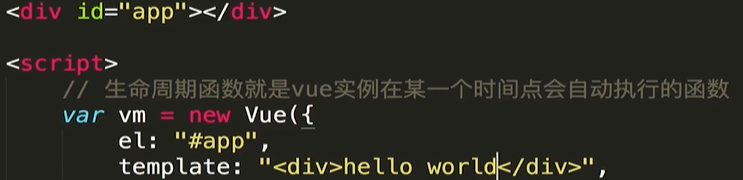


下面2种代码的显示方式相同：

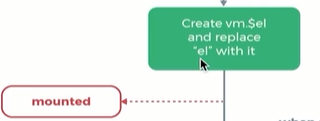
1、没有模板



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



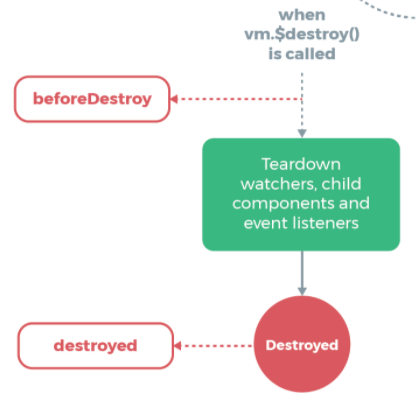
beforeMount页面还没有被渲染，mounted时候页面已经被渲染，可以看到页面内容



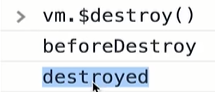
执行顺序和显示内容如下所示：



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



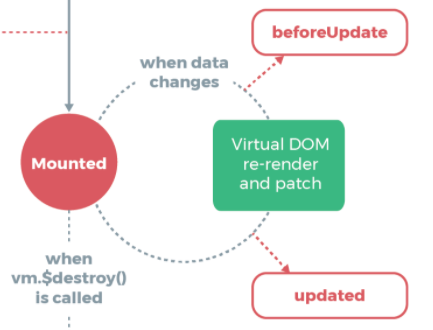
调用顺序如下所示：



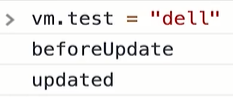
具体内容显示到页面上（页面有显示内容）后自动执行mounted函数，mouted函数执行后

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

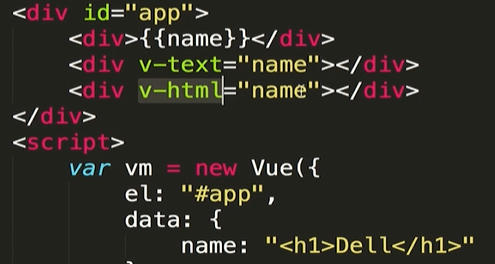


数据发生改变的时候函数执行的顺序如下所示：



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



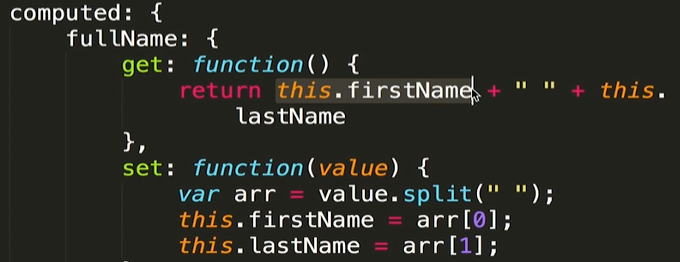
侦听器有缓存，当侦听的变量不发生变化的时候，就用上一次的缓存值，不会每次都重新计算（比计算属性语法复杂）



