# Dom事件探秘

**(document)文档对象是HTML文档和所有其他节点的"所有者"的根节点：**
（元素节点，文本节点，属性节点和注释节点）。

**事件流**
浏览器发展到第四代的时候，
事件流---描述的是从页面中接受事件的顺序

**IE的事件冒泡流**
**Netscape的事件捕捉流**
事件冒泡：即最开始由最具体的元素（文档中嵌套层次最深的那个节点）接受，然后逐级向上传播至最不具体的那个节点（文档）
事件捕捉：不太具体的节点应该最早接受到事件，而最具体的节点最后接受到事件

**使用事件处理程序：**
**HTML事件处理程序**
直接把事件加载到html结构中的元素上的事件方法：叫做HTML事件

HTML事件处理程序，现在不建议使用了。
html代码与js代码链接太多紧密
缺点：HTML和js代码高耦合，如果修改，就要修改两個地方：HTML元素內和script函数。
**［dom0］级处理程序：**
1.是较传统的一种方式：把一个函数赋值给一个事件的处理程序属性用的比较多的方法
2.简单易用
3.有跨浏览器的优势.
方法：var btn1 = document.getElementByld();
btn1.onclick = function() {
//content
}

**DOM2级处理事件程序**
DOM2级事件定义俩个方法：
用于处理指定和删除事件处理程序的操作
addEventListener（）和removeEventlistner（）
接收三个参数：要处理的事件名,作为事件处理的函数和布尔值 （true表示在捕捉阶段处理程序，false是表示在冒泡阶段处理程序）

1、若事件名称有on，则需去掉on。onclick -----> click，onmouseover -----> mouseover等等；
2、false 兼容所有浏览器-----事件冒泡流。
3.通过addEventListener添加的事件只能通过removeEventListener来删除。---参数要一样
btn.removeEventListener(参数);
//参数必须和btn.addEventListener的参数一致，删除事件

dom0级和dom2级事件处理程序 可以给一个事件绑定多个函数，事件触发的時候会按照綁定順序执行各个函数。
btn3.addEventListener('click',showMes,false);
btn3.addEventListener('click',showMes,function(){alert(this.value）;},false);

PS:「this」关键字是指"引用被触发的元素

缺点：IE不支持该事件

**IE事件处理程序**

IE也提供了类似DOM0级和DOM2级事件处理程序

attachEvent()--------添加事件
detachEvent()--------删除事件
接收相同的两个参数，事件处理程序的名称 和 事件处理程序的函数
去掉了布尔值，不使用第三個参数的原因:
--IE8以及更早的浏览器版本只支持事件冒泡！
在ie下给btn3添加事件

btn3.attachEvent('onclick',showMes);
注意：在ie事件处理程序上，又要把on加上
btn3.detachEvent('onclick',showMes);
支持ie事件处理程序的浏览器主要有ie和opera两个浏览器。

[跨浏览器解决事件处理程序]

要达到跨浏览器的事件处理程序, 我们需要判断使用者的浏览器支持哪一种事件处理程序, 所以我们将判断以及函式包在一个物件裡面, 这里用函数elementUtil求包.

```
var elementUtil = {
// 添加事件
addHandler: function(element,type,handler) {
if(element.addEventListener) {
element.addEventListener(type,handler,false);
} else if(element.attachEvent) {
element.attachEvent('on'+type,handler);
}else {
element['on'+type]=handler;
// element.'on'+type = handler; 这种写法是錯的!
// 要写成用中括号的形式
// 因为：element.onclick === element[onclick];
}
}

// 删除事件
removeHandler: function(element,type,handler) {
if(element.removeEventListener) {
element.removeEventListener(type,handler,false);
} else if(element.detachEvent) {
element.detachEvent('on'+type,handler);
}else {
element['on'+type]=null;
}
}
};
```

**事件对象**
什么是事件对象？在触发DOM上的事件时都会产生一个对象
事件对象event
①DOM中的事件对象 属性
（1）、type属性用于获取事件类型
（2）、target属性用于获取事件目标
（3）、stopPropagation()方法 用于阻止事件冒泡
（4）、preventDefault() 方法 阻止事件的默认行为

如果一个div里面有一个button，button和div都绑定了click事件，如果是事件冒泡，那么点击button的時候，首先触发button的处理函数，后後触发div的处理函数。（由內而外）
如果有时候不想冒泡，也就是不希望div的事件被触发，那么需要阻止事件冒泡。stopPropagation()方法
如果需要阻止a标签的默认属性跳转，可以使用阻止事件的默认行为preventDefault()方法

**对于ie浏览器的一些特殊处理:**
在ie8以前,event属性的获取是通过 window.event.而且,事件的目标对象不一样, 谷歌 火狐的是target代表目标事件, 而ie浏览器是srcElement代表目标事件.
综上:我们的代码应该这样写

