事件——怎样使用事件以及IE和DOM事件模型之间存在哪些主要差别，有需要的朋友可以参考一下

（1）冒泡型事件：事件按照从最特定的事件目标到最不特定的事件目标(document对象)的顺序触发。

  IE 5.5: div -> body -> document

  IE 6.0: div -> body -> html -> document

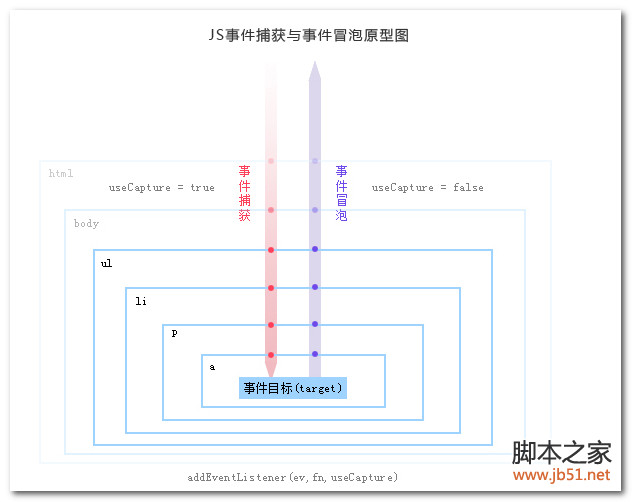
  Mozilla 1.0: div -> body -> html -> document -> window

（2）捕获型事件(event capturing)：事件从最不精确的对象(document 对象)开始触发，然后到最精确(也可以在窗口级别捕获事件，不过必须由开发人员特别指定)。

（3）DOM事件流：同时支持两种事件模型：捕获型事件和冒泡型事件，但是，捕获型事件先发生。两种事件流会触及DOM中的所有对象，从document对象开始，也在document对象结束。

  DOM事件模型最独特的性质是，文本节点也触发事件(在IE中不会)。

支持W3C标准的浏览器在添加事件时用addEventListener(event,fn,useCapture)方法，基中第3个参数useCapture是一个Boolean值，用来设置事件是在事件捕获时执行，还是事件冒泡时执行。而不兼容W3C的浏览器(IE)用attachEvent()方法，此方法没有相关设置，不过IE的事件模型默认是在事件冒泡时执行的，也就是在useCapture等于false的时候执行，所以把在处理事件时把useCapture设置为false是比较安全，也实现兼容浏览器的效果。



事件捕获阶段：事件从最上一级标签开始往下查找，直到捕获到事件目标(target)。  
事件冒泡阶段：事件从事件目标(target)开始，往上冒泡直到页面的最上一级标签。  
  
假设一个元素div，它有一个下级元素p。  
<div>  
　　<p>元素</p>  
</div>  
这两个元素都绑定了click事件，如果用户点击了p，它在div和p上都触发了click事件，那这两个事件处理程序哪个先执行呢？事件顺序是什么？  
**两种模型**以前，Netscape和Microsoft是不同的实现方式。

Netscape中，div先触发，这就叫做事件捕获。

Microsoft中，p先触发，这就叫做事件冒泡。

两种事件处理顺序刚好相反。IE只支持事件冒泡，Mozilla, Opera 7 和 Konqueror两种都支持，旧版本的Opera's 和 iCab两种都不支持 。

**事件捕获**当你使用事件捕获时，父级元素先触发，子级元素后触发，即div先触发，p后触发。

**事件冒泡**当你使用事件冒泡时，子级元素先触发，父级元素后触发，即p先触发，div后触发。

**W3C模型**W3C模型是将两者进行中和，在W3C模型中，任何事件发生时，先从顶层开始进行事件捕获，直到事件触发到达了事件源元素。然后，再从事件源往上进行事件冒泡，直到到达document。

程序员可以自己选择绑定事件时采用事件捕获还是事件冒泡，方法就是绑定事件时通过addEventListener函数，它有三个参数，第三个参数若是true，则表示采用事件捕获，若是false，则表示采用事件冒泡。

ele.addEventListener('click',doSomething2,true)

true=捕获

false=冒泡

**传统绑定事件方式**在一个支持W3C DOM的浏览器中，像这样一般的绑定事件方式，是采用的事件冒泡方式。

ele.onclick = doSomething2

**IE浏览器**如上面所说，IE只支持事件冒泡，不支持事件捕获，它也不支持addEventListener函数，不会用第三个参数来表示是冒泡还是捕获，它提供了另一个函数attachEvent。

ele.attachEvent("onclick", doSomething2);

附：事件冒泡（的过程）：事件从发生的目标（event.srcElement||event.target）开始，沿着文档逐层向上冒泡，到document为止。

**事件的传播是可以阻止的：**• 在W3c中，使用stopPropagation（）方法  
• 在IE下设置cancelBubble = true；  
在捕获的过程中stopPropagation（）；后，后面的冒泡过程也不会发生了~  
3.阻止事件的默认行为，例如click <a>后的跳转~  
• 在W3c中，使用preventDefault（）方法；  
• 在IE下设置window.event.returnValue = false;  
4.哇，终于写完了，一边测试一边写的额，不是所有的事件都能冒泡，例如：blur、focus、load、unload，（这个是从别人的文章里摘过来的,我没测试）。