**[JS阻止冒泡和取消默认事件(默认行为)](http://caibaojian.com/javascript-stoppropagation-preventdefault.html)**

2018-09-0718522View0

js冒泡和捕获是事件的两种行为，使用event.stopPropagation()起到阻止捕获和冒泡阶段中当前事件的进一步传播。使用event.preventDefault()可以取消默认事件。对于冒泡和捕获的优先顺序请看之前文章：[JavaScript捕获和冒泡探讨](http://caibaojian.com/javascript-capture-bubble.html)

**防止冒泡和捕获**

**w3c的方法是e.stopPropagation()，IE则是使用e.cancelBubble = true**

stopPropagation也是事件对象(Event)的一个方法，作用是阻止目标元素的冒泡事件，但是会不阻止默认行为。什么是冒泡事件？如在一个按钮是绑定一个”click”事件，那么”click”事件会依次在它的父级元素中被触发 。stopPropagation就是阻止目标元素的事件冒泡到父级元素。如：

<div id='div' onclick='alert("div");'>

<ul onclick='alert("ul");'>

<li onclick='alert("li");'>test</li>

</ul>

</div>

上面的代码，Demo如下,我们单击test时，会依次触发alert(“li”),alert(“ul”),alert(“div”)，这就是事件冒泡。

* 冒泡事件

阻止冒泡

window.event? window.event.cancelBubble = true : e.stopPropagation();

* 停止冒泡

**取消默认事件**

**w3c的方法是e.preventDefault()，IE则是使用e.returnValue = false;**

preventDefault它是事件对象(Event)的一个方法，作用是取消一个目标元素的默认行为。既然是说默认行为，当然是元素必须有默认行为才能被取消，如果元素本身就没有默认行为，调用当然就无效了。什么元素有默认行为呢？如链接<a>，提交按钮<input type=”submit”>等。当Event 对象的 cancelable为false时，表示没有默认行为，这时即使有默认行为，调用preventDefault也是不会起作用的。

我们都知道，链接<a>的默认动作就是跳转到指定页面，下面就以它为例，阻止它的跳转：

//假定有链接<a href="http://caibaojian.com/" id="testA" >caibaojian.com</a>

var a = document.getElementById("testA");

a.onclick =function(e){

if(e.preventDefault){

e.preventDefault();

}else{

window.event.returnValue == false;

}

}

演示：阻止链接跳转的默认行为  
[caibaojian.com](http://caibaojian.com/)

**return false**

javascript的return false只会阻止默认行为，而是用[jQuery](http://caibaojian.com/jquery/" \o "jQuery手册)的话则既阻止默认行为又防止对象冒泡。

下面这个使用原生js，只会阻止默认行为，不会停止冒泡

<div id='div'  onclick='alert("div");'>

<ul  onclick='alert("ul");'>

<li id='ul-a' onclick='alert("li");'><a href="http://caibaojian.com/"id="testB">caibaojian.com</a></li>

</ul>

</div>

var a = document.getElementById("testB");

a.onclick = function(){

return false;

};

演示：阻止链接默认行为，没有停止冒泡

* [caibaojian.com](http://caibaojian.com/)

下面这个是使用[jQuery](http://caibaojian.com/t/jquery" \o "View all posts in jQuery" \t "_blank)，既阻止默认行为又停止冒泡

<div id='div'  onclick='alert("div");'>

<ul  onclick='alert("ul");'>

<li id='ul-a' onclick='alert("li");'><a href="http://caibaojian.com/"id="testC">caibaojian.com</a></li>

</ul>

</div>

$("#testC").on('click',function(){

return false;

});

演示：既停止冒泡又阻止默认行为

* [caibaojian.com](http://caibaojian.com/)

**总结使用方法**

当需要停止冒泡行为时，可以使用

function stopBubble(e) {

//如果提供了事件对象，则这是一个非IE浏览器

if ( e && e.stopPropagation )

//因此它支持W3C的stopPropagation()方法

e.stopPropagation();

else

//否则，我们需要使用IE的方式来取消事件冒泡

window.event.cancelBubble = true;

}

当需要阻止默认行为时，可以使用

//阻止浏览器的默认行为

function stopDefault( e ) {

//阻止默认浏览器动作(W3C)

if ( e && e.preventDefault )

e.preventDefault();

//IE中阻止函数器默认动作的方式

else

window.event.returnValue = false;

return false;

}

**事件注意点**

* 1. event代表事件的状态，例如触发event对象的元素、鼠标的位置及状态、按下的键等等；
  2. event对象只在事件发生的过程中才有效。

firefox里的event跟IE里的不同，IE里的是全局变量，随时可用；firefox里的要用参数引导才能用，是运行时的临时变量。  
在IE/Opera中是window.event，在Firefox中是event；而事件的对象，在IE中是window.event.srcElement，在Firefox中是event.target，Opera中两者都可用。

* 1. 下面两句效果相同：

function a(e){

var e = (e) ? e : ((window.event) ? window.event : null);

var e = e || window.event; // firefox下window.event为null, IE下event为null

}