

# jQuery事件、动画

## 一、事件

### 1.1 鼠标事件

事件	说明
click	鼠标点击某个对象时触发
dblclick	鼠标双击某个对象时触发
mouseover	鼠标被移到某元素之上，穿过子元素时事件会再次触发
mouseout	鼠标从某元素移开，穿过子元素时事件会再次触发
mouseenter	鼠标被移到某元素之上，穿过子元素时事件不会再次触发
mouseleave	鼠标从某元素移开，穿过子元素时事件不会再次触发
mousemove	鼠标移动时触发
mousedown	某个鼠标按键按下时触发
mouseup	某个鼠标按键松开时触发

#### 代码演示

```
<div id="div1">
  <div id="div2"></div>
</div>
<h1></h1>
```

```
// mouseover和mouseout在父子元素之间切换时会再次触发进入离开的事件
var num = 0;
$('.parent').mouseover(function(){
    $('h1').html(++num);
})

$('.parent').mouseout(function(){
    $('h1').html(++num);
})

// mouseenter和mouseleave在父子元素之间切换时不会再次触发进入离开的事件
$('.parent').mouseenter(function(){
    $('h1').html(++num);
});

$('.parent').mouseleave(function(){
    $('h1').html(++num);
})
```

## 1.2 键盘事件

事件	说明
keydown	键盘的某个键被按下或按住时触发
keyup	键盘的某个键被松开时触发

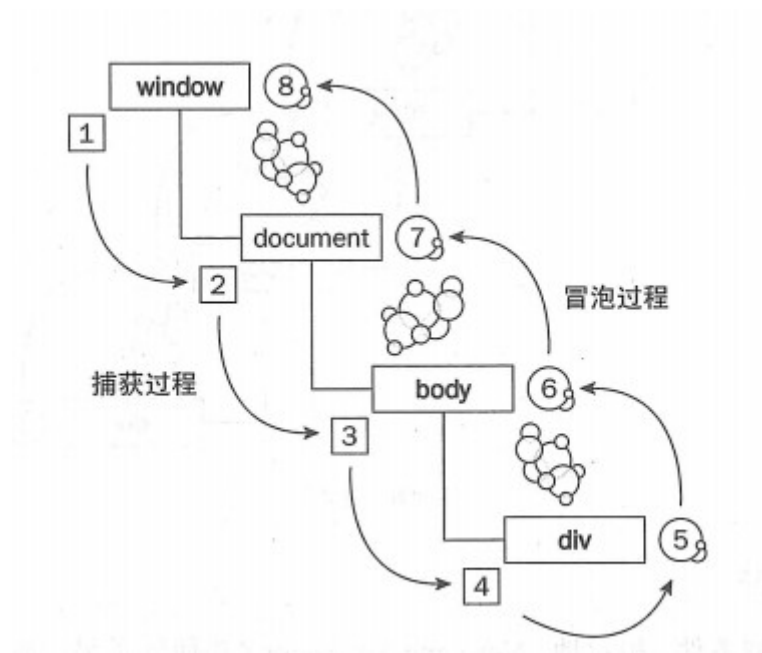
## 1.3 表单事件

事件	说明
focus	当元素获得焦点时触发
blur	当元素失去焦点时触发
submit	提交按钮被点击时触发
select	文本内容被选定时触发
change	用户改变表单元素的内容时触发

## 二、事件处理

### 2.1 事件冒泡

- 事件发生时会在元素节点之间按照特定的顺序传播，这个传播过程即DOM事件流



- DOM事件流分为3个阶段：
  - 捕获阶段：IE最早提出，事件逐级向下传播
  - 当前目标阶段
  - 冒泡阶段：网景最早提出，事件逐级向上传播
- 事件逐级向上传播的过程就是事件冒泡。
- 阻止事件冒泡：**
  - `event.stopPropagation()`
  - `return false`

