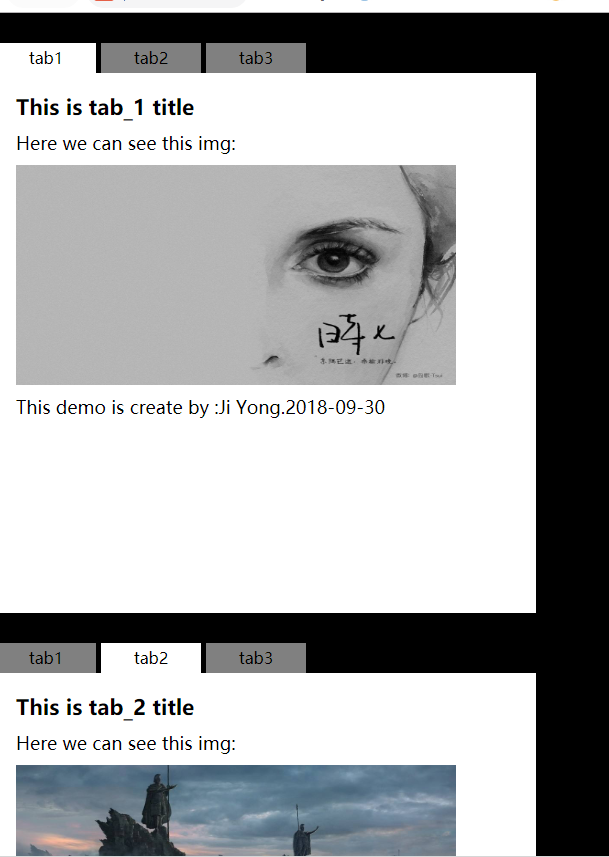
**JQ TABS 组件文档说明**

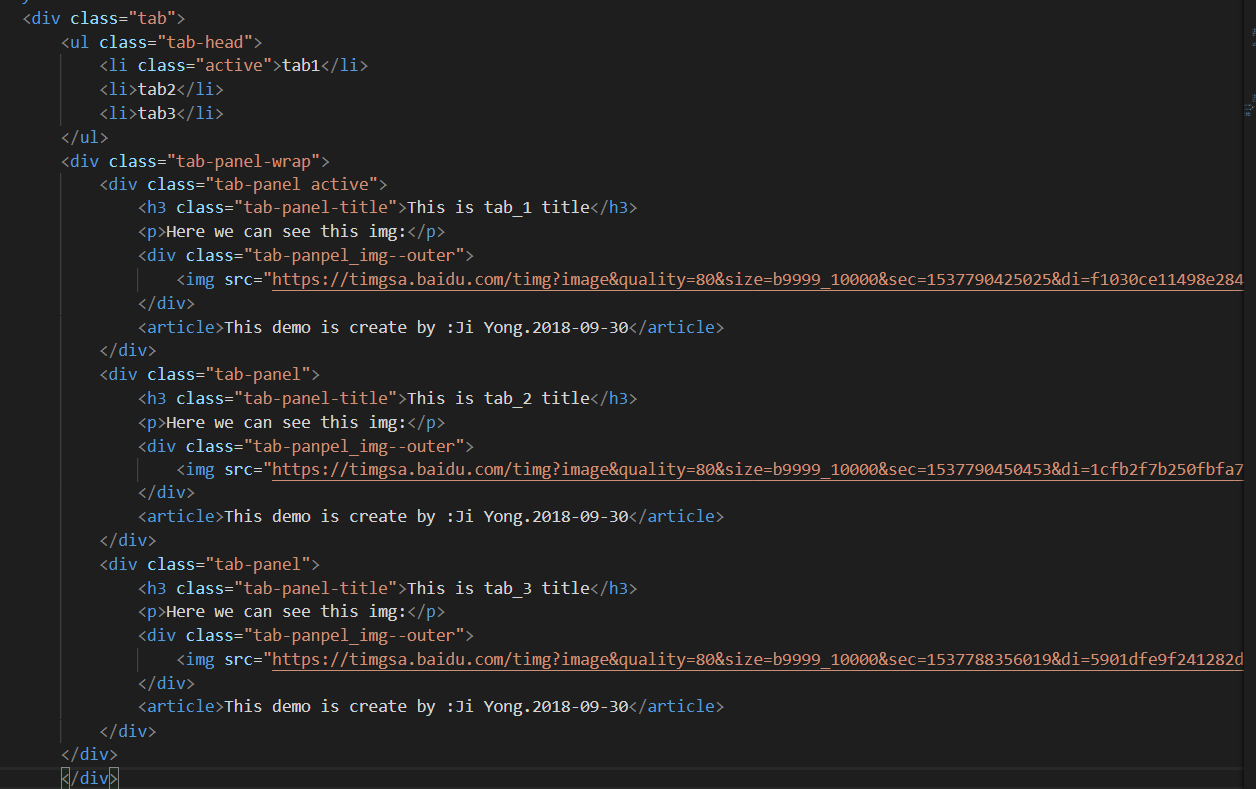
概述：JQ TABS DEMO主要使用对象级别插件开发，传递jQuery实参，立即调用执行，内部运用原型链上的方法，保证了独立性，从而实现Tabs组件开发。