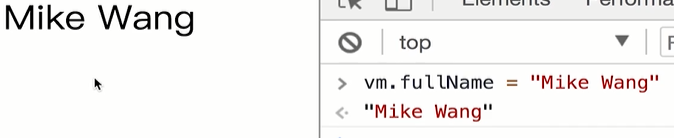
如果3种方式都可以实现，优先推荐计算属性

计算属性的get方法获取值

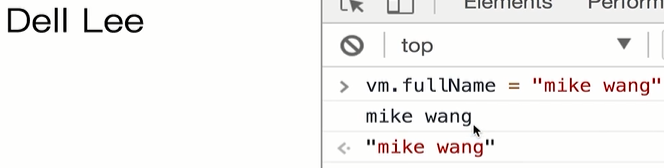
计算属性的set方法通过设置值来改变相关联的值，影响get方法的结果



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

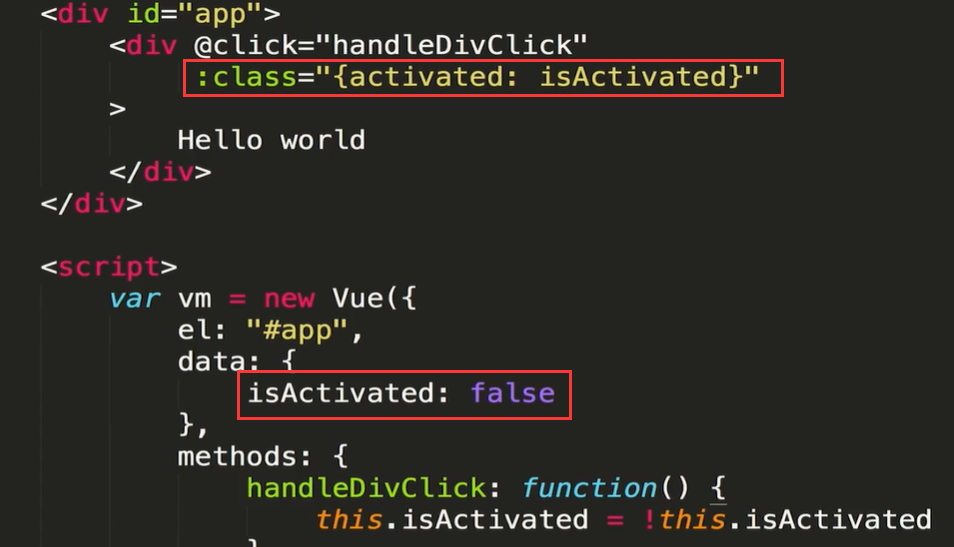


Vue中的class样式绑定

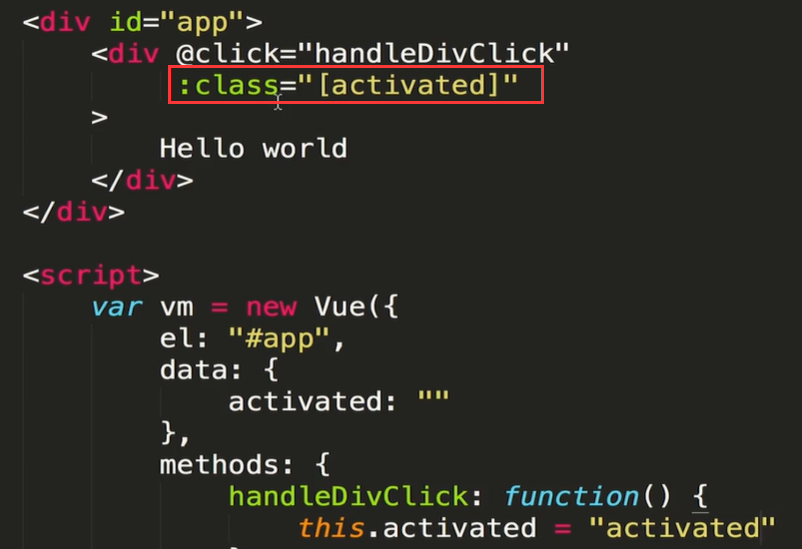
定义显示样式类



第一种采用布尔值判断



第二种采用直接赋值的方式



v-if如果是false则不存在于DOM上

v-show如果是false则存在于DOM上，不过样式为none没有显示

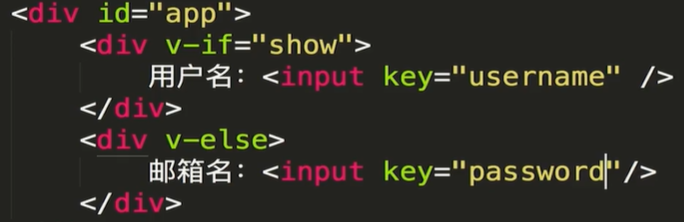
如果需要非常频繁地切换，则使用 v-show 较好；如果在运行时条件很少改变，则使用 v-if 较好



