# JavaScript进阶

第2天课堂笔记（本课程共6天）

# 目录

[JavaScript进阶 1](#_Toc476950464)

[目录 1](#_Toc476950465)

[一、事件对象 2](#_Toc476950466)

[1.1 事件对象是什么 2](#_Toc476950467)

[1.2 IE6、7、8的兼容问题 2](#_Toc476950468)

[二、各种值 4](#_Toc476950469)

[2.1 鼠标位置event.clientX、event.screenX 4](#_Toc476950470)

[2.2 得到盒子在页面内的净位置 6](#_Toc476950471)

[2.3 页面的卷动值 7](#_Toc476950472)

[2.4 在盒子内的位置 7](#_Toc476950473)

[三、窗口卷动 9](#_Toc476950474)

# 一、事件对象

## 1.1 事件对象是什么

[Event模型](https://developer.mozilla.org/zh-CN/docs/Web/API/Event)

我们可以把事件的处理函数，当做一个回调函数。click事件发生的时候，才调用这个function(){}

|  |
| --- |
| 1. oDiv.onclick = function(){ 3. } |

**浏览器在调用这个事件处理函数，调用的时候还传进去一个实际参数**，这个参数是一个对象，表示这次点击事件的所有细节。比如按的什么按钮、点击的位置等等。

所以，我们的事件处理函数，就要通过一个形式参数来接收。

|  |
| --- |
| 1. oDiv.onclick = function(**event**){ 3. } |

比如，获得鼠标当前的位置：

|  |
| --- |
| 1. document.onmousemove = function(event){ 2. h1.innerHTML = "当前鼠标坐标为" + **event.clientX** + "," + **event.clientY**; 3. } |

**event.clientX表示触发事件的时候，鼠标的x位置（相对于视口）**

不同的事件，就有不同的event值。可以想象到键盘事件，没有鼠标的位置。而是有event.keyCode按下去的键盘码的值。

## 1.2 IE6、7、8的兼容问题

超级兼容大坑：在调用事件处理函数的时候，高级浏览器是传给这个事件处理函数一个实际参数event对象。但是IE6、7、8是给window对象绑定一个当前的event属性。

IE中：

|  |
| --- |
| 1. document.onmousemove = function(){ 2. h1.innerHTML = "当前鼠标坐标为" + **window.event.clientX** + "," + **window.event.clientY**; 3. } |

所以现在就有一个兼容性的处理的小窍门：

|  |
| --- |
| 1. //在页面上鼠标移动的时候，触发事件 2. document.onmousemove = function(event){ 3. **event = event || window.event;** 4. h1.innerHTML = "当前鼠标坐标为" + event.clientX + "," + event.clientY; 5. } |

|| 就是或逻辑短路运算符，如果event有定义，那么就event对象；否则就是window.event对象。

[preventDefault](https://developer.mozilla.org/zh-CN/docs/Web/API/Event/preventDefault) [returnValue](https://developer.mozilla.org/zh-CN/docs/Web/API/Event/returnValue) [stopPropagation](https://developer.mozilla.org/zh-CN/docs/Web/API/Event/stopPropagation) [cancelBubble](https://developer.mozilla.org/zh-CN/docs/Web/API/UIEvent/cancelBubble)  [target](https://developer.mozilla.org/zh-CN/docs/Web/API/Event/target) [srcElement](https://developer.mozilla.org/zh-CN/docs/Web/API/Event/srcElement)

IE中有很多方法的名字和高级浏览器不一样，比如昨天说的**阻止默认事件**：

|  |
| --- |
| 1. event.preventDefault(); |

在IE中是一个属性：

|  |
| --- |
| 1. event.returnValue = false; |

这些的兼容性问题，就用能力检测来解决：

|  |
| --- |
| 1. //阻止默认的滚动事件，阻止的方法有浏览器兼容问题： 2. if(event.preventDefault){ 3. event.preventDefault(); 4. }else{ 5. event.returnValue = false; 6. } |

包括preventDefault()方法在IE6、7、8中不支持在内，我们要记住这三个：

event.preventDefault()不兼容！ 在IE6、7、8中用event.returnValue=false; 代替

event.stopPropagation()不兼容！在IE6、7、8中用event.cancelBubble=true; 代替

event.target不兼容，在IE6、7、8中用event.srcElement代替

event.target兼容解决，就是用一个短路语法：

|  |
| --- |
| 1. var targetElem = event.target || event.srcElement; |

# 二、各种值

## 2.1 鼠标位置event.clientX、event.screenX

[clientX](https://developer.mozilla.org/zh-CN/docs/Web/API/MouseEvent/clientX) [clientY](https://developer.mozilla.org/zh-CN/docs/Web/API/MouseEvent/clientY) [screenX](https://developer.mozilla.org/en-US/docs/Web/API/MouseEvent/screenX) [screenY](https://developer.mozilla.org/en-US/docs/Web/API/MouseEvent/screenY)

