## 1：事件流

当页面元素触发事件的时候，该元素的容器以及整个页面都会按照特定顺序响应该元素的触发事件，事件传播的顺序叫做事件流程。

**一：事件流的分类**

1.冒泡型事件(所有的浏览器都支持)

由明确的事件源到最不确定的事件源依次向上触发。

2.捕获型事件(IE不支持 支持w3c标准的浏览器 火狐)

不确定的事件源到明确的事件源依次向下触发。

addEventListener(事件，处理函数，false); 事件在冒泡阶段执行。

addEventListener(事件，处理函数，true) ; 事件在捕获阶段执行。

**二：阻止事件流**

IE：事件对象.cancelBubble=true;

火狐：事件对象.stopPropagation();

捕获型事件流和冒泡型事件流对可以阻止。

**三：执行顺序**

当元素即绑定捕获型事件又绑定冒泡型事件执行顺序：先捕获后冒泡。

如果该元素是目标事件源对象，则谁绑定在前就先执行谁。

## 2：阻止右键菜单

document.oncontextmenu = function(){

alert("右键被按下");

return false; //阻止默认行为；

}

## 3：阻止浏览器的默认行为

非IE：ev.preventDefault();

IE：ev.returnValue = false;

## 应用：

1：掌握兼容问题。

2：自定义右键菜单。

3：掌握事件监听的封装。

4：掌握冒泡和捕获的区别。

## 4：事件委托

利用冒泡机制，将子元素事件委托给父元素执行。

作业：利用事件委托实现删除元素。

## 5：拖拽效果

**一：拖拽原理**

三个事件：onmousedown、onmousemove、onmouseup

实现思路：

1：给目标元素添加**onmousedown**事件，拖拽的前提是在目标元素按下鼠标左键。

2：当**onmousedown**发生以后，此刻给**document**添加**onmousemove**事件，意味着此刻鼠标在网页的移动都将改变目标元素的位置。

3：在**onmousemove**事件中，设定目标元素的left和top，公式：

目标元素的left = 鼠标的clientX – （鼠标和元素的横坐标差，即offsetX）  
目标元素的top = 鼠标的clientY– （鼠标和元素的纵坐标差，即offsetY）。

4：当**onmousedown**发生以后，此刻给**document**添加**onmouseup**事件，意味着此刻鼠标在网页的任意位置松开鼠标，都会放弃拖拽的效果。

5：在**onmouseup**事件中，取消**document**的**onmousemove**事件即可。

**二：注意事项**

1：拖拽过程中，会选中文字，应阻止浏览器的默认行为（return false）。

2：低版本IE下，应使用事件捕获obj.setCapture()，在mouseup时取消事件捕获obj.releaseCapture()。

**三：扩展**

1：限定拖拽范围（讲解）。

## 6：完美拖拽

window.onload = function () {

oDiv.onmousedown = function (ev) {

var

ev = ev || window.event,

iOffsetL = ev.clientX – oDiv.offsetLeft,

iOffsetT = ev.clientY – oDiv.offsetTop;

if(oDiv.setCapture){

oDiv.setCapture();  
}

document.onmousemove = function (ev) {

var

ev = ev || window.event,

iPosX = ev.clientX – iOffsetL,

iPosY = ev.clientY – iOffsetT;

oDiv.style.left = iPosX + "px";

oDiv.style.top = iPosY + "px";

}

document.onmouseup = function () {

document.onmousemove = null;

document.onmouseup = null;

if(oDiv.releaseCapture) {

oDiv.releaseCapture();

}

}

return false;

}

}

## 应用：

1：弹出窗口拖拽。

**扩展案例：**拖拽轨迹回放。

## 综合应用：

1：九宫格拖拽交换。

