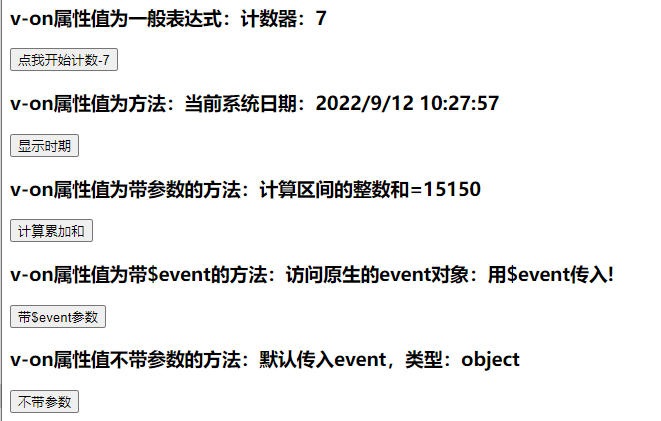
# 事件处理

目的：理解vue的时间处理机制，掌握绑定方法，熟练使用此技术解决业务需求。

## 练习一：综合练习vue事件处理基础操作

页面需求如下：







效果动态呈现：

**案例演示**

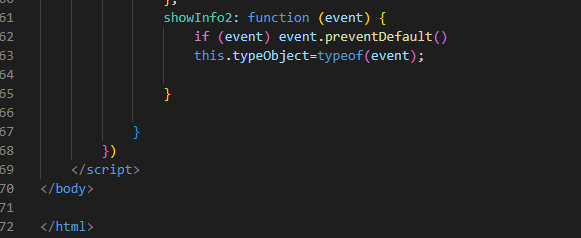
参考代码：

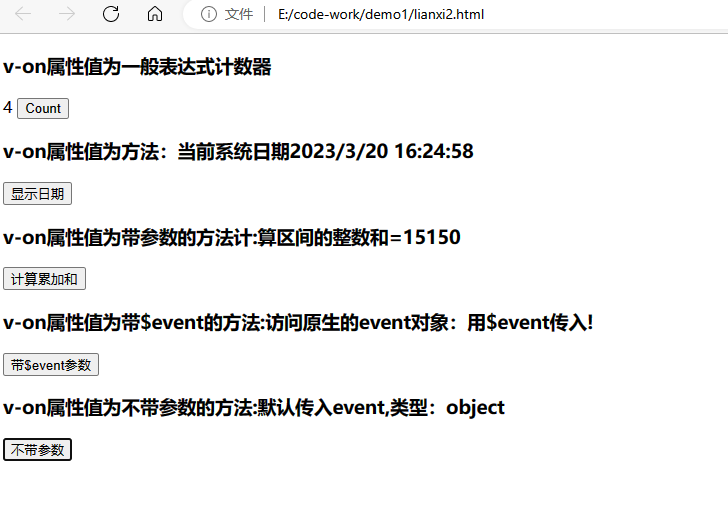






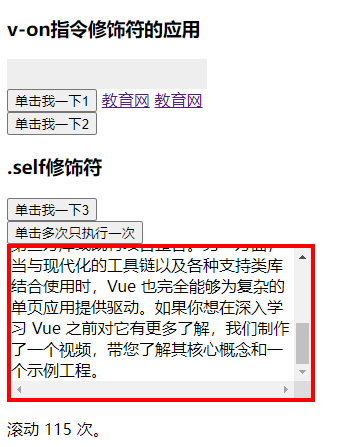






## 练习二：事件处理的修饰符的应用

页面结构;



