



传智播客.黑马程序员

# 第4章 jQuery事件处理机制

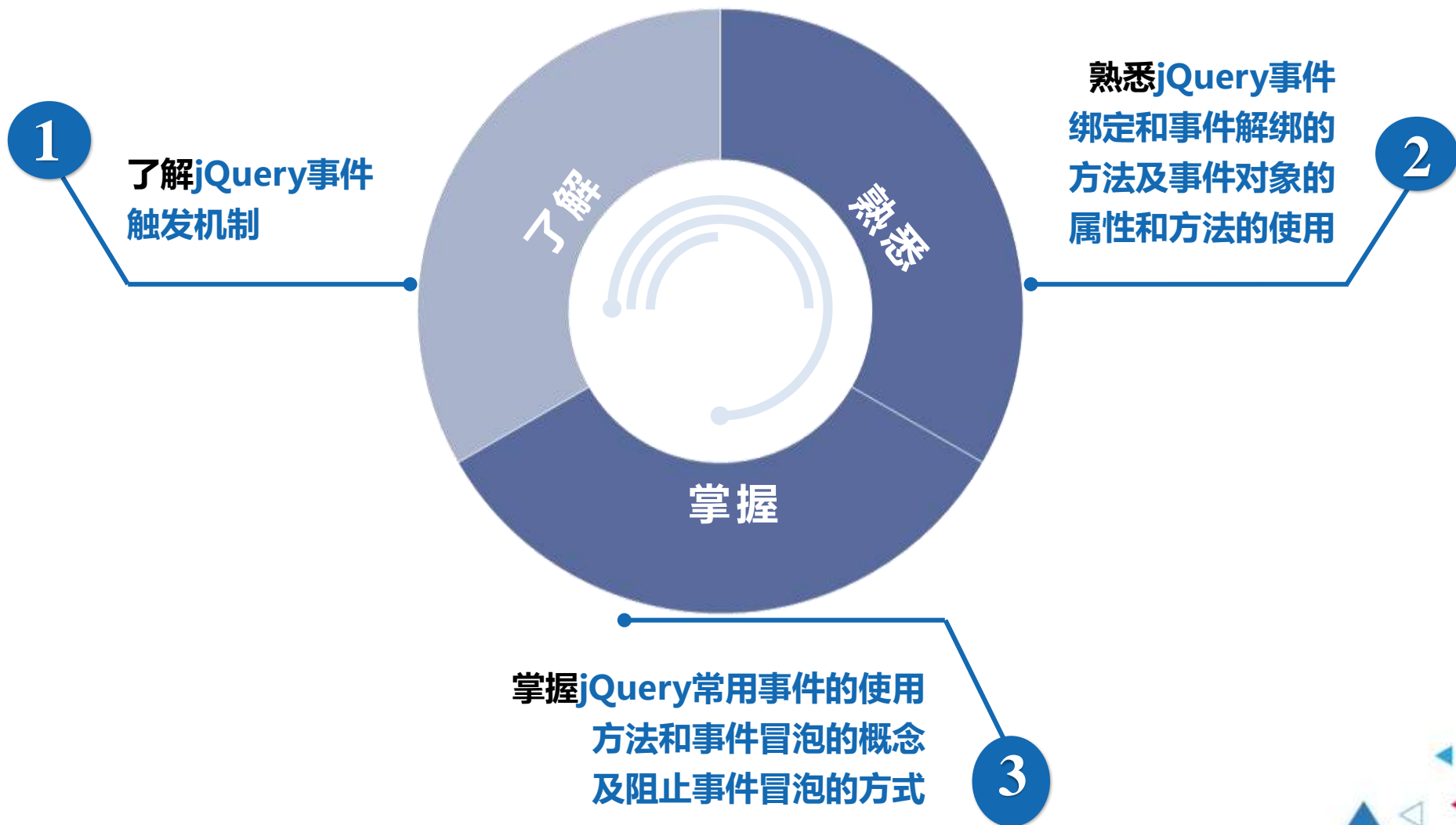
jQuery

- 常用事件
- 事件绑定与事件解绑
- 事件触发和冒泡
- 事件对象



# 学习目标

传智播客·黑马程序员  
改变中国IT教育 我们正在行动





# 目录

传智播客·黑马程序员  
改变中国IT教育 我们正在行动

4.1

## 常用事件

[👉 点击查看本小节知识架构](#)

4.2

## 事件绑定与事件解绑

4.3

## 事件触发

[👉 点击查看本小节知识架构](#)


4.4

## 事件冒泡

[👉 点击查看本小节知识架构](#)



# 目录

 传智播客.黑马程序员  
改变中国IT教育 我们正在行动

4.5

## 事件对象

 [点击查看本小节知识架构](#)



## 4.1 常用事件

### 1. jQuery事件方法

在前端开发中，经常使用事件来完成一些交互操作。

举例：文本框获得焦点、鼠标滑过改变样式等。

jQuery提供的一些事件：

- ❑ 页面加载事件；
- ❑ 鼠标事件；
- ❑ 焦点事件等；



## 4.1 常用事件

### 1. jQuery事件方法

jQuery中事件方法一般与事件名称相同。

例如：单机事件click，对应的事件方法是click()方法。

常用事件方法：

分类	事件方法	描述
鼠标	click()	单机鼠标左键时触发
	dblclick()	双击鼠标左键时触发
键盘	keypress()	键盘按键(Shift、CapsLock等非字符键除外)被按下时触发
	keydown()	键盘按键被按下时触发



## 4.1 常用事件

### 1. jQuery事件方法

常用事件方法：

分类	事件方法	描述
焦点	keyup()	键盘按键被松开时触发
	onfocus()	获取焦点时触发
	onblur()	失去焦点时触发
改变	change()	元素的值发生改变时触发
其它	submit()	当表单提交时触发
	select()	当文本框(包括<input>和<textarea>中的文本被选中的时触发)
	scroll()	当滚动条发生变化时触发
	resize()	当调整浏览器窗口大小时触发



## 4.1 常用事件

### 1. jQuery事件方法

click()事件: 单击事件;

例如: 当单击按钮时就会触发单击事件, 执行事件处理函数。

单击事件

```
$('#btn').click(function() {  
    alert('我被单击了');  
});
```





## 4.1 常用事件

### 1. jQuery事件方法

click()事件：为元素绑定事件处理函数后，还可以手动触发事件。

例如：

```
$('#btn').click();
```

单击事件



## 4.1 常用事件

### 1. jQuery事件方法

多学一招：与JavaScript中的“DOM对象.onclick = function() {};”这种方式不同的是，jQuery允许为同一个对象的同一个事件绑定多个事件处理函数。

