# 第九章 jquery进阶-事件操作和动画

本节所讲内容：

9.1 jQuery节点插入

9.2 jQuery绑定事件

9.3 jQuery动画效果

## 9.1 jQuery节点插入

在创建节点的过程中，其实我们已经演示怎么通过.append()方法来插入一个节点。但除了这个方法之余呢，jQuery 提供了其他几个方法来插入节点。

内部插入节点方法

|  |  |
| --- | --- |
| 方法名 | 描述 |
| append(content) | 向指定元素内部后面插入节点 content |
| append(function (index, html) {}) | 使用匿名函数向指定元素内部后面插入节点 |
| appendTo(content) | 将指定元素移入到指定元素 content 内部后面 |
| prepend(content) | 向指定元素 content 内部的前面插入节点 |
| prepend(function (index, html) {}) | 使用匿名函数向指定元素内部的前面插入节点 |
| prependTo(content) | 将指定元素移入到指定元素 content 内部前面 |

**$('div').append('<strong>节点</strong>');*//向 div 内部插入 strong 节点***

$('div').append(function (index,value) { *//使用匿名函数插入节点，html 是原节点* return '<strong>节点</strong>'  
})

**$('span').appendTo('div'); *//将 span 节点移入 div 节点内***

**$('span').appendTo($('div')); *//同上***

**$('div').prepend('<span>节点</span>'); *//将 span 插入到 div 内部的前面*$('div'). prepend (function (index, html) { *//使用匿名函数，同上* return '<span>节点</span>';  
});  
$('span').prependTo('div'); *//将 span 移入 div 内部的前面*$('span').prependTo($('div')); *//同上***

外部插入节点方法

|  |  |
| --- | --- |
| 方法名 | 描述 |
| after(content) | 向指定元素的外部后面插入节点 content |
| after(function (index, html) {}) | 使用匿名函数向指定元素的外部后面插入节点 |
| before(content) | 向指定元素的外部前面插入节点 content |
| before(function (index, html) {}) | 使用匿名函数向指定元素的外部前面插入节点 |
| insertAfter(content) | 将指定节点移到指定元素 content 外部的后面 |
| insertBefore(content) | 将指定节点移到指定元素 content 外部的前面 |

**$('div').after('<span>节点</span>'); *//向 div 的同级节点后面插入 span*$('div').after(function (index, html) { *//使用匿名函数，同上* return '<span>节点</span>';  
});**

**$('div').before('<span>节点</span>'); *//向 div 的同级节点前面插入 span*$('div').before(function (index, html) { *//使用匿名函数，同上* return '<span>节点</span>';  
});**

**$('span').insertAfter('div'); *//将 span 元素移到 div 元素外部的后面*$('span').insertBefore('div'); *//将 span 元素移到 div 元素外部的前面***

## 9.2 jQuery绑定事件

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***

## 9.3 jQuery动画效果

### 9.3.1 jQuery 中显示方法

jQuery 中显示方法为：.show(speed,callback)，隐藏方法为：.hide(speed,callback),停止方法：stop(stopAll,gotoEnd)。在无参数的时候，只是硬性的显示内容和隐藏内容。

**Speed可选的 speed 参数规定隐藏/显示的速度，可以取以下值："slow"、"fast" 或毫秒。**

**callback可选的 callback 参数是隐藏或显示完成后所执行的函数名称。**

**stopAll 参数规定是否应该清除动画队列。默认是 false，即仅停止活动的动画，允许任何排入队列的动画向后执行。**

**可选的 goToEnd 参数规定是否立即完成当前动画。默认是 false。**

**<script type="text/javascript" src="https://code.jquery.com/jquery-3.4.1.min.js"></script>  
 <style>  
 p{  
 width: 100px;  
 height: 100px;  
 background: cyan;  
 margin: 2px;  
 }  
 </style>  
……  
<body>  
<p id="box"></p>  
<button class="show" >显示</button>  
<button class="hide" >隐藏</button>  
<script>  
 $('.show').click(function () { *//显示  
 // alert('1');* $('#box').show();  
 });  
 $('.hide').click(function () { *//隐藏* $('#box').hide();  
 });**

注意：.hide()方法其实就是在行内设置 CSS 代码：display:none; 而.show()方法要根据原来元素是区块还是内联来决定，如果是区块，则设置 CSS 代码：display:block; 如果是内联，则设置 CSS 代码：display:inline;。

