

第8章 基础事件

学习要点：

1. 绑定事件
2. 简写事件
3. 复合事件

主讲教师：李炎恢

官方网站：<http://www.ycku.com>

合作网站：<http://www.ibeifeng.com>

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({
  'mouseout' : function () {
    alert('移出');
  },
  'mouseover' : function () {
    alert('移入');
  }
});
```

//传递一个对象

//事件名的引号可以省略

//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。

简写事件绑定方法

方法名	触发条件	描述
click(fn)	鼠标	触发每一个匹配元素的 click(单击)事件
dblclick(fn)	鼠标	触发每一个匹配元素的 dblclick(双击)事件
mousedown(fn)	鼠标	触发每一个匹配元素的 mousedown(点击后)事件
mouseup(fn)	鼠标	触发每一个匹配元素的 mouseup(点击弹起)事件
mouseover(fn)	鼠标	触发每一个匹配元素的 mouseover(鼠标移入)事件
mouseout(fn)	鼠标	触发每一个匹配元素的 mouseout(鼠标移出)事件

mousemove(fn)	鼠标	触发每一个匹配元素的 mousemove(鼠标移动)事件
mouseenter(fn)	鼠标	触发每一个匹配元素的 mouseenter(鼠标穿过)事件
mouseleave(fn)	鼠标	触发每一个匹配元素的 mouseleave(鼠标穿出)事件
keydown(fn)	键盘	触发每一个匹配元素的 keydown(键盘按下)事件
keyup(fn)	键盘	触发每一个匹配元素的 keyup(键盘按下弹起)事件
keypress(fn)	键盘	触发每一个匹配元素的 keypress(键盘按下)事件
unload(fn)	文档	当卸载本页面时绑定一个要执行的函数
resize(fn)	文档	触发每一个匹配元素的 resize(文档改变大小)事件
scroll(fn)	文档	触发每一个匹配元素的 scroll(滚动条拖动)事件
focus(fn)	表单	触发每一个匹配元素的 focus(焦点激活)事件
blur(fn)	表单	触发每一个匹配元素的 blur(焦点丢失)事件
focusin(fn)	表单	触发每一个匹配元素的 focusin(焦点激活)事件
focusout(fn)	表单	触发每一个匹配元素的 focusout(焦点丢失)事件
select(fn)	表单	触发每一个匹配元素的 select(文本选定)事件
change(fn)	表单	触发每一个匹配元素的 change(值改变)事件
submit(fn)	表单	触发每一个匹配元素的 submit(表单提交)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 () {  
    $( 'strong' ).html( function ( index, value ) {  
        return value + '1';  
    });  
});
```

//穿过 div 或者 p
//在这个区域只触发一次

//mouseout 移出

```
$( 'div' ).mouseout( function () {  
    $( 'strong' ).html( function ( index, value ) {  
        return value + '1';  
    });  
});
```

//移出 p 会触发，移出 div 再触发

//mouseleave 穿出

```
$( 'div' ).mouseleave( function () {  
    $( 'strong' ).html( function ( index, value ) {  
        return value + '1';  
    });  
});
```

//移出整个 div 区域触发一次

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

//当前元素触发

```
});
```

```
//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。

复合事件

方法名	描述
ready(fn)	当 DOM 加载完毕触发事件
hover([fn1,]fn2)	当鼠标移入触发第一个 fn1，移出触发 fn2
toggle(fn1,fn2[,fn3..])	已废弃，当鼠标点击触发 fn1,再点击触发 fn2...

```
//背景移入移出切换效果
```

```
$( 'div' ).hover( function () {
    $( this ).css( 'background', 'black' );                    //mouseenter 效果
}, function () {
    $( this ).css( 'background', 'red' );                      //mouseleave 效果，可省略
});
```

注意：.hover()方法是结合了.mouseenter()方法和.mouseleave()方法，并非.mouseover()和.mouseout()方法。

.toggle()这个方法比较特殊，这个方法有两层含义，第一层含义就是已经被 1.8 版废弃、1.9 版删除掉的用法，也就是点击切换复合事件的用法。第二层函数将会在动画那章讲解到。既然废弃掉了，就不应该使用。被删除的原因是：以减少混乱和提高潜在的模块化程度。

但你又非常想用这个方法，并且不想自己编写类似的功能，可以下载 jquery-migrate.js 文件，来向下兼容已被删除掉的方法。

```
//背景点击切换效果(1.9 版删除掉了)
```

```
<script type="text/javascript" src="jquery-migrate-1.2.1.js"></script>
$( 'div' ).toggle( function () {                                //第一次点击切换
    $( this ).css( 'background', 'black' );
}, function () {                                                //第二次点击切换
    $( this ).css( 'background', 'blue' );
});
```

```
}, function () {  
    $(this).css('background', 'red');  
});
```

//第三次点击切换

注意：由于官方已经删除掉这个方法，所以也是不推荐使用的，如果在不基于向下兼容的插件JS。我们可以自己实现这个功能。

```
var flag = 1; //计数器  
$('#div').click(function () {  
    if (flag == 1) { //第一次点击  
        $(this).css('background', 'black');  
        flag = 2;  
    } else if (flag == 2) { //第二次点击  
        $(this).css('background', 'blue');  
        flag = 3;  
    } else if (flag == 3) { //第三次点击  
        $(this).css('background', 'red');  
        flag = 1;  
    }  
});
```

感谢收看本次教程！

本课程是由北风网(ibeifeng.com)

瓢城 Web 俱乐部(ycku.com)联合提供：

本次主讲老师：李炎恢

谢谢大家，再见！