单击事件

```
$('#btn').click(function() {  
    console.log('text1');  
});  
$('#btn').click(function() {  
    console.log('text2');  
});
```

当触发事件时，其  
执行顺序与绑定时  
的顺序相同。



## 4.1 常用事件

### 2. 页面加载事件

在网页开发中，操作DOM时，会因为页面元素还没有加载而执行失败，因此引入事件处理函数；

解决办法：

□ jQuery提供了ready事件作为页面加载事件；

□ Javascript提供了onload页面加载事件；

区别：ready事件只需页面DOM元素加载完全后便可触发。

好处：提高页面的响应速度与onload相比ready事件语法比较灵活；



## 4.1 常用事件

### 2. 页面加载事件

语法: document参数可以省略, 写法2比较简洁。

对比

// 写法1

```
$(document).ready(function(  
    ) {  
    // 页面加载后要执行的代码  
});
```

// 写法2

```
$(function() {  
    // 页面加载后要执行的代码  
});
```



## 4.1 常用事件

### 2. 页面加载事件——一个页面编写多个onload事件

描述: 一个页面只能编写一个onload的事件，并且只能执行一次。

```
window.onload = function() {  
    console.log('text1');  
};  
window.onload = function() {  
    console.log('text2');  
};
```

执行结果只输出  
“text2”。



## 4.1 常用事件

### 2. 页面加载事件——一个页面编写多个ready事件

描述: 一个页面中可以包括多个ready事件，多个事件按照编写顺序依次执行。

```
$(function() {  
    console.log('text1');  
});  
$(function() {  
    console.log('text2');  
});
```

在控制台依次输出  
“text1” 和 “text2”



## 4.1 常用事件

多

学

一

招

### jQuery的load()方法

jQuery提供的另一个关于页面加载的方法——load()方法。load()方法会在元素的onload事件中绑定一个事件处理函数，对于不同的元素对象，事件触发的时机也不同。

对比

```
$('#img').load(function() {  
    // 元素内容加载完毕后要执行的  
    代码  
});
```

```
$(window).load(function() {  
    // 页面加载后要执行的代码  
});  
window.onload=function(){};
```

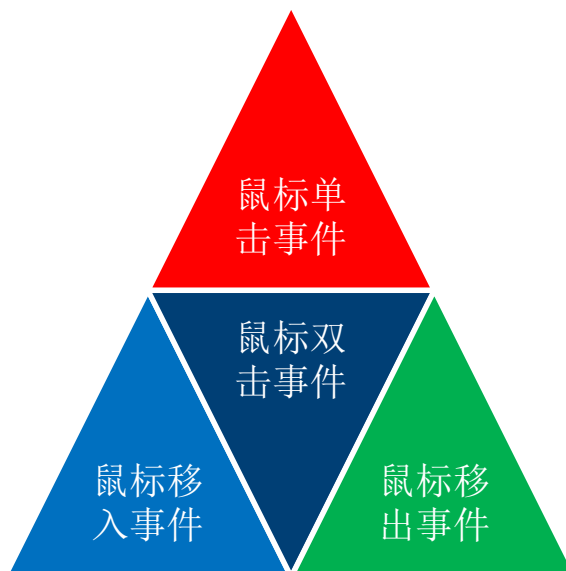


## 4.1 常用事件

### 3. 鼠标事件

鼠标事件是指：用户在点击鼠标或者移动鼠标时触发的事件。

jQuery中鼠标事件：



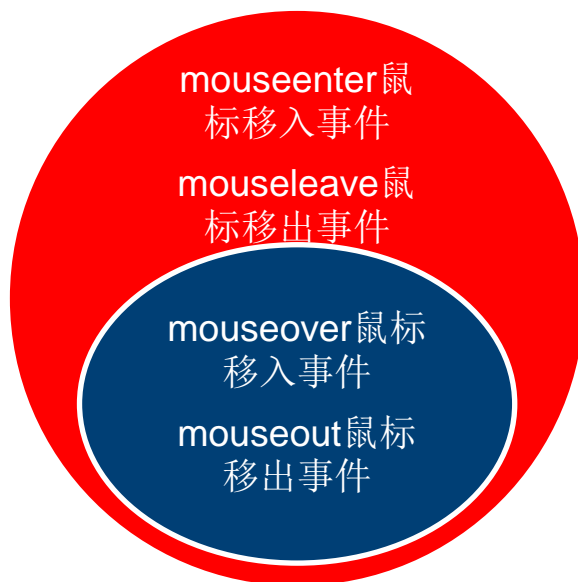




## 4.1 常用事件

### 3. 鼠标事件

区别: `mouseenter`和`mouseleave`触发事件的范围更小，要求鼠标必须移入或移出被选中的元素才可触发。





## 4.1 常用事件

### 3. 鼠标事件

事件	描述
click	单击鼠标左键时触发的事件
dblclick	双击鼠标左键时触发的事件
mousedown	按下鼠标时触发的事件
mouseup	松开鼠标时触发的事件
mouseover	鼠标指针移入被选元素或其子元素，都会触发的事件
mouseout	鼠标指针移出被选元素或任何子元素，都会触发的事件
mouseenter	鼠标指针移入被选元素时，才会触发的事件
mouseleave	鼠标指针移出被选元素时触发的事件

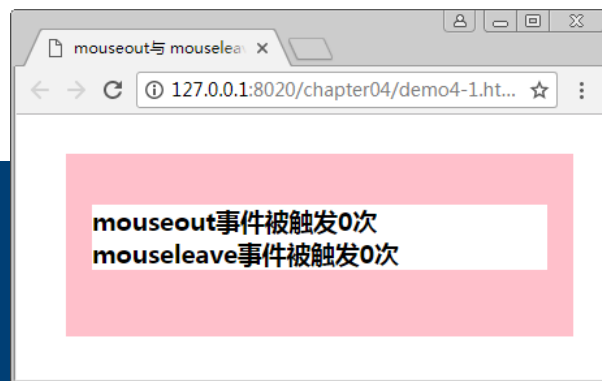


## 4.1 常用事件

### 3. 鼠标事件

#### 案例演示:

```
div {background-color: pink;padding: 20px;
margin: 30px;width: 350px;height: 100px;}
p {margin: 20px;padding: 10px;}
<div class="dv">
<h3 style="background-color:white;">
    mouseout事件被触发<span id="mOut">0</span>次</br>
    mouseleave事件被触发<span id="mLeave">0</span>次
</h3>
</div>
```