在.show()和.hide()方法可以传递一个参数，这个参数以毫秒(1000 毫秒等于 1 秒钟)来控制速度。并且里面富含了匀速变大变小，以及透明度变换。

**$('.show').click(function () { *//显示用了1秒  
 // alert('1');* $('#box').show(1000);  
});  
$('.hide').click(function () { *//隐藏* $('#box').hide(1000);  
});**

除了直接使用毫秒来控制速度外，jQuery 还提供了三种预设速度参数字符串：slow、

normal 和 fast，分别对应 600 毫秒、400 毫秒和 200 毫秒。

**$('.show').click(function () {  
 $('#box').show('fast'); *//200 毫秒*});  
$('.hide').click(function () {  
 $('#box').hide('slow'); *//600 毫秒*});**

注意：不管是传递毫秒数还是传递预设字符串，如果不小心传递错误或者传递空字符串,那么它将采用默认值：400 毫秒。

**$('.show').click(function () {  
 $('#box').show(''); *//默认 400 毫秒*});**

//使用.show()和.hide()的回调函数，可以实现列队动画效果。

**$('.show').click(function () {  
 $('#box').show('slow', function () {  
 *alert*('动画持续完毕后，执行我！');  
 });  
});**

//列队动画，使用函数名调用自身

**$('.show').click(function () {  
 $('div').first().show('fast', function showSpan() {  
 $(this).next().show('fast', *showSpan*);  
 });  
});**

//列队动画，使用 arguments.callee 匿名函数自调用

**$('.hide').click(function () {  
 $('div').last().hide('fast', function() {  
 $(this).prev().hide('fast', *arguments*.callee);  
 });  
});**

我们在使用.show()和.hide()的时候，如果需要一个按钮切换操作，需要进行一些条件判

断。而 jQuery 提供给我们一个类似功能的独立方法：.toggle()。toggle [ˈtɒɡl]:切换

**$('.show').click(function () {  
 $('#box').toggle('slow');  
});**

### 9.3.2 滑动、卷动

jQuery 提供了一组改变元素高度的方法：.slideUp()、.slideDown()和.slideToggle()。顾名思义，向上收缩(卷动)和向下展开(滑动)。  
$('.show').click(function () {  
 $('#box').slideDown();  
});  
$('.hide').click(function () {  
 $('#box').slideUp();  
});  
$('.show').click(function () {  
 $('#box').slideToggle();  
});

注意：滑动、卷动效果和显示、隐藏效果一样，具有相同的参数。

### 9.3.3 淡入、淡出

jQuery 提供了一组专门用于透明度变化的方法：.fadeIn()和.fadeOut()，分别表示淡入、淡出，当然还有一个自动切换的方法：.fadeToggle()。fade [feɪd]：褪色

**$('.show').click(function () {  
 $('#box').fadeIn('slow');  
});  
$('.hide').click(function () {  
 $('#box').fadeOut('slow');  
});  
$('.toggle').click(function () {  
 $('#box').fadeToggle();  
});**

上面三个透明度方法只能是从 0 到 100，或者从 100 到 0，如果我们想设置指定值就没有办法了。而 jQuery 为了解决这个问题提供了.fadeTo()方法。

**$('.toggle').click(function () {  
 $('#box').fadeTo('slow', 0.33); *//0.33 表示值为 33*});**

注意：淡入、淡出效果和显示、隐藏效果一样，具有相同的参数。对于.fadeTo()方法，如果本身透明度大于指定值，会淡出，否则相反。

### 9.3.4 自定义动画

jQuery 提供了几种简单常用的固定动画方面我们使用。但有些时候，这些简单动画无法满足我们更加复杂的需求。这个时候，jQuery 提供了一个.animate()方法来创建我们的自定义动画，满足更多复杂多变的要求。animate [ˈænɪmeɪt , ˈænɪmət]：使具有活力。

**$('.show').click(function () {  
 $('#box').animate({  
 'width' : '300px',  
 'height' : '200px',  
 'fontSize' : '50px',  
 'opacity' : 0.5  
 });  
});**

注意：一个 CSS 变化就是一个动画效果，上面的例子中，已经有四个 CSS 变化，已经实现了多重动画同步运动的效果。必传的参数只有一个，就是一个键值对 CSS 变化样式的对象。还有两个可选参数分别为速度和回调函数。