思考，以上页面效果用什么去装载？该怎么用html+css实现，你可以将页面设置得更漂亮些！

页面结构参考代码：



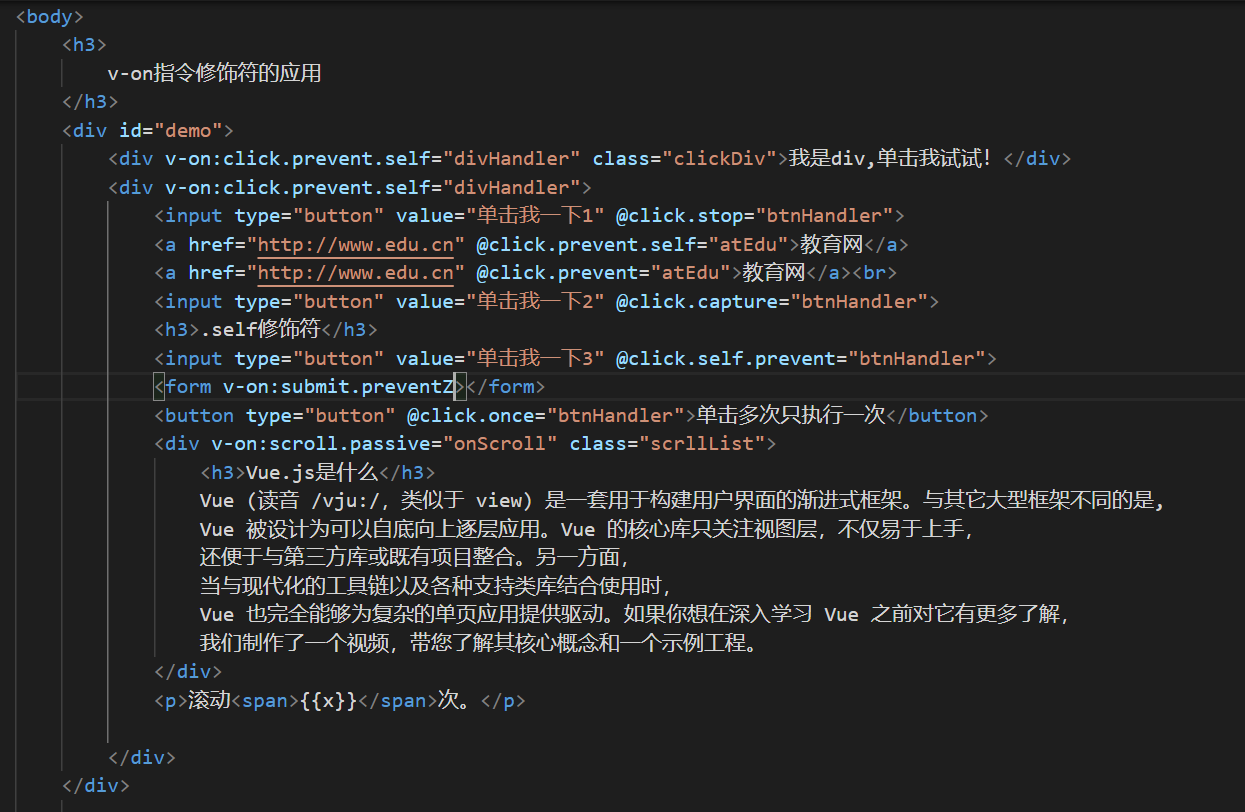
案例文本：

 Vue (读音 /vju:/，类似于 view) 是一套用于构建用户界面的渐进式框架。与其它大型框架不同的是,Vue 被设计为可以自底向上逐层应用。Vue 的核心库只关注视图层，不仅易于上手，还便于与第三方库或既有项目整合。另一方面，当与现代化的工具链以及各种支持类库结合使用时，Vue 也完全能够为复杂的单页应用提供驱动。如果你想在深入学习 Vue 之前对它有更多了解，我们制作了一个视频，带您了解其核心概念和一个示例工程。