默认样式



## 4.1 常用事件

### 3. 鼠标事件

案例演示:

鼠标移入移出div父元素5次

```
var x = 0;  
var y = 0;  
$('div.dv').mouseout(function() {  
    $('#mOut').text(x += 1);  
});  
$('div.dv').mouseleave(function() {  
    $('#mLeave').text(y += 1);  
});
```





## 4.1 常用事件

### 3. 鼠标事件

案例演示：鼠标移出当前元素及其任意子元素时都会触发`mouseout`事件，而`mouseleave`事件只有鼠标移出当前元素时才会触发。





## 4.1 常用事件

多

学

一

招

### hover()方法处理鼠标移入和移出事件

jQuery中提供了hover()语法对鼠标的移入和移出操作进行处理。

语法：\$(selector).hover([over, out]);

#### hover()方法源代码

```
hover: function( fnOver, fnOut ) {  
    return this.mouseenter( fnOver ).mouseleave( fnOut ||  
        fnOver );  
}
```



## 4.1 常用事件

### 4. 焦点事件

jQuery中元素获得焦点时，触发 **focus** 事件，元素失去焦点时触发 **blur** 事件。

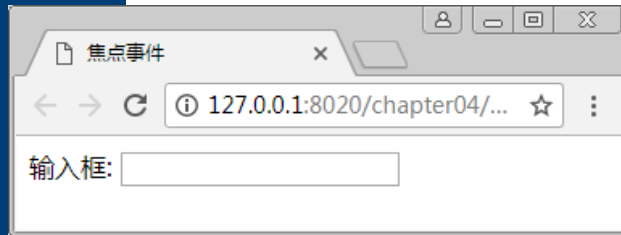
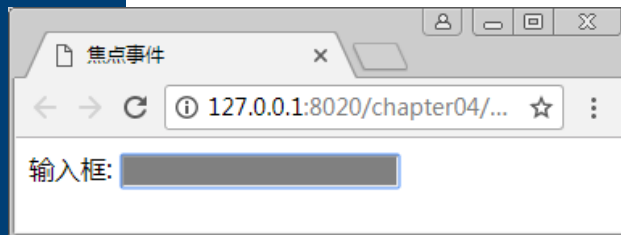
// 获得焦点

```
$('#input').focus(function() {  
    $('#input').css('background-color', 'gray');  
});
```

// 失去焦点

```
$('#input').blur(function() {  
    $('#input').css('background-color', 'white');  
});
```

focus(blur)





## 4.1 常用事件

### 5. 改变事件

jQuery提供的change()事件监控表单中元素内容是否改变。

仅适用于: input、textarea、select元素控件。

例如: 文本框、单选按钮、下拉列表等。





## 4.1 常用事件

### 5. 改变事件

案例演示：

默认样式

```
<p>昵称 : <input class="info" type="text"> </p>
```

```
<p>密码 : <input class="info" type="password"> </p>
```

```
<p>性别 :
```

```
<select class="info" name="male">
```

```
<option value="boy">男</option>
```

```
<option value="girl">女</option>
```

```
</select>
```

```
</p>
```

改变事件

昵称 :

密码 :

性别 : 男 ▼



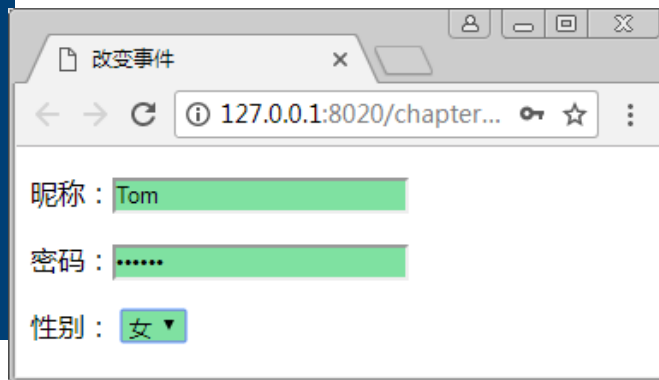
## 4.1 常用事件

### 5. 改变事件

案例演示: input内容发生改变的时候触发change()事件, 并添加背景色。

change()事件

```
<script>
// 注册改变事件
$('.info').change(function() {
    $(this).css('background-color', '#7FE1A1');
});
</script>
```





## 4.1 常用事件

### 6. 案例星级评分

网上购物订单完成后一般会显示评价功能最常见的评价方式就是星级评价。

#### 星级评价描述：

- ❑ 让卖家获取用户体验度是否良好，用户也能通过此功能表达自己的想法。
- ❑ 点亮的星星越多，表示用户的满意度越高店家获得的信誉度也就越高。

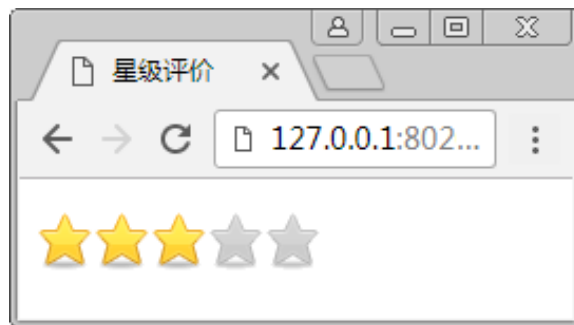
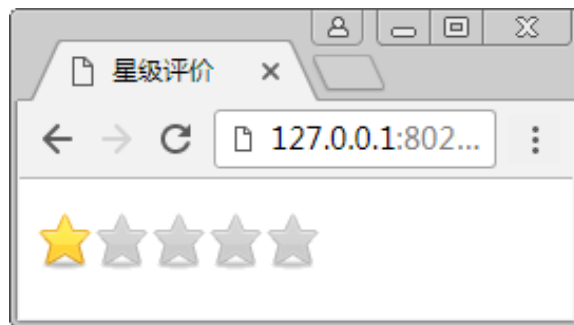


## 4.1 常用事件

### 6. 案例星级评分——案例展示

#### 案例演示步骤:

- ① 当移入某个星星，前面的星星都会被点亮；
- ② 当鼠标移出，星星将会变暗；
- ③ 单击某个星星后即完成评价；
- ④ 鼠标移出后被点击的星星前面的星星都会被点亮；
- ⑤ 后面的星星变暗；





## 4.1 常用事件

### 6. 案例星级评分——案例分析

#### 案例演示步骤:

① HTML结构;

② jQuery特效;



鼠标移入星星  
(mouseover)

- 为当前的li元素和它前面的li元素添加一个预定义的css样式“.light”实现点亮星星的效果。
- 同时删除当前的li元素后的所有li元素的“.light”样式，实现星星变暗的效果。