**$('.show').click(function () {  
 $('#box').animate({  
 'width' : '300px',  
 'height' : '200px'  
 }, 1000, function () {  
 *alert*('动画执行完毕执行我！');  
 });  
});**

到目前位置，我们都是创建的固定位置不动的动画。如果想要实现运动状态的位移动画，那就必须使用自定义动画，并且结合 CSS 的定位功能。

**$('.show').click(function () {  
 $('#box').animate({  
 'top' : '300px', *//先必须设置 CSS 绝对定位* 'left' : '200px'  
 });  
});**

自定义动画中，每次开始运动都必须是初始位置或初始状态，而有时我们想通过当前位置或状态下再进行动画。jQuery 提供了自定义动画的累加、累减功能。

**$('.show').click(function () {  
 $('#box').animate({  
 'left' : '+=100px',  
 });  
});**

自定义实现列队动画的方式，有两种：1.在回调函数中再执行一个动画；2.通过连缀或顺序来实现列队动画。

*//通过依次顺序实现列队动画***$('.show').click(function () {  
 $('#box').animate({'left' : '100px'});  
 $('#box').animate({'top' : '100px'});  
 $('#box').animate({'width' : '300px'});  
});**

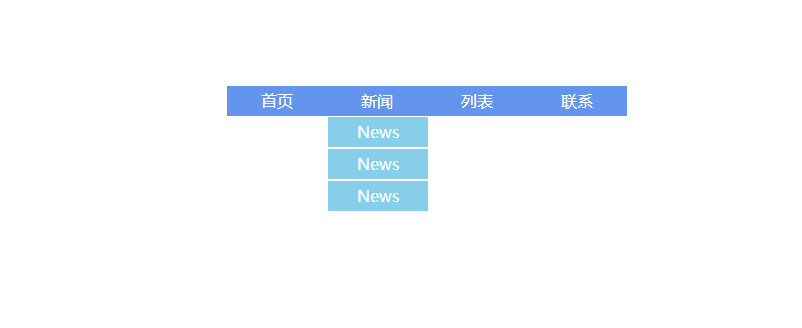
注意：如果不是同一个元素，就会实现同步动画

//通过连缀实现列队动画

**$('.show').click(function () {  
 $('#box').animate({  
 'left' : '100px'  
 }).animate({  
 'top' : '100px'  
 }).animate({  
 'width' : '300px'  
 });  
});**

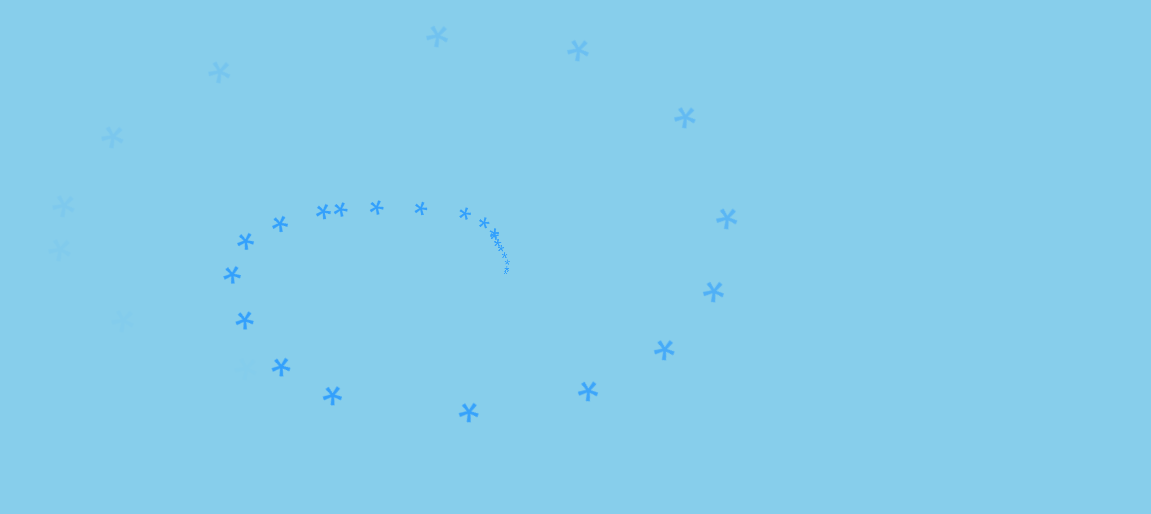
***//通过回调函数实现列队动画*$('.show').click(function () {  
 $('#box').animate({  
 'left' : '100px'  
 }, function () {  
 $('#box').animate({  
 'top' : '100px'  
 }, function () {  
 $('#box').animate({  
 'width' : '300px'  
 });  
 });  
 });  
});**

