**jQuery表单事件之submit事件**

提交表单是一个最常见的业务需求，比如用户注册，一些信息的输入都是需要表单的提交。同样的有时候开发者需要在表单提交的时候过滤一些的数据、做一些必要的操作（例如：验证表单输入的正确性，如果错误就阻止提交，从新输入）此时可以通过submit事件，监听下提交表单的这个动作

使用上非常简单，与基本事件参数处理保持一致

**方法一：$ele.submit()**

绑定$ele元素，不带任何参数一般是用来指定触发一个事件，用的比较少

<div id="test">点击触发<div>

$("ele").submit(function(){

alert('触发指定事件')

})

$("#text").click(function(){

$("ele").submit() //指定触发事件

});

**方法二：$ele.submit( handler(eventObject) )**

绑定$ele元素，每次$ele元素触发点击操作会执行回调 handler函数

这样可以针对事件的反馈做很多操作了

<form id="target" action="destination.html">

  <input type="submit" value="Go" />

</form>

$("#target").submit(function() { //绑定提交表单触发

//this指向 from元素

});

**方法三：$ele.submit( [eventData ], handler(eventObject) )**

使用与方法二一致，不过可以接受一个数据参数，这样的处理是为了解决不同作用域下数据传递的问题

<form id="target" action="destination.html">

  <input type="submit" value="Go" />

</form>

$("#target").submit(11111,function(data) { //绑定提交表单触发

//data => 1111 //传递的data数据

});

通过在<form>元素上绑定submit事件，开发者可以监听到用户的提交表单的的行为

具体能触发submit事件的行为：

* <input type="submit">
* <input type="image">
* <button type="submit">
* 当某些表单元素获取焦点时，敲击Enter（回车键）

上述这些操作下，都可以截获submit事件。

这里需要特别注意：

form元素是有默认提交表单的行为，如果通过submit处理的话，需要禁止浏览器的这个默认行为

传统的方式是调用事件对象 e.preventDefault() 来处理， jQuery中可以直接在函数中最后结尾return false即可

jQuery处理如下：

$("#target").submit(function(data) {

return false; //阻止默认行为，提交表单

});