06-JS事件详解

JS事件详解

事件流

事件处理

事件对象

一、事件流

1. 事件流

描述的是在页面中接受事件的顺序

1. 事件冒泡

由最具体的元素接受，然后逐级向上传播至最不具体的元素的节点（文档）

1. 事件捕获

最不具体的节点先接受事件，而最具体的节点应该是最后接受事件

二、事件处理

1．HTML事件处理：

直接添加到HTML结构中

2．DOM0级事件处理

把一个函数赋值给一个事件处理程序属性

1. DOM2级事件处理

addEventListener（”事件名”,”事件处理函数”,”布尔值”）;

true:事件捕获

false：事件冒泡

removeEventListener();

1. IE事件处理程序

attachEvent

detachEvent

eg1:

<button id="btn" onclick="demo()">Go</button>

<script type="text/javascript">

function demo(){

alert("hello");

}

</script>

坏处在于修改一处demo后要在另一处也修改

可以利用DOM0级事件处理。

eg2：

<button id="btn">Go</button>

<script type="text/javascript">

var btn1=document.getElementById("btn");

btn1.onclick=function(){alert("hello,这是DOM0级事件处理程序")};

btn1.onclick=null；//设置为空，可以清楚当前事件

</script>

eg3：

<button id="btn">Go</button>

<script type="text/javascript">

var btn1=document.getElementById("btn");

btn1.onclick=function(){alert("hello,这是DOM0级事件处理程序1")};//会被覆盖

btn1.onclick=function(){alert("hello,这是DOM0级事件处理程序2")};

btn1.onclick=null；//设置为空，可以清楚当前事件

</script>

如上程序一会被覆盖掉

如果有多个事件，只会显示最后一个事件，前面的事件会被覆盖掉

DOM2级事件

<buuton id="btn2">Goo</buuton>

<script type="text/javascript">

var btn1=document.getElementById("btn2").addEventListener("click",demo);

function demo(){

alert("DOM2级事件处理程序");

}

</script>

或者

<buuton id="btn2">Goo</buuton>

<script type="text/javascript">

var btn2=document.getElementById("btn2");

btn2.addEventListener("click",demo1);

btn2.addEventListener("click",demo2);

function demo1(){

alert("DOM2级事件处理程序1");

}

function demo2(){

alert("DOM2级事件处理程序2");

}

</script>

两个程序会依次弹出，DOM2级事件不会相互覆盖。

利用btn2.removeEventListener（”click”,demo2）;来移除事件。

兼容各种浏览器用法：

IE8或IE8以下浏览器只能使用attachEvent和detachEvent

<script type="text/javascript">

var btn3=document.getElementById("btn3");

if(btn3.addEventListener){

btn3.addEventListener("click",demo);

}else if(btn3.attachEvent){

btn3.attachEvent("onclick",demo);

}else{

btn3.onclick=demo();

}

</script>

HTML处理方式需要修改的地方比较多，DOM0级会被覆盖掉，DOM2级不会被覆盖掉，使用起来比较方便。

三、事件对象

1．事件对象：

在触发DOM事件的时候都会产生一个对象

2．事件对象event：

1）：type：获取事件类型

2）：target：获取事件目标

3）：stopPropagation（）：阻止事件冒泡

4）：preventDefault（）：阻止事件默认行为