鼠标单击星星  
(click)

- 为被点击的当前li元素做一个标记，即设置属性light=on。
- 删除其它包含li元素的light属性(上次单击操作左的标记)。

鼠标移出星星  
(mouseout)

- 删除所有li元素的“.light”样式，实现星星变暗的效果。
- 找到包含light=on属性的li元素，将该元素和它前面的li元素都变成点亮效果

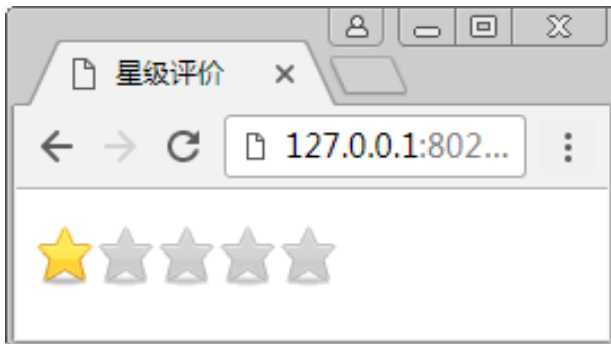


## 4.1 常用事件

### 6. 案例星级评分——案例实现

#### 案例演示步骤:

- ① 设计HTML结构与样式;
- ② 添加jQuery特效;

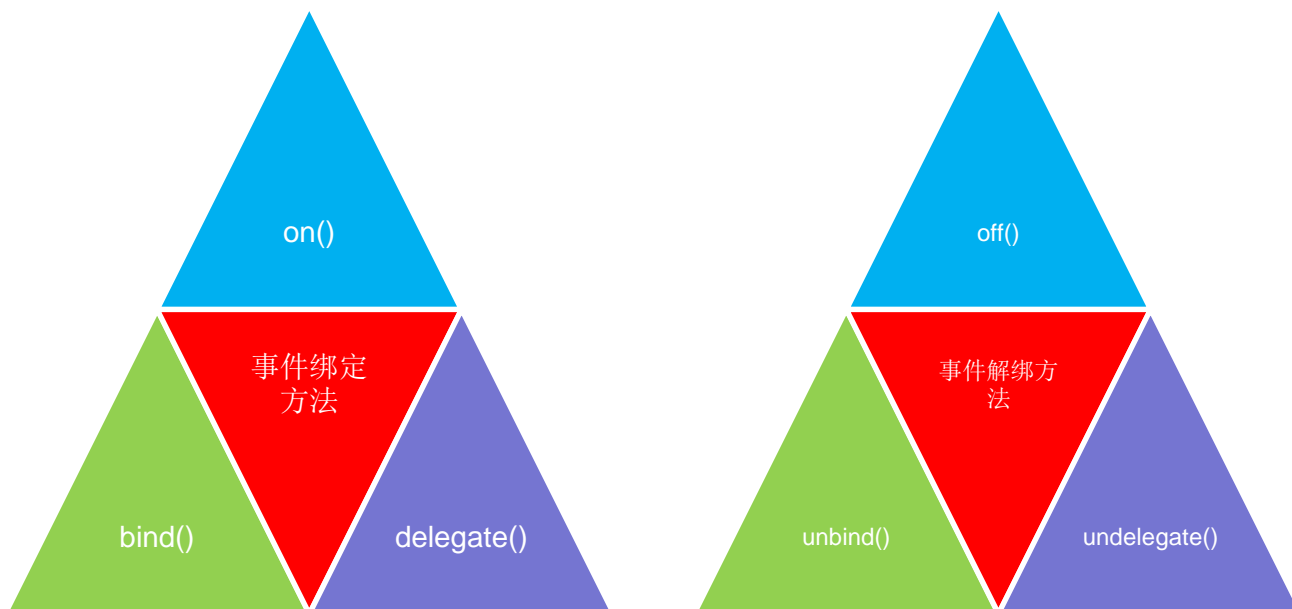




## 4.2 事件的绑定与解绑

### 1. 绑定事件

- 事件绑定方式，分别是on()、bind()和delegate()方法。
- 用off()、unbind()和undelegate()方法来解除绑定。





## 4.2 事件的绑定与解绑

### 1. 绑定事件

使用`on()`方法绑定事件不仅适用于当前元素，也适用于动态添加的元素。

语法：`$(selector).on(event, childSelector, data, function);`

参数	描述
event	必须。事件类型，如click、change、mouseover等
childSelector	可选。要绑定事件的一个或多个子元素
data	可选。传入事件处理函数的数据，可通过“事件对象.data”获取参数值
function	必须。事件被触发运行的事件处理函数





## 4.2 事件的绑定与解绑

### 1. 绑定事件——on方法绑定事件

on()方法在不设置任何可选参数时，表示为指定的元素绑定指定的事件。

```
$('#btn').on('click', function() {  
    alert('我被单击了');  
});
```

on()



## 4.2 事件的绑定与解绑

### 1. 绑定事件——多个事件使用相同的事件处理函数

当多个事件处理函数相同时可以利用on方法依次为多个事件绑定相同处理函数。

```
on(events, function(){})
```

```
$('#btn').on('mouseover mouseout', function() {  
    console.log('事件被触发');  
});
```



## 4.2 事件的绑定与解绑

### 1. 绑定事件——绑定多个事件

为on方法传递对象型的参数时，可以为指定元素绑定多个事件。

on({})

```
$('#btn').on({'click': function() {  
    alert('我被单击了');  
}, 'mouseover': function() {  
    $(this).css('backgroundColor', 'red');  
}});
```



## 4.2 事件的绑定与解绑

### 1. 绑定事件——事件委托

为on方法设置可选参数childSelector时，表示将子元素的事件委托给了父元素进行进行监控。

```
on(event,childSeletor,function(){}))
```

```
$('#dv').on('click', 'p', function() {  
    $(this).css('background-color', 'red');  
});
```



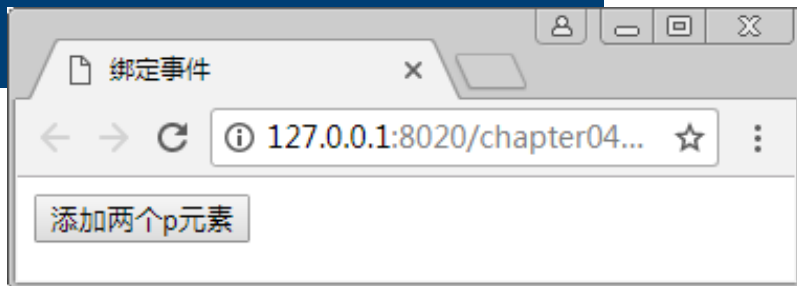
## 4.2 事件的绑定与解绑

### 1. 绑定事件——事件委托

案例演示：

示例

```
<input id="btn" type="button" value="添加两个p元素">  
<div id="dv"> </div>
```





