**IE事件对象（The Internet Explorer Event Object）**

作者： 字体：[[增加](javascript:turnbig()) [减小](javascript:turnsmall())] 类型：转载

不同于DOM事件对象，基于Event Handler授权这种方式，IE事件对象可以用不同的方式进行访问。当一个事件Handler通过DOM 0 级的方式被授权，则这个事件对象将作为window对象的属性而存在

看下面的例子，

复制代码 代码如下:

　　var btn ＝ document.getElementById('mybtn');   
　　btn.onclick = function(){   
　　　　var event = window.event;   
　　　　alert(event.type);//"click"   
　　}

　　此时，event对象来源于window.event对象，并且之后又用于确定事件类型。然而，当eventHandler 通过attachEvent()这种方式来授权的话，event 对象将作为function的唯一参数，请看如下代码

复制代码 代码如下:

var btn = document.getElementById("myBtn");   
btn.attachEvent("onclick", function(event){   
alert(event.type); //"click"   
});

　　当使用attachEvent() 方法时，事件对象同时也可以在window对象上进行访问，和dom 0级的实现方式一样，并且事件对象同样被传进来作为一个参数。   
　　如果event Handler 是通过HTML属性被授权的，event是作为一个一个变量被访问的，叫做event。例如

复制代码 代码如下:

<input type="button" value="Click Me" onclick="alert(event.type)">

　　IE 事件对象同样包含属性和方法，这与创建该特定事件有关。这些属性和方法，要么直接映射到DOM的属性和方法，要么与DOM的属性和方法相关。象DOM事件对象的属性和方法回由于出发事件的的不同而不同，但共有的方法如下：   
　　cancleBubble 布尔 可读/可写 默认值时false，但可以被设置成true来取消事件冒泡，与dom中的 stopPropagation()方法相同。   
　　returnValue 布尔 可读/可写 默认值是true，当设置成false时用以取消事件的默认行为 与dom中的preventDefault（）相同。   
　　srcElement　 元素 只读　　　事件的目标，与dom中的target属性相同。   
　　type　　　　　字符串 只读 被触发的事件类型。   
　　因为event handler的事件作用域是由授权方式决定的，所以this的值不应该总指向这个事件的目标，所以用event.SrcElement 来代替。例子如下

复制代码 代码如下:

btn.onclick = function(){   
alert(window.event.srcElement === this); //true   
};   
btn.attachEvent("onclick", function(event){   
alert(event.srcElement === this); //false   
});

　　在第一个event handler中通过DOM0级的方式来授权，所以srcElement属性指向this，但在第二个 event handler中这两个值时不同的。   
　　returnValue属性等同于DOM中的preventDefault（）方法，同样是用来取消事件的默认行为，你需要将returnValue的属性值设置成false来取消事件的默认动作，请看如下例子；

复制代码 代码如下:

var link ＝ document.getElementById('myLink');   
link.onclick = function(){   
window.event.returnValue = false;   
};

　　在这个例子中，用returnValue属性来取消连接的默认行为。与dom不同的是，没有方法能够决定一个事件是否可以被取消或者禁用javascript。   
　　cancleBubble属性与都没中stopPropagation（）的功能相同，阻止事件的冒泡。因为IE8或者更早的IE版本，不支持事件捕捉阶段，而且cancleBubble只支持冒泡的取消，而stopPropagation（）是取消事件的捕捉和冒泡。例如：

复制代码 代码如下:

var btn = document.getElementById("myBtn");   
btn.onclick = function(){   
alert("Clicked");   
window.event.cancelBubble = true;   
};   
document.body.onclick = function(){   
alert("Body clicked");   
};

　　通过爱 onclick 这个handler中设置cancelBubble的值为true，他阻止了事件冒泡至document.body 的事件处理，因此当btn被点击的时候，结果只会弹出一个提示对话框，即"click"。