[**JavaScript事件冒泡简介及应用**](http://www.cnblogs.com/webflash/archive/2009/08/23/1552462.html)

**一、什么是事件冒泡**

在一个对象上触发某类事件（比如单击onclick事件），如果此对象定义了此事件的处理程序，那么此事件就会调用这个处理程序，如果没有定义此事件处理程序或者事件返回true，那么这个事件会向这个对象的父级对象传播，从里到外，直至它被处理（父级对象所有同类事件都将被激活），或者它到达了对象层次的最顶层，即document对象（有些浏览器是window）。

打个比方说：你在地方法院要上诉一件案子，如果地方没有处理此类案件的法院，地方相关部门会帮你继续往上级法院上诉，比如从市级到省级，直至到中央法院，最终使你的案件得以处理。

**二、事件冒泡有什么作用**

（1）事件冒泡允许**多个操作被集中处理**（把事件处理器添加到一个父级元素上，避免把事件处理器添加到多个子级元素上），它还可以让你**在对象层的不同级别捕获事件**。

**【集中处理例子】**

[复制代码](javascript:void(0);)

<div onclick="eventHandle(event)" id="outSide" style="width:100px; height:100px; background:#000; padding:50px">  
<div id="inSide" style="width:100px; height:100px; background:#CCC"></div>  
</div>  
<script type="text/javascript">  
//本例子只在外面盒子定义了处理方法，而这个方法一样可以捕获到子元素点击行为并处理它。假设有成千上万子元素要处理，难道我们要为每个元素加“onclick="eventHandle(event)"”？显然没有这种集中处理的方法来的简单，同时它的性能也是更高的。  
function eventHandle(e)  
{  
    var e=e||window.event;  
    var obj=e.target||e.srcElement;  
    alert(obj.id+' was click')  
}  
</script>

[复制代码](javascript:void(0);)

（2）让**不同的对象同时捕获同一事件**，并调用自己的专属处理程序做自己的事情，就像老板一下命令，各自员工做自己岗位上的工作去了。

**【同时捕获同一事件例子】**

[复制代码](javascript:void(0);)

<div onclick="outSideWork()" id="outSide" style="width:100px; height:100px; background:#000; padding:50px">  
<div onclick="inSideWork()" id="inSide" style="width:100px; height:100px; background:#CCC"></div>  
</div>  
<script type="text/javascript">  
function outSideWork()  
{  
    alert('My name is outSide,I was working...');  
}  
  
function inSideWork()  
{  
    alert('My name is inSide,I was working...');  
}  
  
//因为下面程序自动激活单击事件，有些浏览器不允许，所以请单击灰色盒子，从这里开始下命令，这样因为冒泡的原因，黑色大盒子也会收到单击事件，并调用了自己的处理程序。如果还有更多盒子嵌套，一样道理。  
  
/\*  
function bossOrder()  
{  
    document.getElmentById('inSide').click();  
}  
bossOrder();  
\*/  
</script>

[复制代码](javascript:void(0);)

**三、需要注意什么**

●事件捕获其实有三种方式，事件冒泡只是其中的一种：（1）IE从里到外（inside→outside）的冒泡型事件。（2）Netscape4.0从外到里（outside→inside）的捕获型事件。（3）DOM事件流，先从外到里，再从里到外回到原点（outside→inside→outside）的事件捕获方法（似乎对象将触发两次事件处理，这有什么作用？鄙人不懂！）。

●不是所有的事件都能冒泡。以下事件不冒泡：blur、focus、load、unload。

●事件捕获方式在不同浏览器，甚至同种浏览器的不同版本中是有所区别的。如Netscape4.0采用捕获型事件解决方案，其它多数浏览器则支持冒泡型事件解决方案，另外DOM事件流还支持文本节点事件冒泡。

●事件捕获到达顶层的目标在不同浏览器或不同浏览器版本也是有区别的。在IE6中HTML是接收事件冒泡的，另外大部分浏览器将冒泡延续到window对象，即……body→documen→window。

●阻止冒泡并不能阻止对象默认行为。比如submit按钮被点击后会提交表单数据，这种行为无须我们写程序定制。

**四、阻止事件冒泡**

通常情况下我们都是一步到位，明确自己的事件触发源，并不希望浏览器自作聪明、漫无目的地去帮我们找合适的事件处理程序，即我们明确最精准目标，这种情况下我们不需要事件冒泡。另外通过对事件冒泡的理解，我们知道程序将做多较多额外的事情，这必然增大程序开销。还有一个重要的问题是：事件冒泡处理可能会激活我们本来不想激活的事件，导致程序错乱，甚至无从下手调试，这常成为对事件冒泡不熟悉程序员的棘手问题。所以必要时，我们要阻止事件冒泡。

**【不想激活的事件被激活例子】**

[复制代码](javascript:void(0);)

<div onclick="openWin('http://www.baidu.com')" id="outSide" style="width:100px; height:100px; background:#000; padding:50px">  
<div onclick="openWin('http://www.google.com')" id="inSide" style="width:100px; height:100px; background:#CCC"></div>  
</div>  
  
<script type="text/javascript">  
//本例你实际希望点击灰色盒子打开google首页，而点击黑色盒子打开baidu首页，但结果你点击灰色盒子的时候，却是同时打开了两个网页。其实在实际设计中较少遇到此问题，你可能会想如果我在页面不同DOM深处安置了不同的按钮或链接，深层处的事件触发会不会波及顶层的按钮呢？不会，因为按钮不能形成嵌套关系。  
function openWin(url)  
{  
    window.open(url);  
}  
</script>

[复制代码](javascript:void(0);)

下面是本人在网上抄的一个方法，把这个方法放在精准目标对象处理程序结尾，本事件触发处理结束后，事件将不在进行冒泡处理。

**【阻止事件冒泡例子】**

[复制代码](javascript:void(0);)

<div onclick="showMsg(this,event)" id="outSide" style="width:100px; height:100px; background:#000; padding:50px">  
<div onclick="showMsg(this,event)" id="inSide" style="width:100px; height:100px; background:#CCC"></div>  
</div>  
<script type="text/javascript">  
//阻止事件冒泡后，你点击灰色盒子，整个过程只弹一次对话框了（注意与默认情况对比）  
function showMsg(obj,e)  
{  
    alert(obj.id);  
    stopBubble(e)  
}  
  
//阻止事件冒泡函数  
function stopBubble(e)  
{  
    if (e && e.stopPropagation)  
        e.stopPropagation()  
    else  
        window.event.cancelBubble=true  
}  
</script>