**实战：结合本节课实现如下图所示内容**



<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>下拉展示</title>  
 <script type="text/javascript" src="https://code.jquery.com/jquery-3.4.1.min.js"></script>  
 <style type="text/css" rel="stylesheet">  
 \*{  
 margin: 0;  
 padding:0;  
 }  
 div{  
 width: 400px;  
 height: 50px;  
 margin: 0 auto;  
 top: 200px;  
 */\*相对于body进行定位保证top属性能够正常执行\*/* position: relative;  
 }  
 ul{  
 list-style-type: none;  
 }  
 ul li{  
 float: left;  
 width: 100px;  
 height:30px;  
 line-height: 30px;  
 text-align:center;  
 color: #fff;  
 background: cornflowerblue;  
 }  
 .item ul{  
 display: none;  
 }  
 .item ul li{  
 border: 1px solid white;  
 background: skyblue;  
 }  
 </style>  
</head>  
<body>  
<div>  
 <ul>  
 <li class="item">首页  
 <ul>  
 <li>Home</li>  
 <li>Home</li>  
 <li>Home</li>  
 </ul>  
 </li>  
 <li class="item">美女  
 <ul>  
 <li>孙二</li>  
 <li>rain</li>  
 <li>gerber</li>  
 </ul>  
 </li>  
 <li class="item">帅哥  
 <ul>  
 <li>for</li>  
 <li>for</li>  
 <li>for</li>  
 </ul>  
 </li>  
 <li class="item">联系  
 <ul>  
 <li>Tel</li>  
 <li>Tel</li>  
 <li>Tel</li>  
 </ul>  
 </li>  
 </ul>  
</div>  
<script type="text/javascript">  
 $(function () {  
 $(".item").mouseenter(function(){  
 $(this).find("ul").stop(false,true).slideDown(500);  
 });  
 $(".item").mouseleave(function () {  
 $(this).find("ul").stop(false,true).slideUp(500);  
 });  
 });  
   
</script>  
</body>

**趣味实战：**



<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>Title</title>  
 <script type="text/javascript" src="https://code.jquery.com/jquery-3.4.1.min.js"></script>  
 <style>  
 body{background-color: skyblue}  
 html {  
 */\*设置鼠标\*/* cursor: url('./menghuan.ico'),auto;  
 }  
 .draw {  
 position: fixed;  
 width: 1px;  
 line-height: 1px;  
 pointer-events: none;  
 }  
 */\*使用@keyframes规则，你可以创建动画。\*/  
 /\*创建动画是通过逐步改变从一个CSS样式设定到另一个。\*/* @keyframes floatOne {  
 0% {  
 opacity:1;  
 }  
 50% {  
 opacity:1;  
 }  
 100% {  
 */\* opacity[əʊˈpæsəti]：不透明度 scale：[skeɪl]：规模\* rotate[rəʊˈteɪt]：转动\*/* opacity:0;  
 transform:translate3D(0, -20px, 0) scale(5) rotate(45deg);  
 }  
 }  
 </style>  
</head>  
<body>  
<script>  
 $(*document*).bind('mousemove touchmove',function(e) {  
 e.preventDefault();*//取消事件的默认行为* var drawSize = 10;  
 var drawType = 'for';  
 var floatType = 'floatOne';  
 var xPos = e.originalEvent.pageX;*//事件作用的可见的最大区域* var yPos = e.originalEvent.pageY;*//事件作用的可见的最大区域* $('body').append('<div class="draw" style=" font-size:'+drawSize+'px;left:'+xPos+'px;top:'+yPos+'px;-webkit-animation:'+floatType+' .9s 1;-moz-animation:'+floatType+' .9s 1;color:#33a0fc;">'+drawType+'</div>');  
  
 $('.draw').each(function() {  
 var div = $(this);  
 *setTimeout*(function() {$(div).remove();},800);  
 });  
 });  
  
</script>  
</body>  
</html>

总结：

9.1 jQuery节点插入

9.2 jQuery绑定事件

9.3 jQuery动画效果