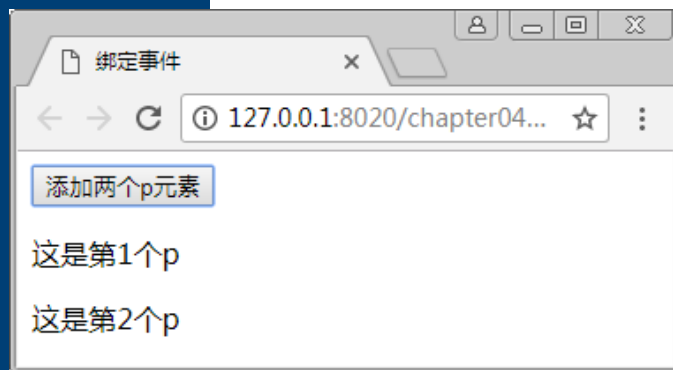
## 4.2 事件的绑定与解绑

### 1. 绑定事件——事件委托

案例演示：绑定click事件动态添加p元素。

示例

```
// 单击按钮通过on()方式为div中添加一个元素
$('#btn').on('click', function() {
    // 创建两个p标签，添加到div中
    $('<p>这是第1个p</p>').appendTo($('#dv'));
    $('<p>这是第2个p</p>').appendTo($('#dv'));
});
```





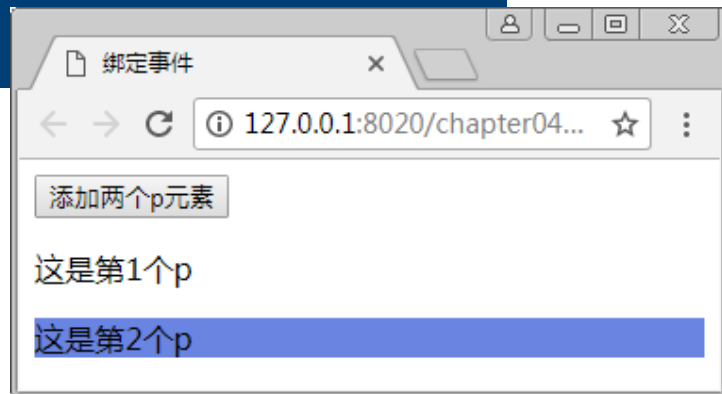
## 4.2 事件的绑定与解绑

### 1. 绑定事件——事件委托

案例演示：为动态添加p元素添加点击事件并点击后改变背景色。

事件委托

```
// 为div中p元素绑定事件
$('#dv').on('click', 'p', function() {
    $(this).css('background-color', '#6A84E1')
});
```





## 4.2 事件的绑定与解绑

传智播客·黑马程序员  
改变中国IT教育 我们正在行动

多

学

一

招

bind()和delegate()绑定方法

### 值得一提

值得一提的是，在jQuery中还有bind()和delegate()方法也可以完成事件绑定，但在jQuery新版本中已经由on()方法替代。





## 4.2 事件的绑定与解绑

传智播客·黑马程序员  
改变中国IT教育 我们正在行动

多

学

一

招

bind()和delegate()绑定方法

### 值得一提

bind()和delegate()这两个方法都是利用on()方法实现的。两者的区别在于，bind()方法将on()方法的第2个参数设为null，而delegate()方法传递了selector参数。



## 4.2 事件的绑定与解绑

传智播客·黑马程序员  
改变中国IT教育 我们正在行动

多

学

一

招

使用one()方法绑定单次事件

### 值得一提

使用jQuery绑定事件时，如果事件只需要在页面中执行一次，便可以使用one()方法绑定事件。该事件在触发一次后就会失效，不需要手动解绑事件。



## 4.2 事件的绑定与解绑

### 2. 事件解绑

元素上绑定的事件，在不需要使用时，可以利用jQuery提供的方法进行事件解绑。

**off()方法:** 适用于解绑事件方法（如click()事件）或on()方法绑定的事件。

示例

```
$(selector).off();           // 解绑匹配元素的所有事件
```

```
$(selector).off('click');    // 解绑匹配元素的指定事件，如click
```



## 4.2 事件的绑定与解绑

### 2. 事件解绑

off()方法还可以在第2个参数中传入特殊值 “\*\*”，用于解绑被委托的事件。

示例

```
// li元素将click事件委托给父元素ul监听
$('ul').on('click', 'li', function() {
    console.log('哈哈');
})
// 解绑ul元素上被委托的click事件
$('ul').off('click', '**');
```



## 4.2 事件的绑定与解绑

传智播客·黑马程序员  
改变中国IT教育 我们正在行动

多

学

一

招

### jQuery中的bind()和delegate()方法

#### 值得一提

对于jQuery中bind()方法和delegate()方法，jQuery也提供了相应解绑事件的方法unbind()和undelegate()，但在jQuery新版本中已经由off()方法替代。



## 4.2 事件的绑定与解绑

### 2. 事件解绑

unbind()和undelegate()方法的源码展示:

```
unbind: function( types, fn ) {  
    return this.off( types, null, fn );  
},  
undelegate: function( selector, types, fn ) {  
    // ( namespace ) or ( selector, types [, fn] )  
    return arguments.length === 1 ?  
        this.off( selector, "**" ) :  
        this.off( types, selector || "**", fn );  
}
```



## 4.2 事件的绑定与解绑

### 2. 事件解绑

案例演示:

```
div {width: 200px;height: 100px;  
      border: 1px solid black;}
```

```
p {background-color: #6A84E1;}
```

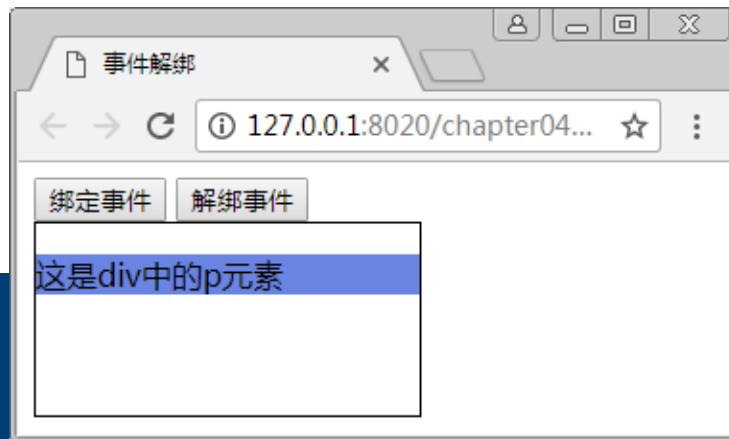
```
<input type="button" value="绑定事件" id="btn1">
```

```
<input type="button" value="解绑事件" id="btn2">
```

```
<div id="dv">
```

```
  <p>这是div中的p元素 </p>
```

```
</div>
```



