# jquery.placeholder.1.3.min.js

一个jquery的占位符插件

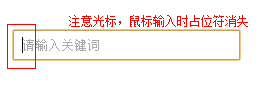
整搜索框，需要默认占位符为“请输入关键词”，获取焦点时，占位符消失或不可用（不影响正常输入），丢失焦点后，若用户无内容输入，占位符继续出现，继续占位。这种代码我想前端们已经很容易就写出来了吧，现在HTML5原生就有这个“placeholder”属性，效果与上边案例描述的一样（各支持的浏览器内部表现可能不一致，但是作用是一致的），那么这一属性该怎么优雅降级到支持所有现代浏览器呢？答案还是脚本即JavaScript。

上述案例的一个图例：

普通状态：

[](http://0.xiaosong.me/up/OOXX-4f6f0d319ea28-286-94.png)

获取焦点但未输入：

[](http://3.xiaosong.me/up/OOXX-4f6f0d31d25a7-265-91.png)

输入后：

[](http://0.xiaosong.me/up/OOXX-4f6f0d31cb7a6-298-83.png)

注：获取焦点但未输入时的状态各浏览器表现有差异，但不影响功能。

下面是我自写的一个jQuery插件，我博客就在用，代码如下：

注：获取焦点但未输入时的状态各浏览器表现有差异，但不影响功能。

下面是我自写的一个jQuery插件，我博客就在用，代码如下：

;(function ($) {

$.fn.extend({

placeholder : function () {

if ("placeholder" in document.createElement("input")) {

return this //如果原生支持placeholder属性，则返回对象本身

} else {

return this.each(function () {

var \_this = $(this);

\_this.val(\_this.attr("placeholder")).focus(function () {

if (\_this.val() === \_this.attr("placeholder")) {

\_this.val("")

}

}).blur(function () {

if (\_this.val().length === 0) {

\_this.val(\_this.attr("placeholder"))

}

})

})

}

}

})

})(jQuery);

你可以通过脚本判断浏览器是否原生支持placeholder属性来动态加载这个脚本，代码如下：

if(!"placeholder" in document.createElement("input")){

$.getScript("jquery.placeholder.js",function(){

$("#keyword").placeholder();//让id=keyword的元素支持placeholder，换成你自己的选择器

})

}

这样，高级浏览器就能调用原生方法，而不支持“placeholder”的浏览器就会通过脚本来模拟这一属性了！