我们给一个dom同时绑定两个点击事件，一个用捕获，一个用冒泡，会执行几次事件，会先执行冒泡还是捕获？

在这里，我先补充一点小知识，addEventListener() 方法用于向指定元素添加事件句柄。element.addEventListener(event, function, useCapture)，

其中的event，必须。字符串，指定事件名。 不要使用 "on" 前缀。 例如，使用 "click" ,而不是使用 "onclick"。

function  必须。指定要事件触发时执行的函数

useCapture  可选。布尔值，指定事件是否在捕获或冒泡阶段执行。

因为我们在下面的例子中就是用addEventListener来添加句柄的，所以在这里简单介绍一些相关的知识。但是这里有个兼容性问题，Internet Explorer 8 及更早IE版本不支持 addEventListener() 方法，，Opera 7.0 及 Opera 更早版本也不支持。 但是，对于这些不支持该函数的浏览器，你可以使用 attachEvent() 方法来添加事件句柄。

对于兼容写法，如下：

var x = document.getElementById("myBtn");

if (x.addEventListener) {                    //所有主流浏览器，除了 IE 8 及更早 IE版本

    x.addEventListener("click", myFunction);

} else if (x.attachEvent) {                  // IE 8 及更早 IE 版本

    x.attachEvent("onclick", myFunction);

}

好了，正式进入冒泡和捕获的话题。那么什么是冒泡，什么是捕获？我们先来简单的了解一下这两个概念。

<style>

#div1{  
width: 400px;  
height: 400px;  
background: #f00;  
}  
#div2{  
width: 300px;  
height: 300px;  
background: #0f0;  
}  
#div3{  
width: 200px;  
height: 200px;  
background: #00f;  
}  
#div4{  
width: 100px;  
height: 100px;  
background: #f0f;  
}

</style>

<div id="div1">我是div1  
<div id="div2">我是div2  
<div id="div3">我是div3  
<div id="div4">我是div4</div>  
</div>  
</div>  
</div>

var div1=document.getElementById("div1");

var div2=document.getElementById("div2");

var div3=document.getElementById("div3");

var div4=document.getElementById("div4");

div1.addEventListener("click",function(){

alert("我是div1");

})

div2.addEventListener("click",function(){

alert("我是div2");

})

div3.addEventListener("click",function(){

alert("我是div3");

})

div4.addEventListener("click",function(){

alert("我是div4");

})

那么，当我们点击div1的时候，只弹出“我是div1”，但是当我们点击div2的时候，就先弹出“我是div2”，再弹出“我是div1”，当我们点击div4 的时候，则是4-3-2-1这样的顺序，这样就叫做冒泡，它就像鱼儿吐泡泡一样，从下到上，泡泡从水下上来不正是越到上面越泡泡就越大的吗，那么从这个角度思考，先从子元素开始执行，然后是父元素，再然后是祖先元素，他们的等级不也是越来越大的吗。因为之前在addEventListener里面说了，false是默认的，表示的就是冒泡。那么捕获又是什么呢，我们在上面代码中的添加进第三个参数true 以后，再运行一下代码，会发现这时，我们点击div4的时候，出现的顺序是1-2-3-4，它是从祖先元素开始慢慢找，最后找到我们的点击目标，这个行为不就像警察叔叔抓坏人一样，逐渐的缩小抓捕范围，最后确定到某一个人身上，所以这个叫做捕获。

那么，如果一个dom元素中，既有冒泡，又有捕获的话，会先执行哪个呢？w3c规定了，任何发生在w3c事件模型中的事件，首是进入捕获阶段，直到达到目标元素，再进入冒泡阶段。这句话怎么理解，我们举个例子就明白了。我们简单的修改一下代码:

div1.addEventListener("click",function(){  
alert("div1");  
},false);  
div2.addEventListener("click",function(){  
alert("div2");  
},true);  
div3.addEventListener("click",function(){  
alert("div3");  
},false);  
div4.addEventListener("click",function(){  
alert("div4");  
},true);

这时的div1和div3是冒泡事件，div2和div4是捕获事件，那么我们点击div4以后，弹出的顺序是2-4-3-1，为什么这样呢？因为我们先执行捕获过程，看看这个例子中有哪几个是捕获的，有div2和div4，那么捕获又是从大到小，所以，先弹出div2，再弹出div4，捕获结束以后就该是冒泡了，那么冒泡的顺序呢？从小到大，从子到父，所以就先弹出div3，再就是div1，所以最后的顺序是2-4-3-1.那么问题又来了，要是我们点击div3，会有什么样的效果呢，结果是2-3-1，同样的道理，先捕获，那么捕获是从div1开始到div3 的，这中间只有div2是捕获，div4并没有执行到，因为我们点击的目标是div3，后面的步骤和前面说过的一样，先3后1.

那么最后一个问题来了，如果我们再修改一下代码：

div1.addEventListener("click",function(){  
alert("div1");  
},false);  
div2.addEventListener("click",function(){  
alert("div2\_捕获");  
},true);  
div2.addEventListener("click",function(){  
alert("div2\_冒泡");  
},false);

那么，当我们点击div2的时候，顺序是怎么样的呢，结果是 div2\_捕获  -> div\_2冒泡 -> div1这样的顺序，我们再修改一下代码：

div1.addEventListener("click",function(){  
alert("div1");  
},false);  
div2.addEventListener("click",function(){  
alert("div2\_冒泡");  
},false);  
div2.addEventListener("click",function(){  
alert("div2\_捕获");  
},true);

那么，当我们点击div2的时候，顺序是怎么样的呢，结果是 div2\_冒泡  -> div\_2捕获 -> div1这样的顺序

所以，综合以上的几个例子，我们可以得出这么一个结论：

绑定在被点击元素的事件是按照代码的顺序发生的，其他非绑定的元素则是通过冒泡或者捕获的触发。按照W3C的标准，先发生捕获事件，后发生冒泡事件。所以事件的整体顺序是：非目标元素捕获 -> 目标元素代码顺序 -> 非目标元素冒泡。