默认页面



## 4.2 事件的绑定与解绑

### 2. 事件解绑

案例演示: 使用on()方法在div元素上注册单击事件, 然后使用delegate()方法为div元素的子元素p注册单击事件。

绑定事件

```
// 单击第1个按钮为div和div中p元素绑定单击事件
$('#btn1').click(function() {
    $('#dv').on('click', function() {alert('div元素被单击了'); });
    $('#dv').delegate('p', 'click', function() {alert('p元素被单击了');});
    alert('事件绑定成功');
});
```





## 4.2 事件的绑定与解绑

### 2. 事件解绑

案例演示: 无论是单击div元素或p元素都会弹出当前元素被单击的提示框。

127.0.0.1:8020 显示

div元素被单击了

确定

127.0.0.1:8020 显示

p元素被单击了

确定



## 4.2 事件的绑定与解绑

### 2. 事件解绑

点击解绑事件”按钮后，div元素和p元素的click事件都会被解绑。

分析：父元素调用off()方法，父元素和子元素上的相同事件都会被解绑。

需要注意：点击p元素后，其父元素上的click事件也会被触发。



## 4.2 事件的绑定与解绑

### 3. 案例动态添加和删除表格数据

- 添加和删除表格数据是开发后台管理系统的经常遇到的功能。
- 这些功能通常需要依赖脚本代码来实现一些动态操作数据的特效。

脚本代码：

- jQuery;
- Javascript;



## 4.2 事件的绑定与解绑

### 3. 案例动态添加和删除表格数据——案例展示

#### 案例演示：

① 默认页面：

② 删除数据：



课程名称	所属学院	操作
JavaScript	信息工程系	删除
css	信息工程系	删除
html	信息工程系	删除
jQuery	信息工程系	删除



课程名称	所属学院	操作
css	信息工程系	删除
html	信息工程系	删除
jQuery	信息工程系	删除



## 4.2 事件的绑定与解绑

### 3. 案例动态添加和删除表格数据——案例展示

案例演示：

③ 添加数据：

添加数据

X

课程名称：

所属学院：

添加

jQuery动态添加和删除

127.0.0.1:8020/chapter04/dynamicData/d...

添加数据

课程名称	所属学院	操作
css	信息工程系	删除
html	信息工程系	删除
jQuery	信息工程系	删除
VUE	信息工程系	删除

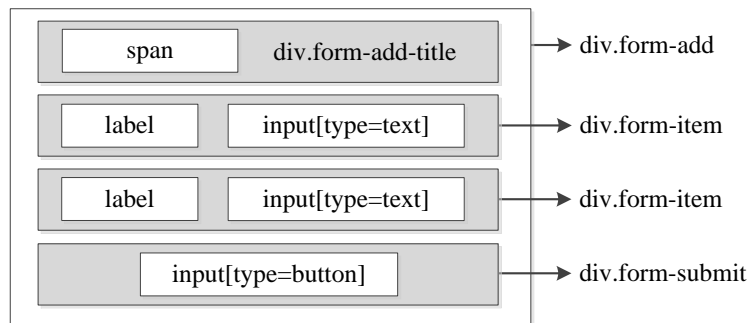
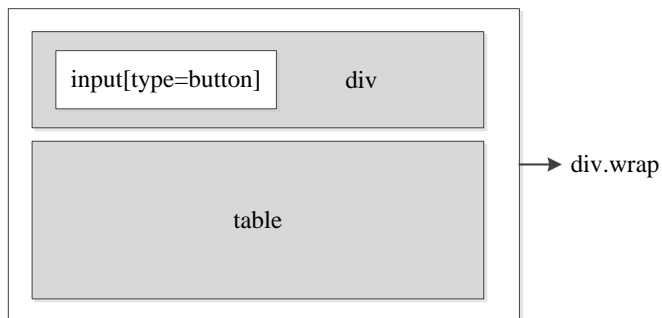


## 4.2 事件的绑定与解绑

### 3. 案例动态添加和删除表格数据——案例分析

#### 案例演示：

#### ① HTML结构：





## 4.2 事件的绑定与解绑

### 3. 案例动态添加和删除表格数据——案例分析

案例演示：

② jQuery特效：

添加数据

- 为“添加数据”按钮绑定单击事件，事件被触发后，
- 首先获取输入框中的课程和学院信息数据。
- 将获取到的数据拼接成一段HTML，在表格中添加一个行。
- 添加数据后隐藏遮罩层和弹出框，清空input输入框输入的课程信息。

删除数据

- 为表格中的“删除”链接绑定单击事件，
- 当某个“删除”的单击事件被触发，删除此“删除”按钮所属的当前行。



## 4.2 事件的绑定与解绑

### 3. 案例动态添加和删除表格数据——案例实现

#### 案例演示：

#### ① 设计HTML结构与样式：



**添加数据** X

课程名称：

所属学院：







## 4.2 事件的绑定与解绑

### 3. 案例动态添加和删除表格数据——案例实现

#### 案例演示：

#### ② 添加jQuery特效：

The image shows two browser windows side-by-side, both displaying a web application with a table of course data. The browser address bar shows the URL: 127.0.0.1:8020/chapter04/dynamicData/d... .

**Left Screenshot:** The table has four rows. The first row is 'JavaScript' with '信息工程系' (Information Engineering Department) and a '删除' (Delete) button. This '删除' button is circled in red. Below it are rows for 'css', 'html', and 'jQuery', each with a '删除' button.

**Right Screenshot:** The table has three rows. The first row is 'css' with '信息工程系' and a '删除' button. Below it are rows for 'html' and 'jQuery', each with a '删除' button. A red arrow points from the circled '删除' button in the left screenshot to the top of the table in the right screenshot, indicating the removal of the first row.