脚本参考代码：



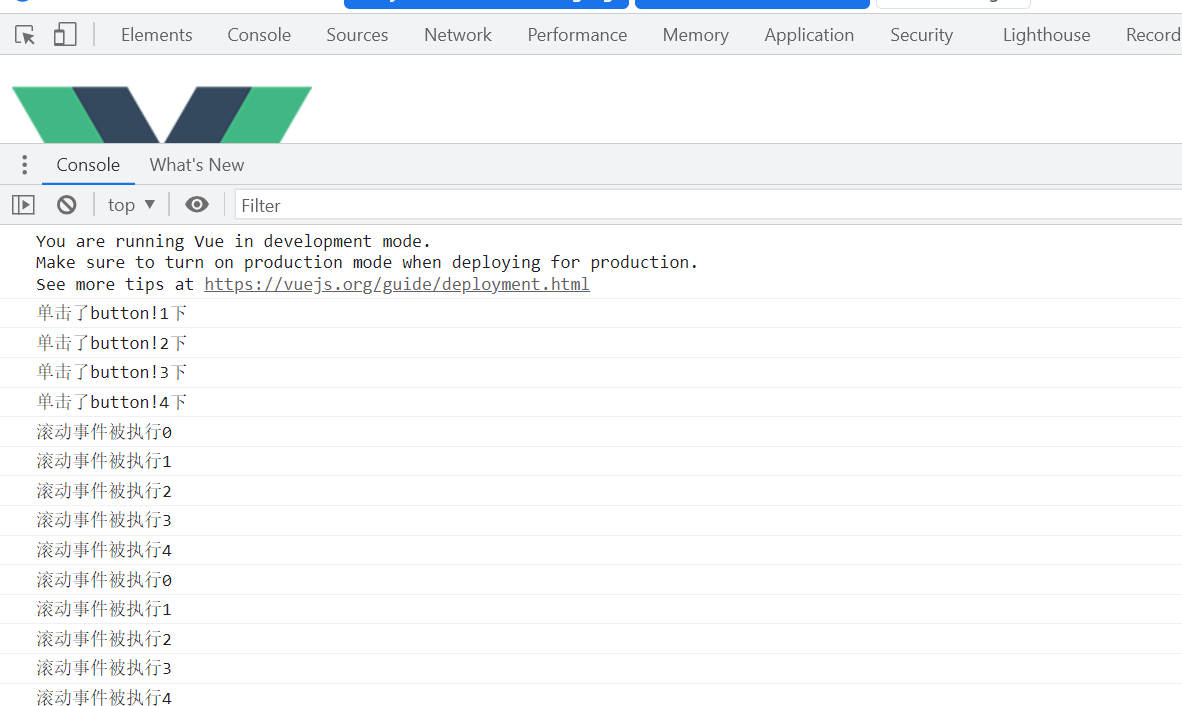
代码实现：

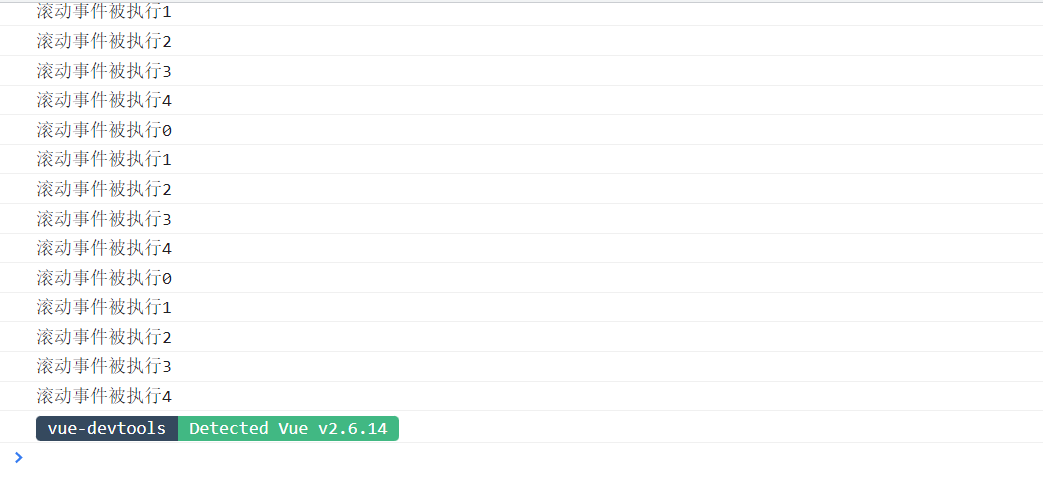




呈现效果：

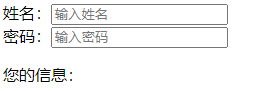






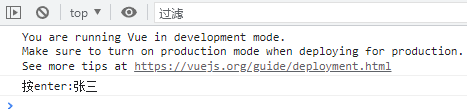
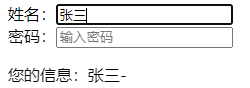
## 练习三：按键修饰符

完成课堂练习后，实现下例。

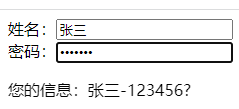


