## 1：事件驱动

1.事件

所谓事件就是js侦测到的用户的操作或是页面的一些行为(怎么发生的)

2.事件源

引发事件的元素。(发生在谁的身上)

3.事件处理程序

对事件处理的程序或是函数 (发生了什么事)

4.事件对象

当事件发生的时候，具体发生了哪些信息，（当按键盘的时候按的哪个键或者鼠标光标在页面的哪个位置等等）。

## 2：事件分类

1.鼠标事件

onclick 点击鼠标 ondblclick 双击鼠标

onmousedown 鼠标按下 onmouseup 鼠标抬起

onmousemove 鼠标移动 onmouseover 鼠标放上去（子元素继承父元素的onmouseover属性）

onmouseout 鼠标离开 onmouseenter 鼠标进入（子元素不继承父元素的onmouseenter属性）

onmouseleave 鼠标离开

2.键盘事件

onkeyup 键盘抬起 onkeydown 键盘按下

onkeypress 键盘按下或按住（对功能键没用）

3. 表单事件

onsubmit 提交

onblur 失去焦点

onfocus 获得焦点

onchange 改变文本区域的内容

4.页面事件

onload 当页面加载完成

onbeforeunload（疑问？？） 当页面卸载完成之前

## 3：绑定事件

**1：绑定一个事件处理程序：**

1.在脚本中绑定。

2.直接在HTML元素中绑定。

**2：绑定多个事件处理程序：**

1.自己写的。

2.IE：（多个函数执行顺序：IE9/10按照绑定的顺序执行，IE7/8顺序相反。）

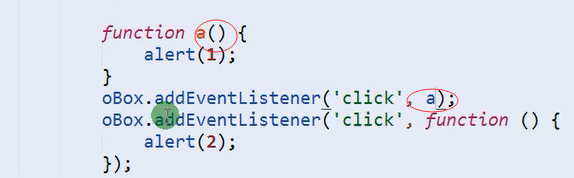
对象.attachEvent("事件(on)","处理程序") 添加

对象.detachEvent("事件(on)","处理程序") 删除

火狐：（多个函数执行顺序：按照绑定的顺序执行。）

对象.addEventListener("事件"(不加on),"处理程序",布尔值) 添加

对象.removeEventListener("事件"(不加on),"处理程序",布尔值) 删除



## 4：事件对象

用来记录一些事件发生时的相关信息的对象。

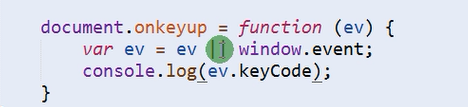
1.只有当事件发生的时候才产生，只能在处理函数内部访问。

2.处理函数运行结束后自动销毁。

**如何获取事件对象**：

IE：window.event

火狐：对象.on事件=function(ev){}



## 5：关于鼠标事件对象属性

**相对于浏览器位置的**：

clientX 当鼠标事件发生的时候，鼠标相对于浏览器X轴的位置。

clientY 当鼠标事件发生的时候，鼠标相对于浏览器Y轴的位置。

**相对于文档位置的**：

pageX 当鼠标事件发生的时候，鼠标相对于文档X轴的位置。（IE7/8无）

pageY 当鼠标事件发生的时候，鼠标相对于文档Y轴的位置。（IE7/8无）

**相对于屏幕位置的**：

screenX 当鼠标事件发生的时候，鼠标相对于屏幕X轴的位置。

screenY 当鼠标事件发生的时候，鼠标相对于屏幕Y轴的位置。

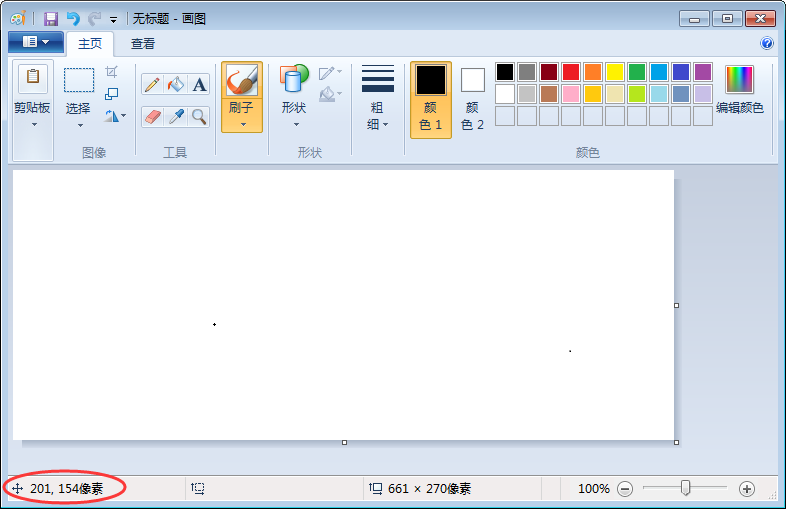
**相对于事件源的位置**：

offsetX 当鼠标事件发生的时候，鼠标相对于事件源X轴的位置。

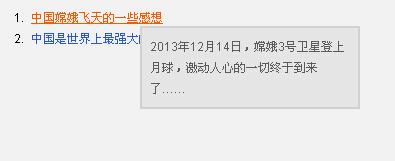
offsetY 当鼠标事件发生的时候，鼠标相对于事件源Y轴的位置。

## 应用：

1：光标的实时显示。



2：跟随鼠标的提示框。



3：输入框默认提示。



## 6：关于键盘事件对象的属性

**keyCode**：获得键

盘码。空格：32 回车13 左上右下：37 38 39 40。which属性有兼容性问题。

ctrlKey：判断ctrl键是否被按下，按下是true,反之是false 布尔值。**还有shiftKey altKey。**

type：用来检测事件的类型。

button：0代表鼠标按下了左键 || 1代表按下了滚轮 || 2代表按下了右键。不过老版本的**IE**并没有遵守W3C的规范，它的button属性含义如下：1鼠标左键 2鼠标右键 3左右同时按 4滚轮 5左键加滚轮 6右键加滚轮 7三个同时。**目前IE11.0版本，无兼容性问题。**

## 7：目标事件源对象

例如：被点击的对象box；

IE：事件对象.srcElement

火狐：事件对象.target

## 应用：

1：CSS模拟下拉菜单。



## 综合应用：

1：聊天对话框。