课程名称	所属学院	操作
JavaScript	信息工程系	删除
css	信息工程系	删除
html	信息工程系	删除
jQuery	信息工程系	删除

课程名称	所属学院	操作
css	信息工程系	删除
html	信息工程系	删除
jQuery	信息工程系	删除



## 4.2 事件的绑定与解绑

### 3. 案例动态添加和删除表格数据——案例实现

案例演示：

② 添加jQuery特效；

添加数据

X

课程名称：

所属学院：

添加

jQuery动态添加和删除

127.0.0.1:8020/chapter04/dynamicData/d...

添加数据

课程名称	所属学院	操作
JavaScript	信息工程系	删除
css	信息工程系	删除
html	信息工程系	删除
jQuery	信息工程系	删除

jQuery动态添加和删除

127.0.0.1:8020/chapter04/dynamicData/d...

添加数据

课程名称	所属学院	操作
css	信息工程系	删除
html	信息工程系	删除
jQuery	信息工程系	删除
VUE	信息工程系	删除



## 4.3 事件触发

### 1. 使用普通事件方法触发事件

案例演示: 单击事件触发后, id值为btn1的按钮会将背景色改为red。

单击事件

```
<input type="button" value="第1个按钮" id="btn1">  
<input type="button" value="第2个按钮" id="btn2">  
$('#btn1').click(function() {  
    $(this).css('backgroundColor', 'red');  
});
```



## 4.3 事件触发

### 1. 使用普通事件方法触发事件

案例演示：调用匹配元素`click()`事件来触发事件。

```
$('#btn2').click(function() {  
    $('#btn1').click();  
});
```

`click()`



## 4.3 事件触发

### 2. 使用trigger()方法触发事件

案例演示: 调用对象是待触发事件对象，参数是要触发的事件名称。

```
$('#btn2').click(function() {  
    $('#btn1').trigger('click');  
});
```

trigger()



## 4.3 事件触发

### 3. 使用triggerHandler()方法触发事件

案例演示：不会触发浏览器的默认行为。

例如：文本框获取焦点后的光标闪烁行为。

triggerHandler()

```
$('#btn2').click(function() {  
    $('#btn1').triggerHandler('click');  
});
```



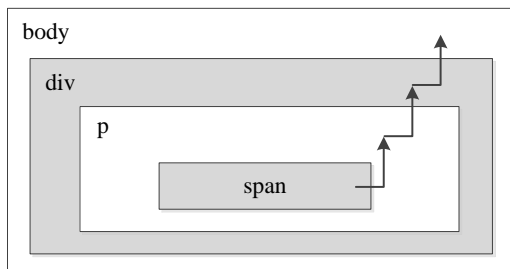
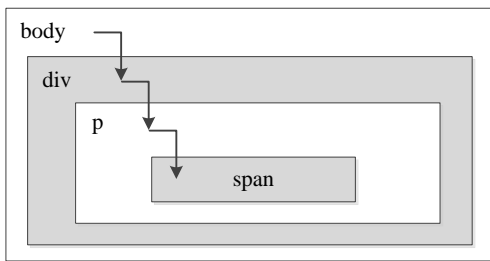
## 4.4 事件冒泡

### 1. 什么是事件冒泡

事件传播的两种策略：事件捕获和事件冒泡。

事件捕获：事件传播的顺序是最外层元素向触发事件的元素传递。

事件冒泡：事件传播顺序是从触发事件元素向最外层元素传递。





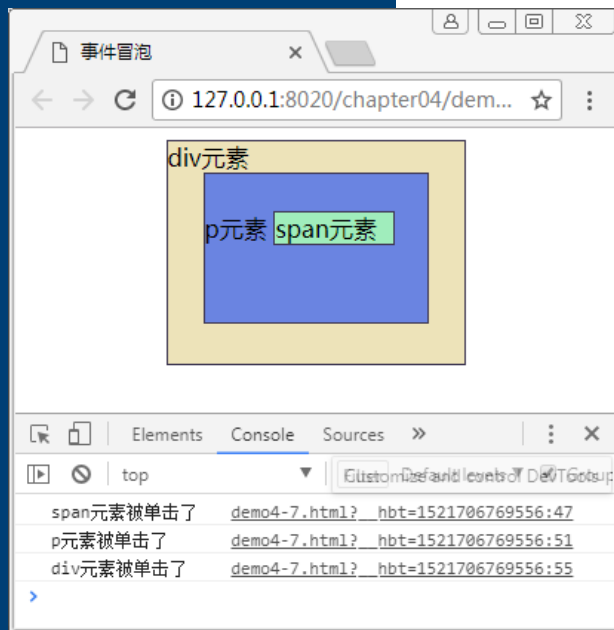
## 4.4 事件冒泡

### 2. 如何实现事件冒泡

案例演示：

默认页面

```
$('#span').click(function() {  
    console.log('span元素被单击了');  
});  
$('#p').click(function() {  
    console.log('p元素被单击了');  
});  
$('#div').click(function() {  
    console.log('div元素被单击了');  
});
```



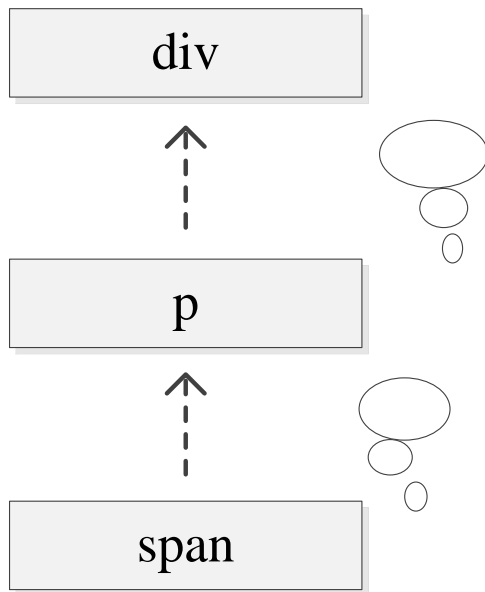




## 4.4 事件冒泡

### 2. 如何实现事件冒泡

案例演示：事件冒泡的事件传播的方向是从触发事件的元素到顶层。





## 4.4 事件冒泡

### 3. 如何阻止事件冒泡

单击子元素，会向上触发其父元素，祖先元素的单击事件。





## 4.4 事件冒泡

### 3. 如何阻止事件冒泡——使用event.stopPropagation()方法阻止事件冒泡

事件的使用会涉及到一个事件对象，即HTML DOM Event对象。

```
$('span').click(function(event) {  
    console.log('span元素被单击了');  
    event.stopPropagation();  
});
```

stopPropagation()