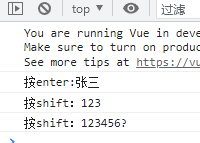
要求：

1.姓名文本输入框，通过v-on指令绑定keyup事件，并且设置，需按下回车键后，调用事件方法inputName，在控制台输出相关信息，并将输入的姓名和密码回显到“您的信息”后边；

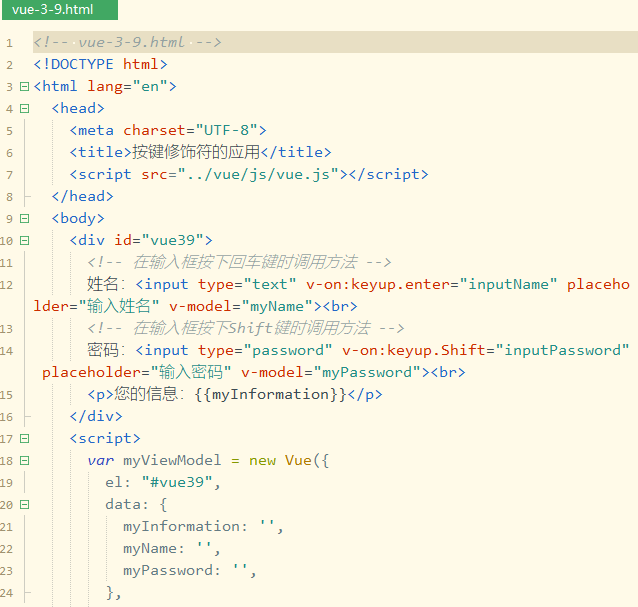


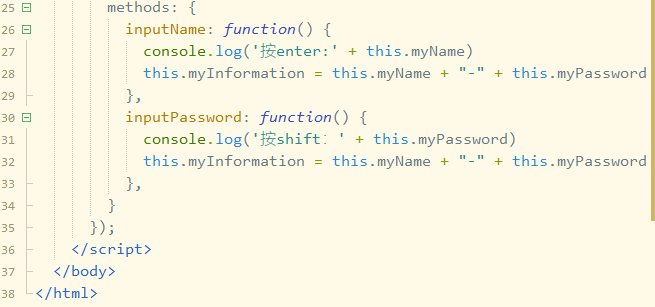
2.为密码框输入框，通过v-on指令绑定keyup事件，并且设置为，当用户在密码框中输入带有shift+按键时，调用inputPassword方法，在控制台输出相关信息，并将输入的姓名和密码回显到“您的信息”后边。



