jQuery 事件

学习内容： jQuery 事件注册，jQuery 事件处理，jQuery 事件对象

1. jQuery 事件注册

语法：element.事件(function(){})

$(“div”).click(function(){ 事件处理程序 })

其他事件和原生基本一致。比如mouseover、mouseout、blur、focus、change、keydown、keyup、resize、scroll 等

2. jQuery 事件处理

2.1 事件处理 on() 绑定事件

**on()** 方法在匹配元素上**绑定一个或多个事件的事件处理函数**

语法：element.on(events,[selector],fn)

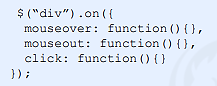
（1）. events:一个或多个用空格分隔的事件类型，如"click"或"keydown" 。

（2）. selector: 元素的子元素选择器 。

（3）. fn:回调函数 即绑定在元素身上的侦听函数。

**on() 方法优势1：**

**可以绑定多个事件，多个处理事件处理程序。**



**--🡪如果事件处理程序相同**



**on() 方法优势2：**

可以**事件委派**操作 。事件委派的定义就是，把原来加给子元素身上的事件绑定在父元素身上，就是把事件委派给父元素



在此之前有bind(), live() delegate()等方法来处理事件绑定或者事件委派，最新版本的请用on替代他们。

**on() 方法优势3：**

**动态创建的元素**，**click() 没有办法绑定事件， on() 可以给动态生成的元素绑定事件，给父元素添加事件，子元素也可以用；on ()**



|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>      <meta charset="UTF-8">      <meta name="viewport" content="width=device-width, initial-scale=1.0">      <meta http-equiv="X-UA-Compatible" content="ie=edge">      <title>Document</title>      <style>          div {              width: 100px;              height: 100px;              background-color: pink;          }            .current {              background-color: purple;          }      </style>      <script src="jquery.min.js"></script>  </head>  <body>      <div></div>      <ul>          <li>我们都是好孩子</li>          <li>我们都是好孩子</li>          <li>我们都是好孩子</li>          <li>我们都是好孩子</li>          <li>我们都是好孩子</li>      </ul>      <ol>      </ol>      <script>          $(function() {              // 1. 单个事件注册              // $("div").click(function() {              //     $(this).css("background", "purple");              // });              // $("div").mouseenter(function() {              //     $(this).css("background", "skyblue");              // });              // 2. 事件处理on              // (1) on可以绑定1个或者多个事件处理程序              // $("div").on({              //     mouseenter: function() {              //         $(this).css("background", "skyblue");              //     },              //     click: function() {              //         $(this).css("background", "purple");              //     },              //     mouseleave: function() {              //         $(this).css("background", "blue");              //     }              // });              $("div").on("mouseenter mouseleave", function() {                  $(this).toggleClass("current");              });              // (2) on可以实现事件委托（委派）              // $("ul li").click();              $("ul").on("click", "li", function() {                  alert(11);              });              // click 是绑定在ul 身上的，但是 触发的对象是 ul 里面的小li              // (3) on可以给未来动态创建的元素绑定事件              // $("ol li").click(function() {              //     alert(11);              // })              $("ol").on("click", "li", function() {                  alert(11);              })              var li = $("<li>我是后来创建的</li>");              $("ol").append(li);          })      </script>  </body>  </html> |

【案例 】发布微博案例

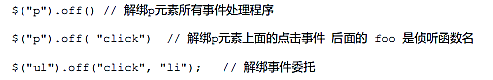
① 点击发布按钮， 动态创建一个小li，放入文本框的内容和删除按钮， 并且添加到ul 中。

② 点击的删除按钮，可以删除当前的微博留言。

|  |
| --- |
| <!DOCTYPE html>  <html>  <head lang="en">      <meta charset="UTF-8">      <title></title>      <style>          \* {              margin: 0;              padding: 0          }            ul {              list-style: none          }            .box {              width: 600px;              margin: 100px auto;              border: 1px solid #000;              padding: 20px;          }            textarea {              width: 450px;              height: 160px;              outline: none;              resize: none;          }            ul {              width: 450px;              padding-left: 80px;          }            ul li {              line-height: 25px;              border-bottom: 1px dashed #cccccc;              display: none;          }            input {              float: right;          }            ul li a {              float: right;          }      </style>      <script src="jquery.min.js"></script>      <script>          $(function() {              // 1.点击发布按钮， 动态创建一个小li，放入文本框的内容和删除按钮， 并且添加到ul 中              $(".btn").on("click", function() {                  var li = $("<li></li>");                  li.html($(".txt").val() + "<a href='javascript:;'> 删除</a>");                  $("ul").prepend(li);                  li.slideDown();                  $(".txt").val("");              })              // 2.点击的删除按钮，可以删除当前的微博留言li              // $("ul a").click(function() {  // 此时的click不能给动态创建的a添加事件              //     alert(11);              // })              // on可以给动态创建的元素绑定事件              $("ul").on("click", "a", function() {                  $(this).parent().slideUp(function() {                      $(this).remove();                  });              })          })      </script>  </head>  <body>      <div class="box" id="weibo">          <span>微博发布</span>          <textarea name="" class="txt" cols="30" rows="10"></textarea>          <button class="btn">发布</button>          <ul>          </ul>      </div>  </body>  </html> |

2.2 事件处理 off() 解绑事件

**off() 方法可以移除通过 on() 方法添加的事件处理程序**



**如果有的事件只想触发一次， 可以使用 one() 来绑定事件。只想让这个事件点击一次，某的事件只想在某种特殊情况下，才能停止用off()**

2.3 自动触发事件 trigger()

有些事件希望自动触发, 比如轮播图自动播放功能跟点击右侧按钮一致。可以利用定时器自动触发右侧按钮点击事件，不必鼠标 点击触发。

element.click() // 第一种简写形式

element.trigger("type") // 第二种自动触发模式

$("p").on("click", function () { alert("hi~"); });

$("p").trigger("click"); // 此时自动触发点击事件，不需要鼠标点击

element.triggerHandler(type) // 第三种自动触发模式

triggerHandler **模式不会触发元素的默认行为（表单获得焦点不会光标闪烁)，**这是和前面两种的区别。

3. jQuery 事件对象

事件被触发，就会有事件对象的产生。

**element.on(events,[selector],function(event) {})**

阻止默认行为：event.preventDefault() 或者 **return false**

阻止冒泡： **event.stopPropagation()**