第 **9** 章 事件对象

学习要点：

1

2

.事件对象

.冒泡和默认行为

JavaScript 在事件处理函数中默认传递了 event 对象，也就是事件对象。但由于浏览器

的兼容性，开发者总是会做兼容方面的处理。jQuery 在封装的时候，解决了这些问题，并且

还创建了一些非常好用的属性和方法。

一．事件对象

事件对象就是 event 对象，通过处理函数默认传递接受。之前处理函数的 e 就是 event

事件对象，event 对象有很多可用的属性和方法，我们在 JavaScript 课程中已经详细的了解

过这些常用的属性和方法，这里，我们再一次演示一下。

//通过处理函数传递事件对象

$

('input').bind('click', function (e) {

//接受事件对象参数

alert(e);

});

event 对象的属性

属性名

描述

type

target

获取这个事件的事件类型，例如：click

获取绑定事件的 DOM 元素

data

获取事件调用时的额外数据

relatedTarget

currentTarget

pageX/pageY

screenX/screenY

clientX/clientY

result

获取移入移出目标点离开或进入的那个 DOM 元素

获取冒泡前触发的 DOM 元素，等同与 this

获取相对于页面原点的水平/垂直坐标

获取显示器屏幕位置的水平/垂直坐标(非 jQuery 封装)

获取相对于页面视口的水平/垂直坐标(非 jQuery 封装)