运行效果图如下：

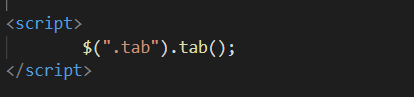


代码解释如下：

HTML

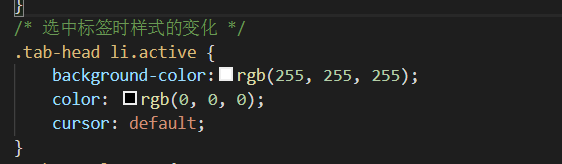


一个正常的Tab组件HTML结构，顶部UL标签底部div面板



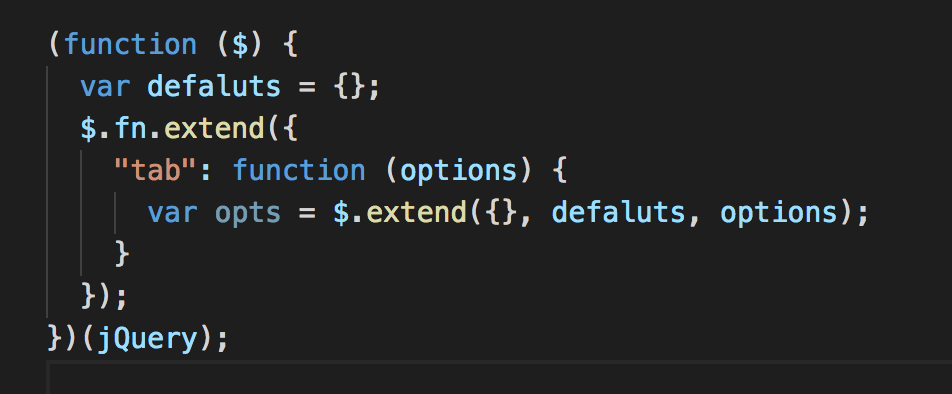
组件的调用方法，该方法写在scripts.js里。

CSS

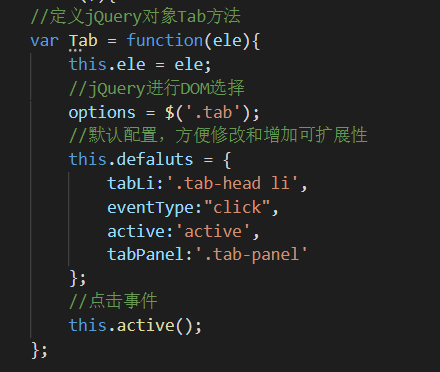


顶部标签被选中时的样式，通过JQ方法添加和删除该类产生效果。

JS



上图为JQ 组件写法，本次也是沿着这个结构进行Tabs组件编写。

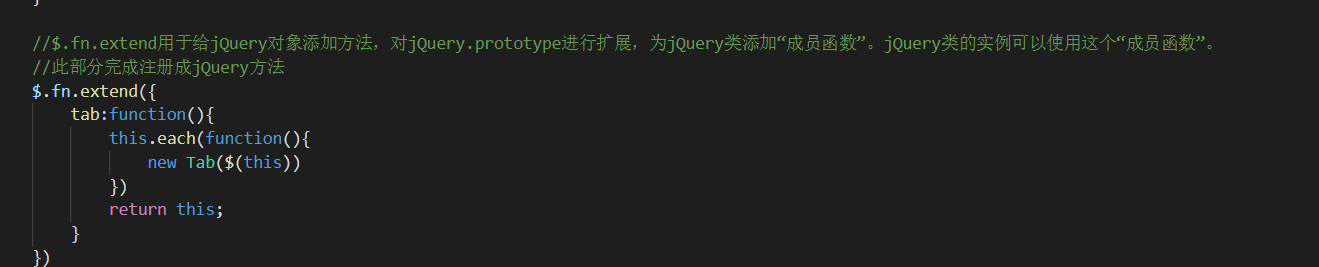


首先定义了JQ对象Tab组件方法，传入ele参数（用于后续监听事件）。然后用JQ DOM类选择器选中tab组件，将其变成JQ对象，从而可以使用JQ方法。再设置默认配置，将类名与方法存储在defaluts配置中，此举是为了方便下面原型链上方法的编写，以及日后若需要修改组件时只需要修改此部分即可，无需大量修改。



这一部分定义了JQ Tab组件原型链上的active方法，用于进行类的切换操作。将ele参数设置为this.ele，并且用save保存当前this的值，主要是因为添加的事件处理程序在其依附的元素的作用域中运行。jQuery中on()方法this指向会产生变换，所以先把this的执行环境保存起来。然后再让options继承this.defaluts的属性和方法，这也是为什么在一开始将defaluts配置写好的原因，这样子可以节省代码量。然后再给顶部li标签设置监听事件，匹配到li用addClass方法添加类，这里可以直接用options.active代替用选择器的方法写类名。因为在一开始的时候将li激活的类名保存在this.defaluts的active中了。然后再用siblings()和removeClass把兄弟元素的激活类名去掉。产生了选中的标签突出显示的效果。

再设置index值把当前JQ this.index保存下来。匹配对应的面板。同样的添加show()显示以及把兄弟元素hide()方法隐藏。这里.save的作用就体现了出来，提前保存了save中this的值，这样子this便不会指向jq on()中的对象，从而能根据li的index进行切换。



最后完成JQ方法注册。$.fn.extend用于给jQuery对象添加方法，对jQuery.prototype进行扩展，为jQuery类添加“成员函数”。jQuery类的实例可以使用这个“成员函数”。因此在html中只需要调用tab()方法便可以实现组件化。