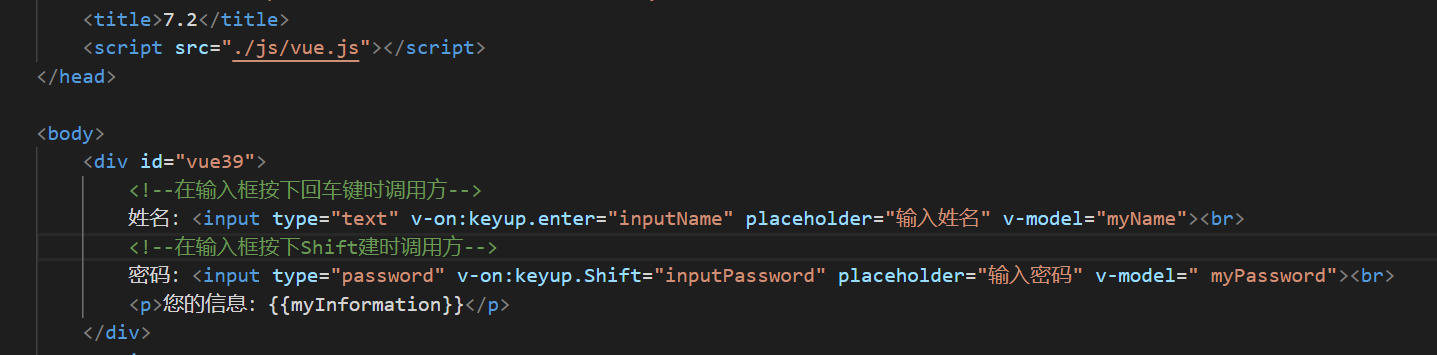


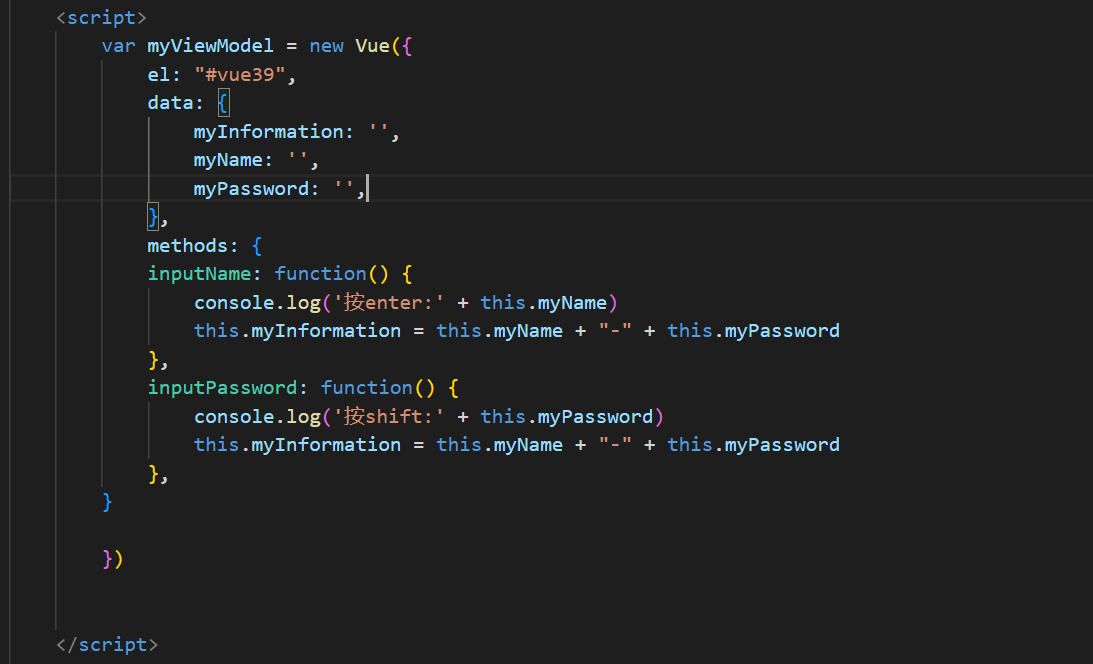
参考代码：





代码实现：





效果呈现：

