**[js当中mouseover和mouseout多次触发(非冒泡)](http://blog.csdn.net/ltx851201/article/details/6800553)**

JS当中，mouseover和mouseout多次触发事件，不光是冒泡会产生，就是不冒泡，在一定条件下 ，也会产生多次触发事件；

例如下面的结构的情况下，在class="ceng\_up feature\_tips"这个层上加这两个事件，就会多次触发mouseover和mouseout而不断的交替事件。

|  |
| --- |
| <div id="tip\_popu">  <!--浮动层-->  <div class="ceng\_up feature\_tips" style="top:4px; left:9px;">  <div class="top\_feature"><div class="inner\_text"><img src="/themes/default/images/icon.gif"><a href="#" class="em\_a">武林英雄</a></div><div class="jishu">5426</div></div>  <div class="xiaceng">  <a class="c\_1" href="#"></a>  <a class="c\_2" href="#"></a>  <a class="c\_3" href="#" title="活动公告"></a>  <div class="benzhou">本周<font color="#FF0000">+5%</font></div>  </div>  </div>  <div style="top:35px; left:239px; " class="ceng\_down feature\_tips">  <div class="top\_feature"><div class="inner\_text"><img src="/themes/default/images/icon.gif"><a href="#" class="em\_a">武林英雄</a></div><div class="jishu">5426</div></div>  <div class="xiaceng">  <a href="#" class="c\_1"></a>  <a href="#" class="c\_2"></a>  <a title="活动公告" href="#" class="c\_3"></a>  <div class="benzhou">本周<font color="#3A9B00">-9%</font></div>  </div>  </div>  </div> |

开始是搞了好久，都没有找到适合的方法处理这个问题，最后终于在一个BLOG上看到这样的一个说法，终于把这个问题解决了；

下面是那个BLOG的原话，直接复制过来了，加的颜色区分一下：

在用到mouseover和mouseout事件来作为事件触发的条件，但是如果我们用做触发的元素内部有其他的元素的时候当鼠标移上的时候会反复的触发mouseover和mouseout事件。因为内部元素在鼠标移上的时候会向它的父对象派发事件，所以外面元素相当于也触发了mouseover 事件。

    为了阻止mouseover和mouseout的反复触发，这里要用到event对象的一个属性relatedTarget，这个属性就是用来判断 mouseover和mouseout事件目标节点的相关节点的属性。简单的来说就是当触发mouseover事件时，relatedTarget属性代表的就是鼠标刚刚离开的那个节点，当触发mouseout事件时它代表的是鼠标移向的那个对象。由于MSIE不支持这个属性，不过它有代替的属性，分别是 fromElement和toElement。

    有了这个属性，我们就能够清楚的知道我们的鼠标是从哪个对象移过来，又是要移动到哪里去了。这样我们就能够通过判断这个相关联的对象是否在我们要触发事件的对象的内部，或者是不是就是这个对象本身。通过这个判断我们就能够合理的选择是否真的要触发事件。

    这里我们还用到了一个用于检查一个对象是否包含在另外一个对象中的方法，contains方法。MSIE和FireFox分别提供了检查的方法，这里封装了一个函数。

|  |
| --- |
| function contains(parentNode, childNode) {  if (parentNode.contains) {  return parentNode != childNode && parentNode.contains(childNode);  } else {  return !!(parentNode.compareDocumentPosition(childNode) & 16);  }  } |

    这个函数用于检查一个对象是否被包含在我们的触发对象里面。

    下面就是我们的重点了，我封装了一个用于检查鼠标是否真正从外部移入或者移出对象的函数checkHover(e,target)，这个函数需要传入当前的事件对象和目标对象。

|  |
| --- |
| function checkHover(e,target){  if (getEvent(e).type=="mouseover") {  return !contains(target,getEvent(e).relatedTarget||getEvent(e).fromElement) && !((getEvent(e).relatedTarget||getEvent(e).fromElement)===target);  } else {  return !contains(target,getEvent(e).relatedTarget||getEvent(e).toElement) && !((getEvent(e).relatedTarget||getEvent(e).toElement)===target);  }  }function getEvent(e){  return e||window.event;  } |

    函数里面用到的getEvent()函数用于在MSIE或者FF下返回一个可用的event对象，这里你可以自己封装成别的函数。

    函数的逻辑很简单，首先判断事件的类型，这个主要是为了迁就MSIE，当是mouseover的时候relatedTarget在MSIE下应该是 fromElement，而mouseout则应该返回toElement，当然在FF下面就好办了，都是同一个属性relatedTarget。首先判断我们的relatedTarget是否在目标对象的内部，如果是的话则直接返回假如果不在内部的话则判断是否是目标对象本身，如果是的话返回假，要是两种情况都不成立则返回真。

    到这里我们的主要工作做完了，有了这个函数我们在进行编程的时候只要在mouseover或者mouseout事件内部先检查一下，再进行下一步操作就能轻松实现hover的效果。

|  |
| --- |
| myElement.onmouseover=function(e){  if(checkHover(e,this)){  do someting...  }  }  myElement.onmouseout=function(e){  if(checkHover(e,this)){  do someting...  }  } |

下面的代码，则是自己写的代码部分，也贴出来，供大家参考(基于**[jQuery](http://lib.csdn.net/base/jquery" \o "jQuery知识库" \t "_blank)**)：

|  |
| --- |
| //取得当前窗口的事件  var e = window.event;  $('.con\_yx ul li img').live(  'mouseover',  function(e) {  if(checkHover(e,this)){  var con\_yx = $('.con\_yx');  var index = con\_yx.index($(this).parent().parent().parent());// 100px;  $('#index\_val').val(index);  var currentspan = $(this).parent().find('span');// down  var position = $(this).offset();  if (currentspan.hasClass('up')) {  $(this).parent().find('img,.em\_a').clone().appendTo('.ceng\_up .inner\_text');  $('.ceng\_up').css('top', (position.top - 5));  $('.ceng\_up').css('left', (position.left - 5));  $('.ceng\_up').css('z-index',9999);  $('.ceng\_down').hide();  $('.ceng\_up').show();  } else {  $(this).parent().find('img,.em\_a').clone().appendTo('.ceng\_down .inner\_text');  $('.ceng\_down').css('top', (position.top - 5));  $('.ceng\_down').css('left', (position.left - 5));  $('.ceng\_down').css('z-index',9999);  $('.ceng\_up').hide();  $('.ceng\_down').show();  }  }  });  //鼠标移除层区域后，触发mouseout事件，把整个层隐藏  $('.ceng\_up,.ceng\_down').live('mouseout', function(e) {  if(checkHover(e,this)){  $(this).find('.inner\_text').html('');  $(this).hide();  }  });    /\*\*  \* 下面是一些基础函数，解决mouseover与mouserout事件不停切换的问题（问题不是由冒泡产生的）  \*/  function checkHover(e, target) {  if (getEvent(e).type == "mouseover") {  return !contains(target, getEvent(e).relatedTarget  || getEvent(e).fromElement)  && !((getEvent(e).relatedTarget || getEvent(e).fromElement) === target);  } else {  return !contains(target, getEvent(e).relatedTarget  || getEvent(e).toElement)  && !((getEvent(e).relatedTarget || getEvent(e).toElement) === target);  }  }  function contains(parentNode, childNode) {  if (parentNode.contains) {  return parentNode != childNode && parentNode.contains(childNode);  } else {  return !!(parentNode.compareDocumentPosition(childNode) & 16);  }  }  //取得当前window对象的事件  function getEvent(e) {  return e || window.event;  } |