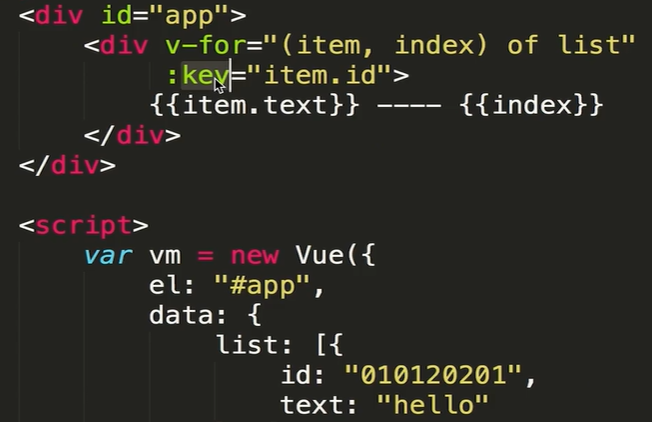


v-if的复杂应用

vue会自动服用页面上的input，在用户名和邮箱名之间切换，input内容没有进行切换，需要给input加key值进行区别



数组显示key值采用数据的唯一id不适用系统自动生成索引值



直接给数组下标赋值不会刷新页面的显示值，需要适用数组操作方法



数组操作有几个方法：

Push向数组里增加一条数据

Pop将数组最后一条数据删除

Shift将数组第一条数据删除

Unshift向数组第一条数据加内容

Splice做数组的截取

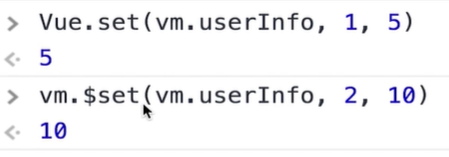
Sort将数组进行排序

Reverse将数组进行取反

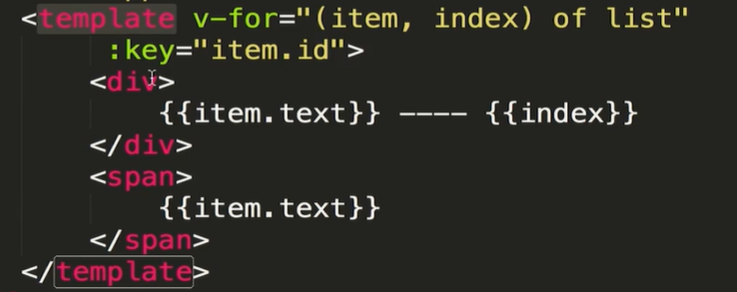
将数组重新指向另外一个数组，页面数据显示会发生变化



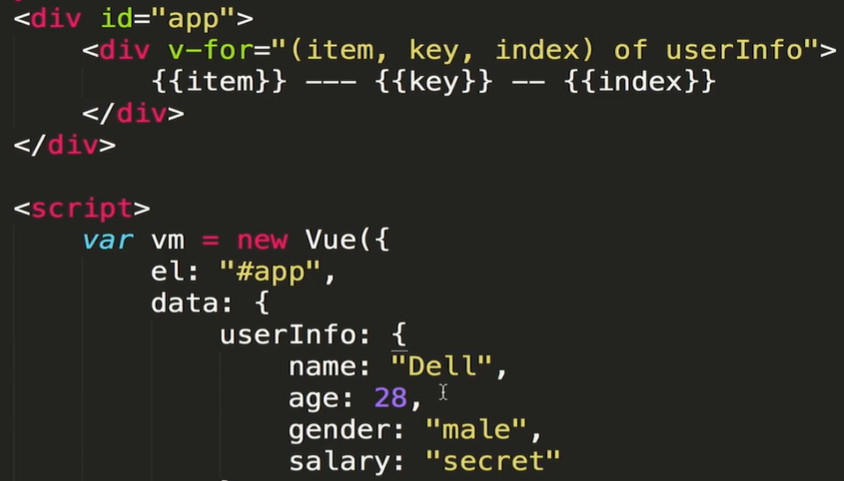