## 2.2 事件绑定

语法:

```
// 1、绑定单个事件
$("#div").on("click", function(){
    console.log("绑定单个事件")
});

// 2、绑定多个事件
// 2.1 不同事件同一操作
$("#div").on("mouseover mouseout", function() {
    $(this).toggleClass("red");
})
// 2.2 不同事件不同操作
$("#div").on({
    "mouseenter": function() {
        $(this).css("background", "red");
    },
    "mouseleave": function() {
        $(this).css("background", "green");
    },
    "click": function() {
        $(this).css("background", "blue");
    }
})

// 3、取消绑定事件
// 3.1 取消全部事件
$("#div").off();
// 3.2 取消指定事件
$("#div").off('click mouseleave');
```

## 2.3 事件委派

事件委派的作用:

- 1、只给上级元素添加事件，不需要给每一个子元素添加事件，从而提高了程序的性能
- 2、委派的事件对后来新增的元素同样有效

语法:

```
// 委派
$("#上级元素").on("事件", "子元素", function(){
    alert( "test");
});

// 取消委派
$("#上级元素").off("事件");
```

## 2.4 事件切换

```
$("#div").hover(  
    function(){  
        // mouseenter事件  
    },  
    function(){  
        // mouseleave事件  
    }  
)
```

## 2.5 事件对象

### 2.5.1 鼠标事件对象

```
$("#div").mousemove(function(event){  
    // event.pageX 鼠标x轴坐标  
    // event.pageY 鼠标y轴坐标  
})
```

### 2.5.2 键盘事件对象

```
$(document).keydown(function(event){  
    // event.which 键盘对应ASCII编码  
})
```

## 案例：发布留言

请输入留言内容

提交

南风知我意，吹梦到西洲。你等南风起,我等故人归

删除

南风知我意，吹梦到西洲

删除

冬天已然来临，春天还会远吗？

删除

HTML部分：

```
<div class="msg">
  <textarea name="content" placeholder="请输入留言内容"></textarea>
  <button type="button">提 交</button>
</div>
<ul class="list"></ul>
```

CSS部分:

```
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

.msg {
  width: 600px;
  height: 200px;
  margin: 20px auto;
  position: relative;
}

.msg textarea {
  width: 100%;
  height: 100%;
  padding: 8px;
  border: 1px solid #ccc;
  border-radius: 5px;
  outline: none;
  resize: none;
}

.msg button {
  position: absolute;
  right: 30px;
  bottom: 10px;
  padding: 5px 35px;
  border: 1px solid #ccc;
  border-radius: 3px;
  background: #efefef;
  outline: none;
  cursor: pointer;
}

.msg button:hover {
  background: #ddd;
}

ul {
  list-style: none;
  width: 600px;
  margin: 0 auto;
}

li {
  padding: 20px 0;
  border-bottom: 1px dotted #ccc;
}
```

```

    color: #333;
    display: none;
}

li:last-of-type {
    border-bottom: 0;
}

li a {
    float: right;
    color: rgb(33, 33, 245);
    text-decoration: none;
}

```

JS部分:

```

// 添加留言
$("#button").click(function() {
    // 生成留言插入列表
    var li = $("- </li>");
    li.html("<span>" + $("#textarea").val() + "</span><a href='#>删除</a>");
    li.prependTo(".list");
    // 滑动显示
    li.slideDown();
    // 清除表单内容
    $("#textarea").val("");
})

// 删除留言
$("#ul").on("click", "li>a", function(event) {
    $(this).parent("li").slideUp(function() {
        $(this).detach();
    });
});
})

```

## 三、动画

### 3.1 隐藏显示效果

- show([speed],[easing],[fn])
  - 显示隐藏的元素
    - speed:三种预定速度之一的字符串("slow","normal", or "fast")或表示动画时长的毫秒数值(如: 1000)
    - easing:(Optional) 用来指定切换效果, 默认是"swing", 可用参数"linear"
    - fn:在动画完成时执行的函数, 每个元素执行一次。
- hide([speed],[easing],[fn])
  - 隐藏元素
- toggle([speed],[easing],[fn])
  - 如果元素是可见的, 切换为隐藏的; 如果元素是隐藏的, 切换为可见的

