基础事件

学习要点：

1.绑定事件

2.简写事件

3.复合事件

JavaScript有一个非常重要的功能，就是事件驱动。当页面完全加载后，用户通过鼠标

或键盘触发页面中绑定事件的元素即可触发。jQuery为开发者更有效率的编写事件行为，封

装了大量有益的事件方法供我们使用。

一．绑定事件

在 JavaScript课程的学习中，我们掌握了很多使用的事件，常用的事件有：click、dblclick、

mousedown、mouseup、mousemove、mouseover、mouseout、change、select、submit、keydown、

keypress、keyup、blur、focus、load、resize、scroll、error。那么，还有更多的事件可以参考

手册中的事件部分。

jQuery通过.bind()方法来为元素绑定这些事件。可以传递三个参数：bind(type, [data], fn)，

type表示一个或多个类型的事件名字符串；[data]是可选的，作为 event.data属性值传递一个

额外的数据，这个数据是一个字符串、一个数字、一个数组或一个对象；fn表示绑定到指

定元素的处理函数。

//使用点击事件

$('input').bind('click', function () {//点击按钮后执行匿名函数

alert('点击！');

});

//普通处理函数

$('input').bind('click', fn);//执行普通函数式无须圆括号

function fn() {

alert('点击！');

}

//可以同时绑定多个事件

$('input').bind('mouseout mouseover', function () {//移入和移出分别执行一次

$('div').html(function (index, value) {

return value + '1';

});

});

//通过对象键值对绑定多个参数

$('input').bind({//传递一个对象

'mouseout' : function () {//事件名的引号可以省略

alert('移出');

},

'mouseover' : function () {

alert('移入');

}

});

//使用 unbind删除绑定的事件

$('input').unbind();//删除所有当前元素的事件

//使用 unbind参数删除指定类型事件

$('input').unbind('click');//删除当前元素的 click事件

//使用 unbind参数删除指定处理函数的事件

function fn1() {

alert('点击 1');

}

function fn2() {

alert('点击 2');

}

$('input').bind('click', fn1);

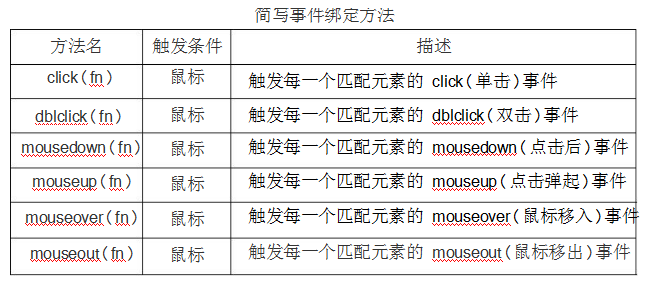
$('input').bind('click', fn2);

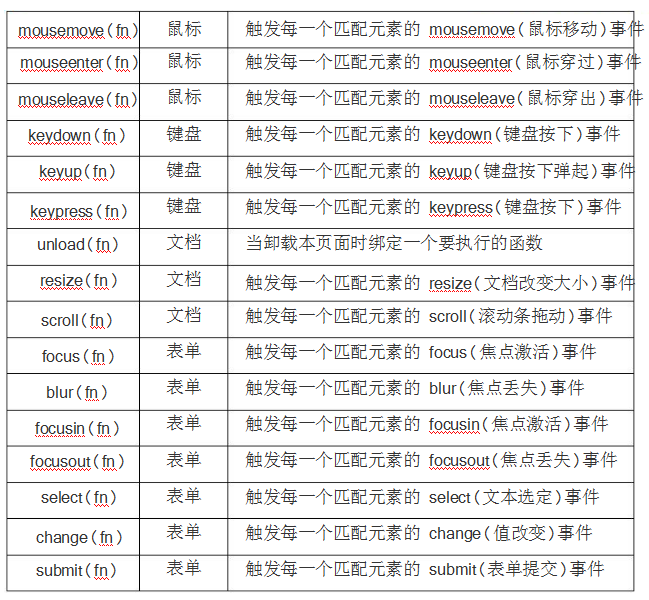
$('input').unbind('click', fn1);//只删除 fn1处理函数的事件

二．简写事件

为了使开发者更加方便的绑定事件，jQuery封装了常用的事件以便节约更多的代码。我

们称它为简写事件。





注意：这里封装的大部分方法都比较好理解，我们没必要一一演示确认，重点看几个需

要注意区分的简写方法。

.mouseover()和.mouseout()表示鼠标移入和移出的时候触发。那么 jQuery还封装了另外

一组：.mouseenter()和.mouseleave()表示鼠标穿过和穿出的时候触发。那么这两组本质上有

什么区别呢？手册上的说明是： .mouseenter()和.mouseleave()这组穿过子元素不会触发，

而.mouseover()和.mouseout()则会触发。

//HTML页面设置

<div style="width:200px;height:200px;background:green;">

<p style="width:100px;height:100px;background:red;"></p>

</div>

<strong></strong>

//mouseover移入

$('div').mouseover(function () {//移入 div会触发，移入 p再触发

$('strong').html(function (index, value) {

return value+'1';

});

});

//mouseenter穿过

$('div').mouseenter(function () {//穿过 div或者 p

$('strong').html(function (index, value) {//在这个区域只触发一次

return value+'1';

});

});

//mouseout移出

$('div').mouseout(function () {//移出 p会触发，移出 div再触发

$('strong').html(function (index, value) {

return value+'1';

});

});

//mouseleave穿出

$('div').mouseleave(function () {//移出整个 div区域触发一次

$('strong').html(function (index, value) {

return value+'1';

});

});

.keydown()、.keyup()返回的是键码，而.keypress()返回的是字符编码。

$('input').keydown(function (e) {

alert(e.keyCode);//按下 a返回 65

});

$('input').keypress(function (e) {

alert(e.charCode);

});//按下 a返回 97

注意：e.keyCode和 e.charCode在两种事件互换也会产生不同的效果，除了字符还有一

些非字符键的区别。更多详情可以了解 JavaScript事件处理那章。

.focus()和.blur()分别表示光标激活和丢失，事件触发时机是当前元素。而.focusin()

和.focusout()也表示光标激活和丢失，但事件触发时机可以是子元素。

//HTML部分

<div style="width:200px;height:200px;background:red;">

<input type="text" value="" />

</div>

<strong></strong>

//focus光标激活

$('input').focus(function () {//当前元素触发

$('strong').html('123');

});

//focusin光标激活

$('div').focusin(function () {//绑定的是 div元素，子类 input触发

$('strong').html('123');

});

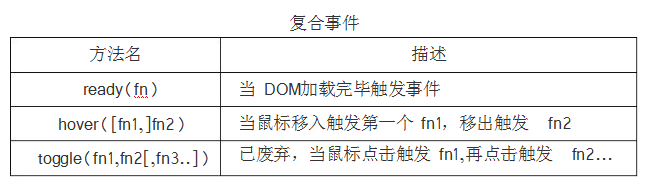
注意：.blur()和.focusout()表示光标丢失，和激活类似，一个必须当前元素触发，一个可

以是子元素触发。

三．复合事件

jQuery提供了许多最常用的事件效果，组合一些功能实现了一些复合事件，比如切换功

能、智能加载等。



//背景移入移出切换效果

$('div').hover(function () {//mouseenter效果

$(this).css('background', 'black');

}, function () {//mouseleave效果，可省略

$(this).css('background', 'red');

});

注意：.hover()方法是结合了.mouseenter()方法和.mouseleva()方法，并非.mouseover()

和.mouseout()方法。