可以使用vue的set方法，如下所示：



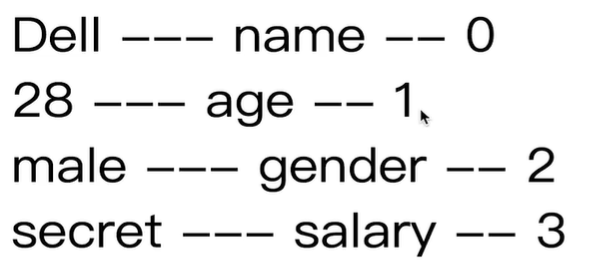
Template占位符，不会在页面显示，可以用来替代div



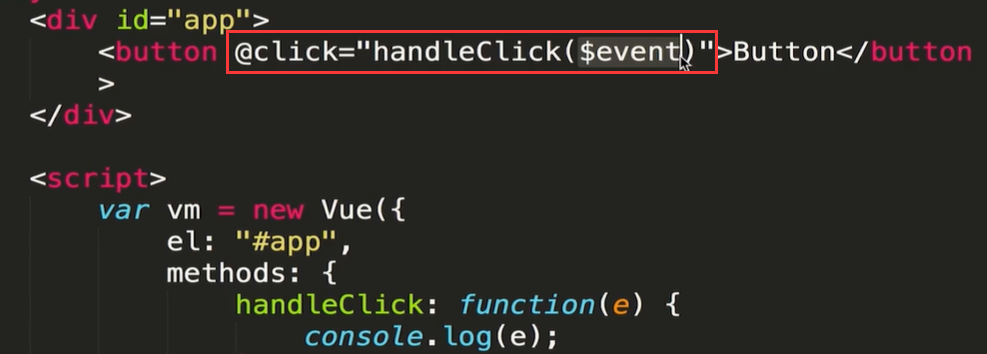
可以对对象进行循环：



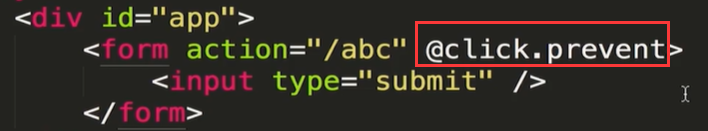
显示的循环结果：



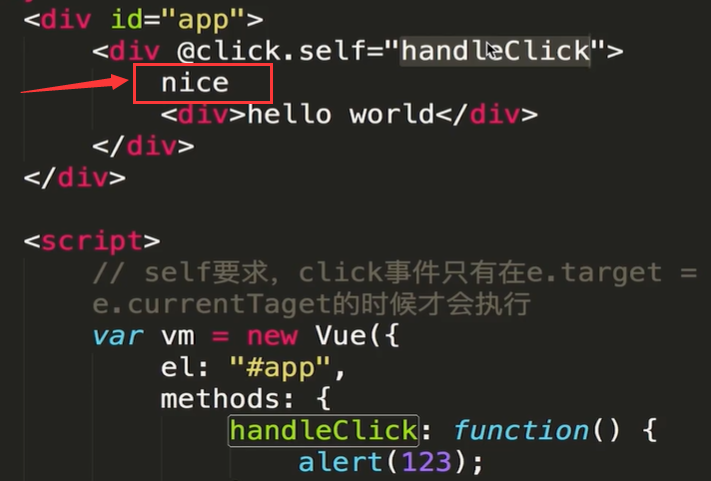
传事件对象，可以打印出来



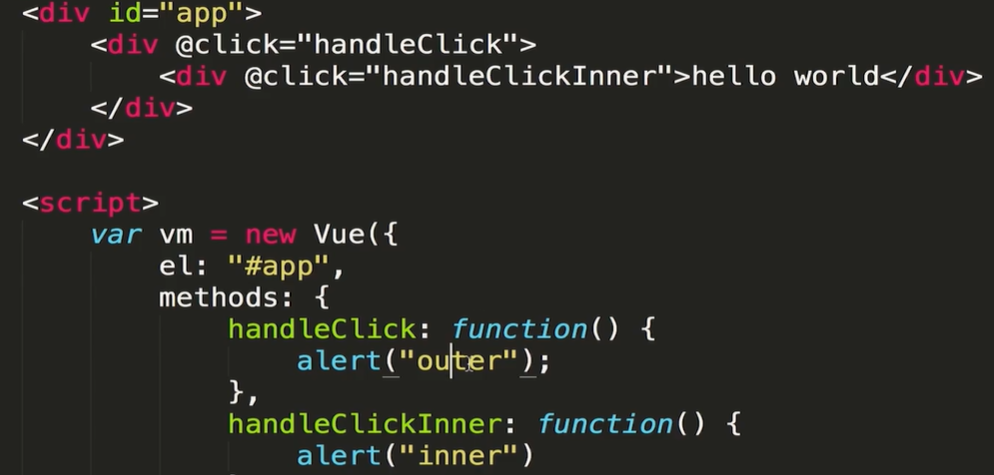
阻止submit后续的网页跳转行为



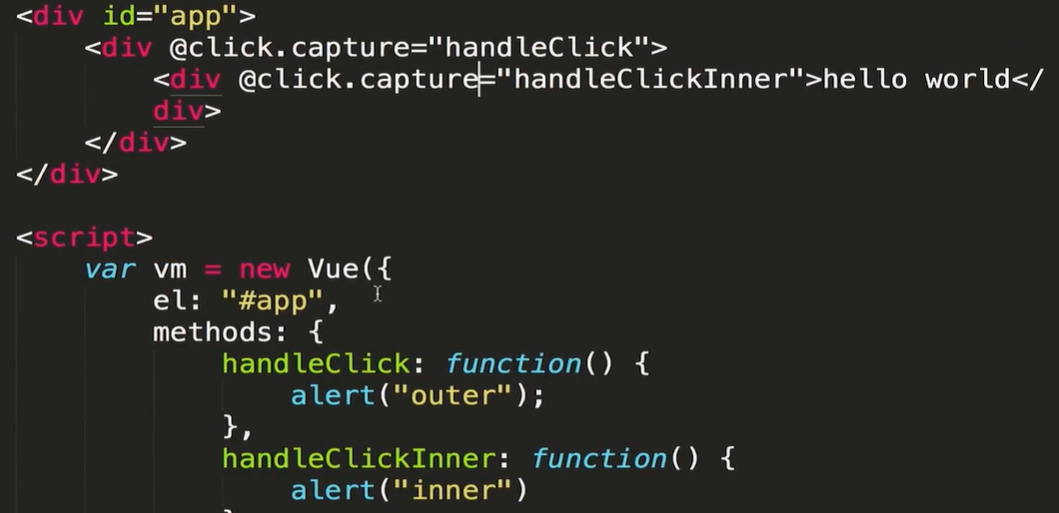
只有点击nice才会执行click方法



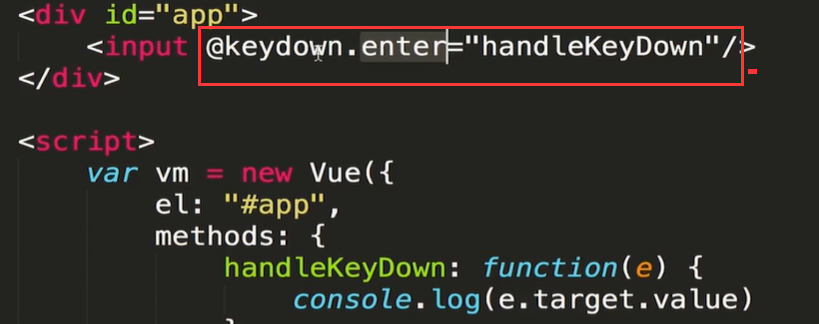
事件冒泡执行顺序，先执行inner，再显示outer



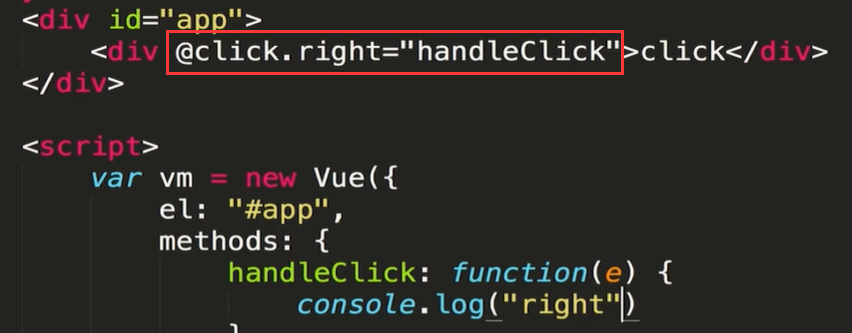