## 4.4 事件冒泡

### 3. 如何阻止事件冒泡——使用return false语句阻止事件冒泡

在span元素的单击事件程序完成之后，调用return false语句。

```
$('#span').click(function() {  
    console.log('span元素被单击了');  
    return false;  
});
```

return false



## 4.5 事件对象

### 1. 什么是事件对象

通过事件对象可以获取事件的状态。

举例：触发事件的元素、键盘按键的状态、鼠标的位置、鼠标按钮的状态。

封装

```
$('element').on('click', function(event) {  
    // event: 事件对象  
});
```



## 4.5 事件对象

### 2. jQuery事件对象的属性和方法

属性/方法	描述
type	事件类型，如果使用一个事件处理函数来处理多个事件，可以使用此种属性获得事件类型，比如click
data	传递给事件处理程序的额外数据(即事件方法中的data参数)
pageX/Y	鼠标事件中，鼠标相对于页面原点的水平/垂直坐标
keyCode	键盘事件中，键盘按键代码
delegateTarget	当前调用的jQuery事件处理程序的元素对象
currentTarget	当前触发事件的DOM对象，等同于this
target	事件源，也叫做事件触发者DOM对象，不一定等同于this
stopPropagation()	阻止事件冒泡
preventDefault()	阻止默认行为，例如a元素的href链接加载、表单提交以及click引起复选框的状态切换



## 4.5 事件对象

### 2. jQuery事件对象的属性和方法

注意

`delegateTarget`、`curentTarget`和`target`属性的使用。通过调用事件本身的方法（如`click()`）为元素绑定事件，并且不通过事件冒泡触发事件时，3个属性返回的元素是相同的。使用事件委托的方式（例如使用`on()`方法）绑定事件，或者通过事件冒泡触发事件时，3个属性返回的元素便有所不同。

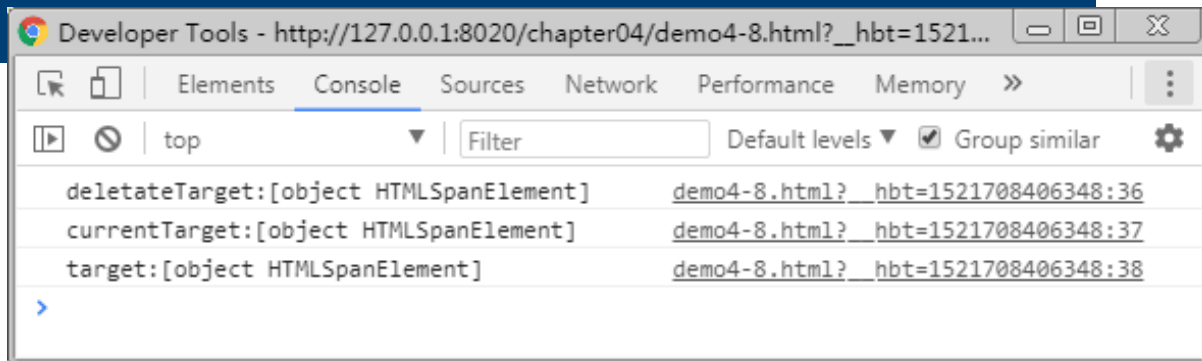


## 4.5 事件对象

### 3. 演示事件对象属性——使用普通方法绑定事件

```
$('span').click(function(event) {  
    console.log('deletateTarget:' + event.delegateTarget);  
    console.log('currentTarget:' + event.currentTarget);  
    console.log('target:' + event.target);  
});
```

delegateTarget/currentTarget/target





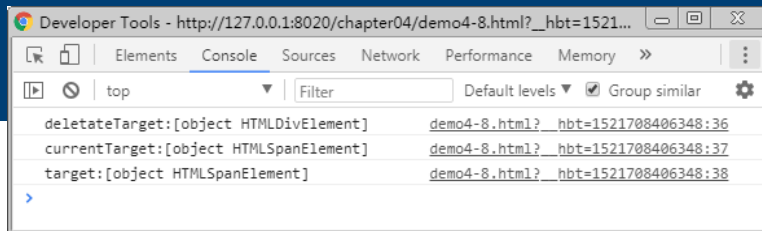


## 4.5 事件对象

### 3. 演示事件对象属性——通过事件委托绑定事件

delegateTarget/currentTarget/target

```
$('#div').on('click', 'span', function(event) {  
    console.log('deletateTarget:' + event.delegateTarget);  
    console.log('currentTarget:' + event.currentTarget);  
    console.log('target:' + event.target);  
});
```



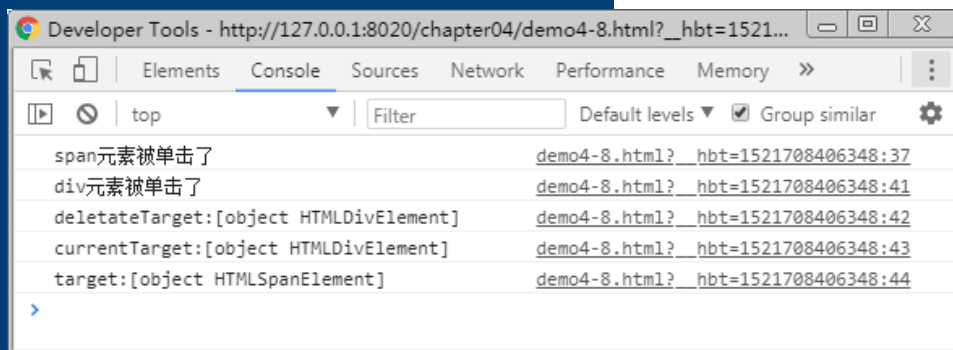


## 4.5 事件对象

### 3. 演示事件对象属性——通过事件冒泡触发事件

delegateTarget/currentTarget/target

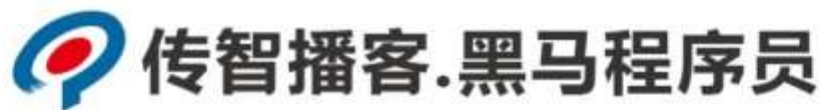
```
$('#span').click(function() {  
    console.log('span元素被单击了');  
});  
$('#div').click(function(event) {  
    console.log('div元素被单击了');  
    console.log('deletateTarget:' + event.delegateTarget);  
    console.log('currentTarget:' + event.currentTarget);  
    console.log('target:' + event.target);  
});
```





## 本章小结

介绍了jQuery的常用事件，阻止事件冒泡的方式以及事件对象的应用。  
熟悉jQuery事件触发机制和事件对象属性的用法，能够应用jQuery事件机制灵活的构建动态页面效果。



# Thank You!

[vx.boxuegu.com](http://vx.boxuegu.com)

