# 

# 【我们的发现】

1. 调用了CSS beforel选择器



所以这个模态框的背景本身就是一个before的content；

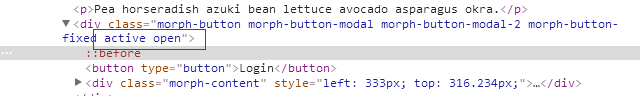
1. 模态背景切换的本质是？

两个状态，或者说你准备两个样式

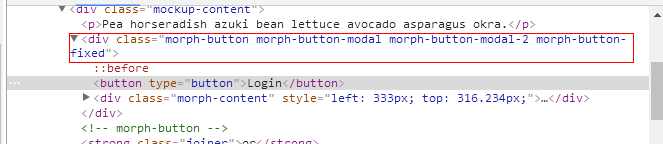
一个是【还没点的样式】【已经点的样式】

通过JS添加一个类名“active”来进行切换两种状态

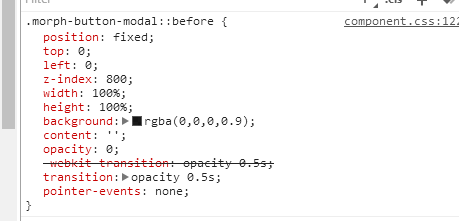
其实也不是两种状态，通过重写before的样式来呈现两种状态



首先，没点之前是这样的



还没有添加这个，但是我们看到已经有个before了，样式是

\

position: fixed;位置是绝对定位，对象是浏览器的窗口

top: 0;从0开始

left: 0;从0开始

z-index: 800;还没显示就已经是最前态800层，只不过透明为0跟鼠标事件失效，它等价不存在

width: 100%;填满整个窗口

height: 100%;填满整个窗口

background: rgba(0,0,0,0.5);最终形态的颜色是黑色，随时待命(通过不透明度来控制所以a它不用变)

content: '';before空白内容‘’使它像个盒子一样可以任意效果

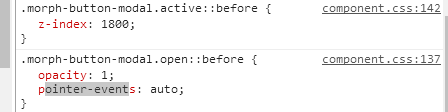
opacity: 0;所以你看不到

-webkit-transition: opacity 0.5s;待命动作是，透明度0到1的转化（就是整个出现过程）在0.5秒内完成

transition: opacity 0.5s;半秒内改变属性

pointer-events: none; 鼠标识别不了

点之后，1.叠加了两个类”active”和”open”



Z-index:1800；激活后，它就是1800层了

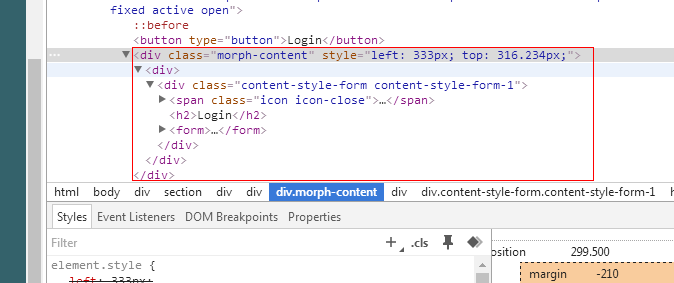
Opacity:1 不透明度为1，就是出现了

Poiner-events:auto;鼠标识别恢复正常，所以我们可以审核元素，不信你去掉试试



1. 但是组件框是怎么弹出来的呢？

它的位置在按钮之后



它有两个状态

第一个是没按之前的待命状态：

三个对象：变形框、触发按钮、组件

.morph-button-fixed变形框

button触发按钮

morph-content组件

之前：

position: fixed; 绝对定位，相对于窗口

z-index: 900; 900层（比模态背景高100）

opacity: 0;又是你看不到的状态待命

width: 300px; 由于是【对象连贯变形】所以跟原型对象一致的高宽

height: 80px;

第一步：