Capture执行捕获规则先执行outer，再执行inner



按钮修饰符，只有点回车键，数据内容才会显示

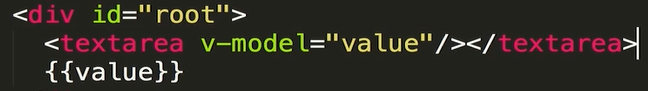


鼠标右键点击的时候才会执行



v-model的双向绑定：

Textarea和input标签一样，双向绑定



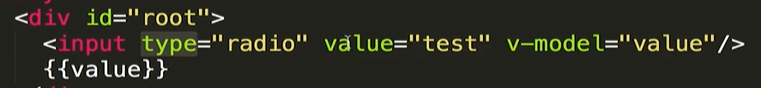


Checkbox复选框的双向绑定是选中后显示为true，不选中显示为false





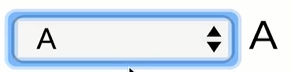
Radio单选框的双向绑定是选中后显示value值





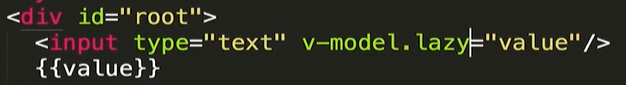
Select的双向绑定，如果option有value则显示的是value值，如果没有则显示选中的值



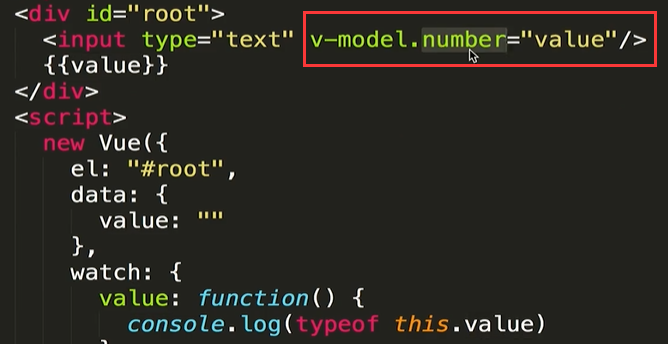




Lazy修饰符，当鼠标焦点离开input时候，再绑定数据{{value}}显示变化



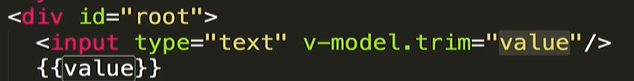
Number修饰符，将input的数字内容转换为number类型







Trim修饰符，去掉首尾空格的修饰符



第3章小结

Vue是由组件组成的，每个组件都是一个vue的实例

模板的基础语法：v-text、v-html、插值表达式{{value}}

计算属性computed：比methods方法有缓存，比watch语法简洁

样式绑定：class样式绑定可以通过对象和数组的方式来进行绑定（复习）

条件渲染：v-if、v-elseif、v-else三个指令必须连续写，template模板占位符，视图不显示

列表渲染：v-for可以循环数组和对象（复习）

事件处理：v-on可以绑定click、input等事件，还有常见事件行为（复习）

表单绑定：v-model可以绑定input、textarea、select、单选复选框（复习）