## 3.2 上卷下拉效果

- `slideDown([speed],[easing],[fn])`
  - 显示隐藏的元素
- `slideUp([speed],[easing],[fn])`
  - 隐藏元素
- `slideToggle([speed],[easing],[fn])`
  - 如果元素是可见的，切换为隐藏的；如果元素是隐藏的，切换为可见的

### 案例：菜单特效



HTML:

```
<dl>
  <dt>商品管理</dt>
  <dd>商品列表</dd>
  <dd>添加商品</dd>
</dl>
<dl>
  <dt>分类管理</dt>
  <dd>分类列表</dd>
  <dd>添加分类</dd>
</dl>
<dl>
  <dt>订单管理</dt>
  <dd>订单列表</dd>
  <dd>添加订单</dd>
</dl>
<dl>
  <dt>广告管理</dt>
  <dd>广告列表</dd>
  <dd>添加广告</dd>
</dl>
<dl>
```

```
<dt>会员管理</dt>
<dd>会员列表</dd>
<dd>添加会员</dd>
</dl>
```

CSS:

```
dl {
  margin: 1px;
}

dt,
dd {
  width: 120px;
  padding: 8px 15px;
  margin: 1px;
  background-color: #216891;
  color: #fff;
  text-align: center;
  cursor: pointer;
}

dt {
  font-weight: bold;
}

dd {
  background-color: rgb(122, 175, 236);
  display: none;
}
```

JS:

```
$('#dt').click(function() {
  $(this).nextAll('dd').slideToggle().parent().siblings('dl').children('dd').slideUp();
})
```

### 3.3 淡入淡出效果

- fadeIn([speed],[easing],[fn])
  - 显示隐藏的元素
- fadeOut([speed],[easing],[fn])
  - 隐藏元素
- fadeToggle([speed],[easing],[fn])
  - 如果元素是可见的，切换为隐藏的；如果元素是隐藏的，切换为可见的
- fadeTo([speed],opacity,[easing],[fn])
  - opacity: 一个0至1之间表示透明度的数字



### 3.4 自定义动画效果

- `animate(params,[speed],[easing],[fn])`
  - `params`:一组包含作为动画属性和终值的样式属性和及其值的集合
- `stop([clearQueue],[jumpToEnd])`
  - `clearQueue`:默认为`false`。如果设置成`true`，则清空队列。可以立即结束动画。
  - `jumpToEnd`:默认为`false`。如果设置成`true`，则完成队列。可以立即完成动画
    - `stop()`: 停止当前动画，如果队列中有等待执行的动画，他们将被马上执行
    - `stop(true)`: 如果同一元素调用多个动画方法，尚未被执行的动画被放置在元素的效果队列中。这些动画不会开始，直到第一个完成
    - `stop(true,true)`: 当前动画将停止，但该元素上的 CSS 属性会被立刻修改成动画的目标值

#### 代码展示

```
<div id="div1"></div>
<button>开始</button>
<button>停止</button>
```

```
//所有动画效果同时执行
$('button:first').click(function(){
    $('#div1').animate({
        left: '1100px',
        top: '500px',
        width: '160px',
        height: '50px'
    },5000,'swing');
})

$('button:last').click(function(){
    $('#div1').stop();
})
```

```
//所有动画效果依次执行
$('button:first').click(function() {
    $('#div1').animate({
        left: '1100px'
    }, 2000)
    .animate({
        top: '500px'
    }, 2000)
    .animate({
        width: '160px'
    }, 2000)
    .animate({
        height: '80px'
    }, 2000)
})
```

//第一个参数表示`clearQueue`(清除队列)，默认为`false`，表示当前动画停止后，后面的动画会继续执行，设置成`true`则后面的动画不会继续执行

//第二个参数表示`gotoEnd`（立即完成），默认为`false`，如果设置为`true`，则表示当前动画停止后，会让当前正在执行的动画效果立即完成

```
$('button:last').click(function() {  
    $('#div1').stop(true, false);  
})
```