获取上一个相同事件的返回值

timeStamp

which

获取事件触发的时间戳

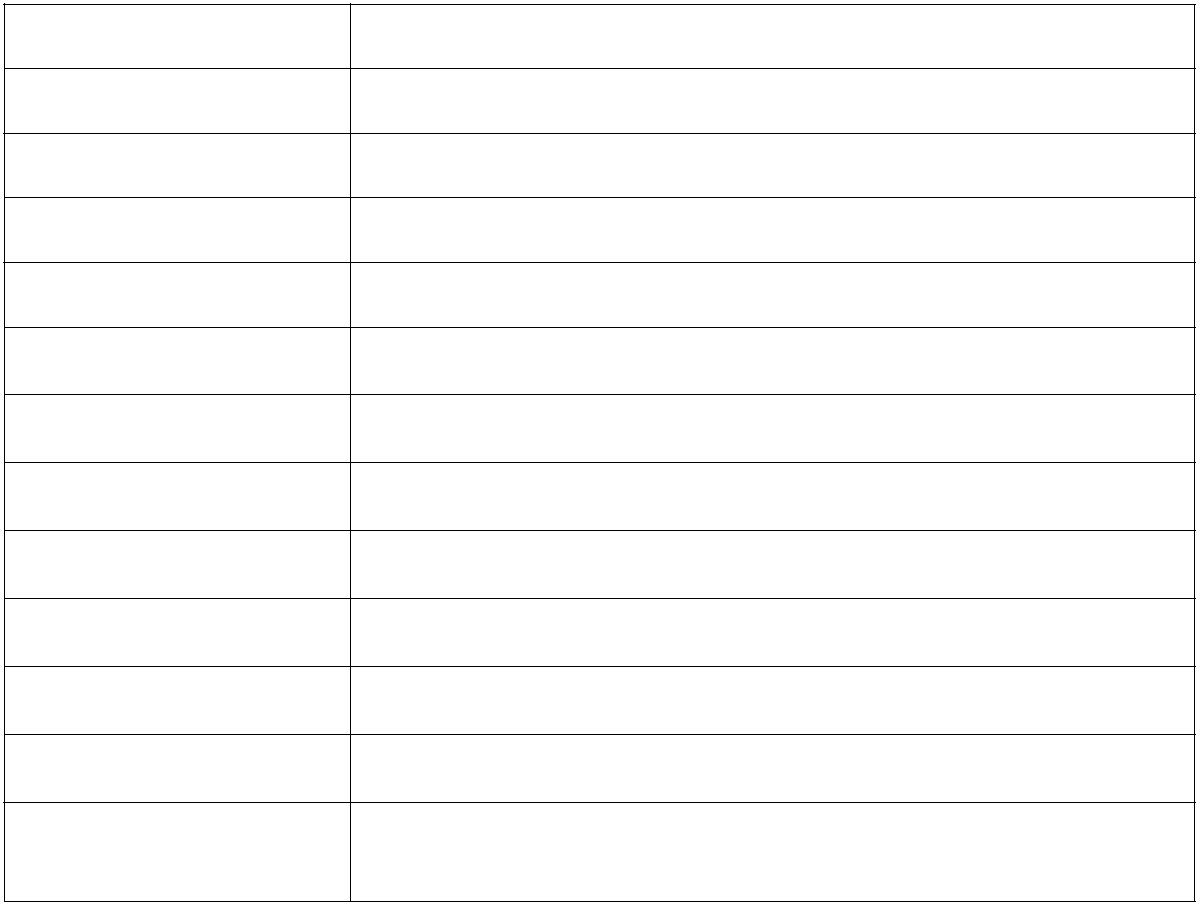
获取鼠标的左中右键(1,2,3)，或获取键盘按键

altKey/shiftKey/

ctrlKey/metaKey

获取是否按下了 alt、shift、ctrl(这三个非 jQuery 封装)或

meta 键(IE 原生 meta 键，jQuery 做了封装)



//通过 event.type 属性获取触发事件名

$

('input').click(function (e) {

alert(e.type);

});

//通过 event.target 获取绑定的 DOM 元素

$

('input').click(function (e) {

alert(e.target);

});

//通过 event.data 获取额外数据，可以是数字、字符串、数组、对象

$

('input').bind('click', 123, function () {

//传递 data 数据

//获取数字数据

alert(e.data);

});

注意：如果字符串就传递：'123'、如果是数组就传递：[123,'abc']，如果是对象就传递：

user : 'Lee', age : 100}。数组的调用方式是：e.data[1]，对象的调用方式是：e.data.user。

{

//event.data 获取额外数据，对于封装的简写事件也可以使用

$

('input').click({user : 'Lee', age : 100},function (e) {

alert(e.data.user);

});

注意：键值对的键可以加上引号，也可以不加；在调用的时候也可以使用数组的方式：

alert(e.data['user']);

//获取移入到 div 之前的那个 DOM 元素

$

('div').mouseover(function (e) {

alert(e.relatedTarget);

});

//获取移出 div 之后到达最近的那个 DOM 元素

$

('div').mouseout(function (e) {

alert(e.relatedTarget);

});

//获取绑定的那个 DOM 元素，相当于 this，区别与 event.target

$

('div').click(function (e) {

alert(e.currentTarget);

);

}

注意：event.target 得到的是触发元素的 DOM，event.currentTarget 得到的是监听元素的

DOM。而 this 也是得到监听元素的 DOM。

//获取上一次事件的返回值

$

}

$

}

('div').click(function (e) {

return '123';

);

('div').click(function (e) {

alert(e.result);

);

//获取当前的时间戳

$

('div').click(function (e) {

alert(e.timeStamp);

});

//获取鼠标的左中右键

$

('div').mousedown(function (e) {

alert(e.which);

});

//获取键盘的按键

$

('input').keyup(function (e) {

alert(e.which);

});

//获取是否按下了 ctrl 键，meta 键不存在，导致无法使用

$

('input').click(function (e) {

alert(e.ctrlKey);

});

//获取触发元素鼠标当前的位置

$

(document).click(function (e) {

alert(e.screenY+ ',' + e.pageY + ',' + e.clientY);

});

二．冒泡和默认行为

如果在页面中重叠了多个元素，并且重叠的这些元素都绑定了同一个事件，那么就会出

现冒泡问题。

//HTML 页面

<

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

input type="button" value="按钮" />

/div>

<

<

//三个不同元素触发事件

$

('input').click(function () {