

课前补充：

Js事件的三个阶段：捕获，目标，冒泡

**捕获：**

事件从根节点流向目标节点，途中流经各个DOM节点，在各个节点上触发捕获事件，直到达到目标节点。

问题：那么捕获阶段有什么用吗？完全可以没有捕获阶段也可以啊

捕获阶段的主要任务是建立传播路经，在冒泡阶段根据这个路经回溯到文档根节点

**目标：**

事件到达目标节点时，就到了目标阶段，事件在目标节点上被触发

**冒泡：**

事件在目标节点上触发后，不会终止，一层层向上冒，回溯到根节点

Case 1:

Html部分：

<div id="parent" style="width:200px;height:200px;background:blue;">

<div id="son1" style="width:100px;height:100px;background:red;"></div>

<div id="son2" style="width:100px;height:100px;background:yellow"></div>

</div>

Javascript部分：

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

var son1 = document.getElementById("son1");

var son2 = document.getElementById("son2");

parent.addEventListener("click",function(event){

alert(1);

// event.stopPropagation();

},true)

son1.addEventListener("click",function(event){

// event.stopPropagation();

alert(2)

},true)

son2.addEventListener("click",function(event){

alert(3);

},true)

当参数为true时，为捕获阶段。点击子元素，会从最外面开始触发，一直到作用事件的元素为止。

当参数为false时，为冒泡阶段，点击子元素，从作用事件元素开始，一直触发到最外面，一般是到window。

如果想阻止捕获或者冒泡，只需要写e.stopPropagation()即可。

注意，无论阻止冒泡还是阻止捕获，都是用e.stopPropagation()

Case2:

Html:

<li id="aa">

<a href="#" style="color:rgb(240,200,160)" class="bb">Stop propogation </a>

<a href="#" style="color:rgb(240,200,160)" class="cc">我的账户</a>

</li>

Js:

li.addEventListener("click",function(event){

if(event.target.tagName.toLowerCase()==="li"){

console.log(event.target,1);

}else if(event.target.tagName.toLowerCase()==="a"){

console.log(event.target,3);

}

console.log(event.currentTarget,2);

},false)

for(var i = 0; i<li.children.length;i++){

li.children[i].addEventListener("click",function(event){

event.stopPropagation();

})

}

// for(var i = 0;i<li.children.length;i++){

// li.children[i].addEventListener("click",function(event){

// event.stopPropagation();

// })

// }

引申问题：event.target和event.currentTarget的区别？

1. target在事件流的目标阶段；currentTarget在事件流的捕获，目标及冒泡阶段。   
   2、event.currentTarget指向事件所绑定的元素，而event.target始终指向事件发生时的元素。   
   3、事件真正的发送者是evt.target（而且是可以变的，根据触发的不同displayObject），注册侦听器的是evt.currentTarget（不会变的）。   
   4、在 Google 浏览器内，同时拥有 srcElement 和 target 这两个标签，这两个标签本质相同，但在 FireFox 浏览器里面却没有 srcElement 标签，如果使用了 srcElement 标签，将导致不兼容 FireFix 浏览器，所以别被用错一个标签导致浏览器不兼容给坑了。