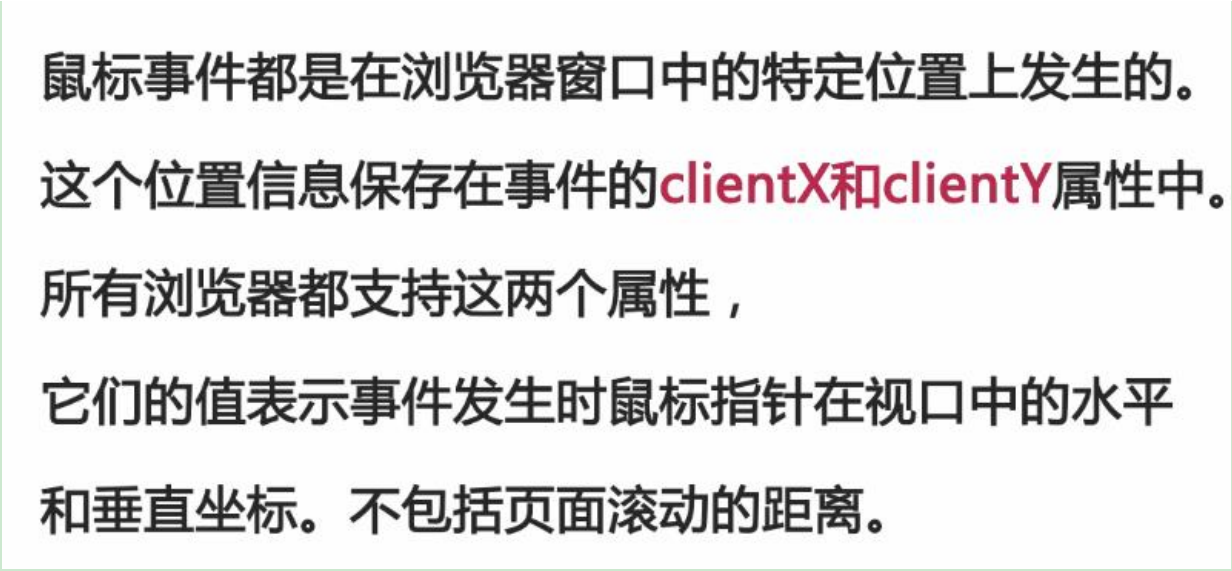
```
event = event || window.event;
var eve = event.target || event.srcElement;
```

（IE中事件对象为window.event)

```
event=event || window.event
```

1. type属性 用于获取属性目标
2. srcElement属性 用与获取属性目标 event.target || event.srcElement
3. cancelBubble属性 用于阻止事件冒泡。( cancelBubble = true阻止 false不阻止 )
4. returnValue属性 阻止事件的默认行为。( returnValue = false阻止 )

**QQ面板拖拽效果**




1、1、getElementsByClassName这个函数兼容性不好，(js ie10以下版本不支持document.getElementsByClassName)
最好自己造一个：js封装getClass方法，获取class。理解getClass方法的构造原理。

2、数组的push方法：可向数组的末尾添加一个或多个元素，并返回新的长度

3、任何能够跟着鼠标移动的东西 都要有一个前提：绝对定位！

4、鼠标事件都是在浏览器窗口中的特定位置上发生的。这个位置信息保存在事件event的clientX和clientY属性中。
所有浏览器都支持这两个属性，它们的值表示事件发生时鼠标指针在视口中的水平和垂直坐标。不包括页面滚动的距离。

5、实现随鼠标拖动，需要让窗口的坐标随着鼠标移动，鼠标的坐标保存在事件event的clientX和clientY两个属性中。
PS：document.onclick是代表在页面的任何地方点击事件。

6、onmousedown鼠标按下，窗口和鼠标的位置同步——onmousemove：当鼠标指针在元素内部移动时重复地触发
element.style.left/top=clientX/Y+'px';（这样有bug，需要求出光标落点距离面板的位置）

document.documentElement.clientWidth 是指可视窗口的宽度
document.documentElement.clientHeight 是指可视窗口的高度
原理：拖动的时候，光标位置在哪，面板位置就在哪，面板的坐标通过左上角的点来确定。
offset 是指对象跟父容器之间的距离
分析并解决Bug
1、确定光标在屏幕中的位置：clientheight/width
2、确定面板的位置：offsetwidth/height
3、面板的移动位置限制：四个方向
4、删除事件，mouseup方法


**1.首先分析实现原理；然后分析要取出的对象，进行取出；再给对象绑定事件；**
**2.分析各种事件，并对其件进行函数封装；**
**3.块的里面的文字（状态、下）不见了：用负缩进把他们搞到窗口之外了，当代码注释使用。**
**4.在适当的地方阻止事件冒泡：**
**ul父元为div，点击li时希望ul隐藏，点击div时希望其显示，在点击li后会冒泡到div ,因而需要阻止冒泡；**
**注意区分onmousedown和onclick，只能阻止相对应类型事件。**
**重点：利用事件冒泡实现切换状态菜单；当一个块内包含众多事件时，必须要注意到事件冒泡的影响。**
**5.在其他任何地方点击，要使列表隐藏：document.onclick是代表在页面的任何地方点击事件。**
**document下的子元素还有一个onclick事件，所以要注意事件冒泡的影响；**

[键盘事件]

**keyDown:当用户按下键盘上的」任意鍵」時触发，而且如果按住不放的话，会重复触发此事件**
**keyPress:当用戶按下键盘上的」字符键」時触发，而且如果按住不放的话，会重复触发此事件**
**keyUp:当用戶释放键盘上的键時触发**

console.log
主要是方便你调式javascript用的,你可以看到你在页面中输出的内容。
相比alert他的优点是：
他能看到结构化的东西，如果是alert，淡出一个对象就是[object object],但是console能看到对象的内容。
console不会打断你页面的操作，如果用alert弹出来内容，那么页面就死了，但是console输出内容后你页面还可以正常操作。
console里面的内容非常丰富，你可以在控制台输入：console，然后就可看到它有网页的各种提示。

键盘事件
onkeydown：按下键盘上任意键时触发，（按住不放会重复触发）
onkeypress：按下键盘上的字符键时触发
onkeyup：释放键盘上的键时触发,（即按住不会重复触发）
keyCode：event.keyCode，获得当前按下键盘上按键的键码，回车键为13

定时器：
注意：使用timer前一定要进行初始化=====>var timer[];
timer=setInterval(function(){},50)：每隔50ms执行一次函数
clearInterval(定时器名)：清除定时器，再加定时器前需清除原来的定时器，防止多个定时器叠加

随机数：
Math.random()：生成0-1的随机数
Math.floor()：向下取整