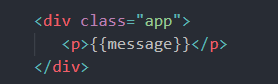
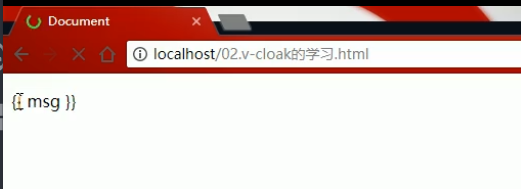
# v-cloak指令

在网速非常慢的情况下，页面的渲染不及时，p标签中的 {{ message }}会直接被显示在网页中，当请求完了之后。数据才正常显示



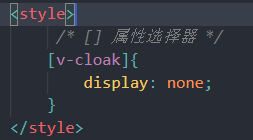


解决办法：

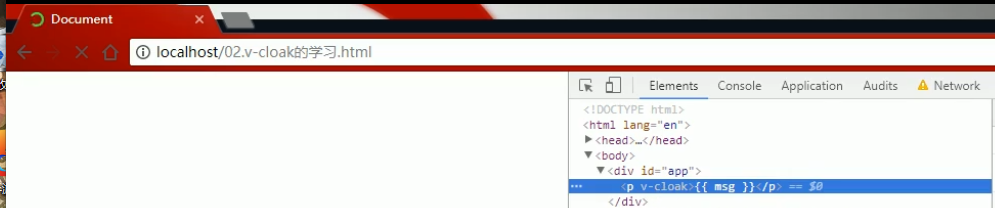
在p标签中添加 v-cloak属性



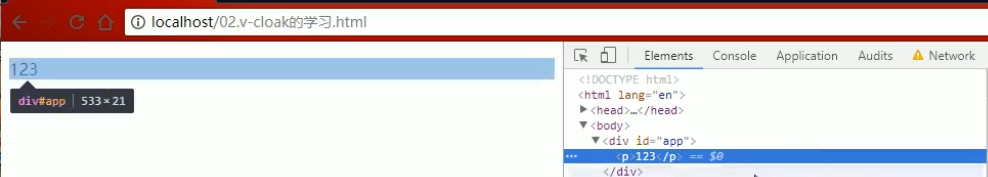
并为这个属性添加一个隐藏样式



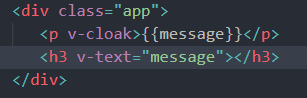
添加 v-cloak属性后，网络在请求时 p标签里面的 {{ message }} 会被隐藏



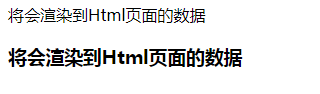
请求完成后 v-cloak 被移除，数据显示



# v-text指令

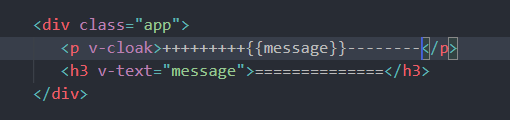


效果：

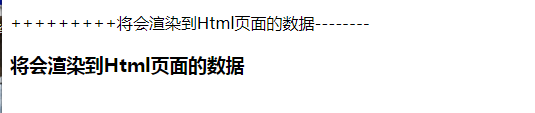


v-text 作用与插值表达式 {{}} 一样，但是不会存在上面存在的渲染页面不及时而直接将插值表达式显示出来的问题，但是 **v-text会覆盖原本元素中的内容，而插值表达式只会替换自己的这个占位符，不会把整个元素内容清空**

实例：



运行结果：



# v-html指令

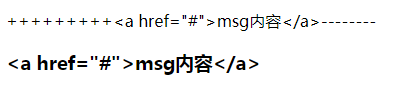
在网页中我们不止会往页面中添加文本数据，也会往页面添加html数据

实例：

在data中添加 msg 用插值表达式和 v-text的方式显示



效果：

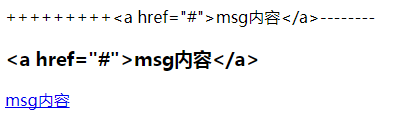


由此可见 **插值表达式和v-text 都无法显示html数据**

用 v-html 渲染数据



效果:



**注意：v-html指令和v-text指令一样会覆盖元素中原本的内容**

# v-bind

**v-bind是 Vue 中，提供的用于绑定属性的指令，被其绑定的属性的“”内的内容会被浏览器认为是js的代码去解析执行**

实例：

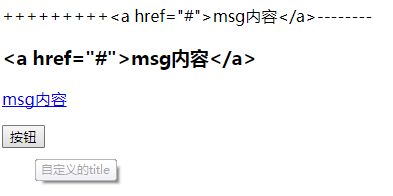
Html代码



Js代码



效果：



如果想在title中追加信息有如下两种方法

在html代码中添加 （将mytitle当作一个变量在后面追加内容）



在js代码中添加



效果：



**v-bind的简写**