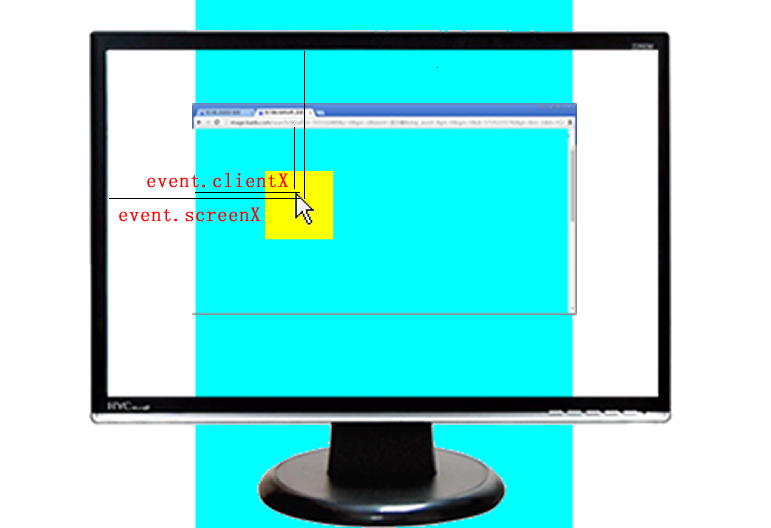
下面4个值是全线兼容的：

|  |
| --- |
| 1. event.clientX 2. event.clientY 3. event.screenX 4. event.screenY |

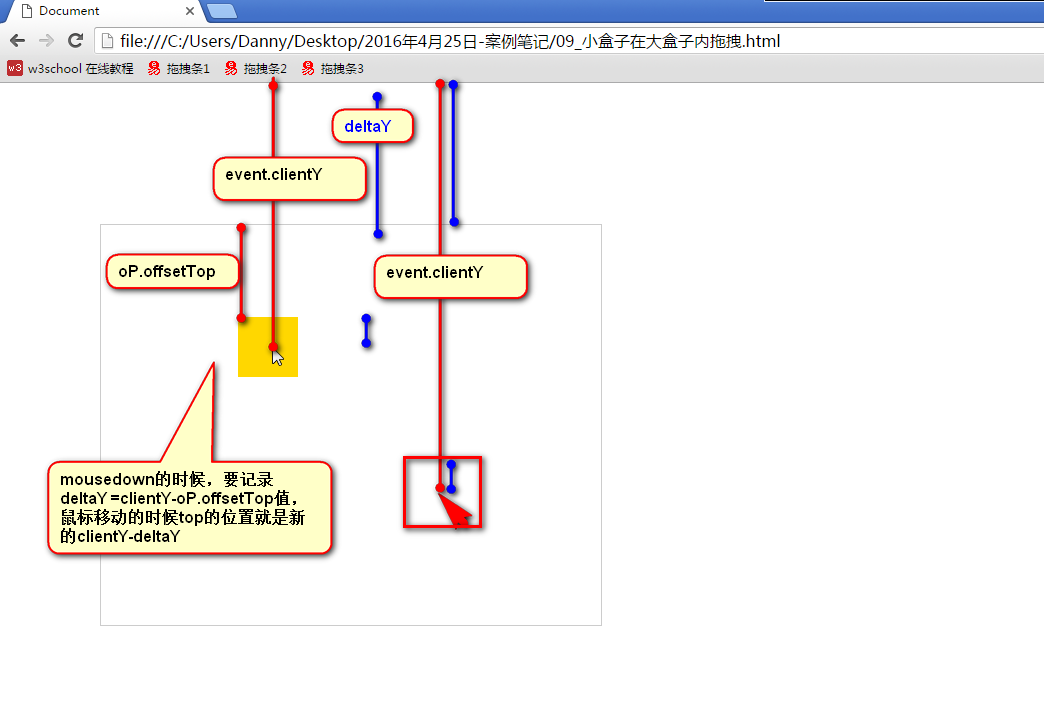
这四个值都没有单位。

clientX表示鼠标距离视口左边x值

screenX表示鼠标距离整个屏幕左边x值



拖拽的算法：



## 2.2 得到盒子在页面内的净位置

我们之前学习过oDiv.offsetTop指的是这个盒子的border外层到自己的offsetParent的border内侧（IE8多算了一条border），至于offsetParent是谁，IE6、7有不同的理解。

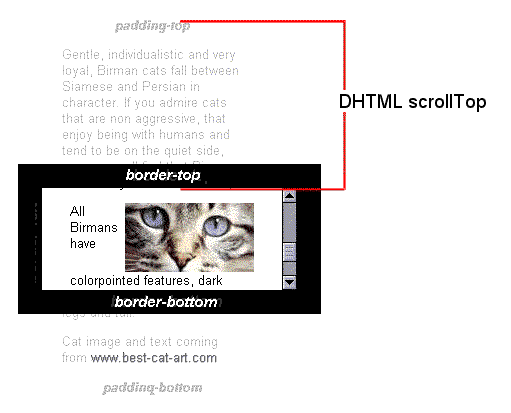
我们现在说的，是oDiv这个元素在页面中的净位置，自己起一个名字allTop。

得到allTop不难，迭代，一层一层的累加offsetTop值。

|  |
| --- |
| 1. //返回这个元素在页面中的净位置 2. //就是这个元素所有offsetParent的offsetTop值的和 3. function getAllTop(obj){ 4. //累加器，累加器的初始值不是0，而是自己现在offsetTop值 5. //一会儿while语句直接从它爸开始了 6. var allTop = obj.offsetTop; 7. //当前正在算高度的元素 8. var currentObj = obj; 9. while(currentObj = currentObj.offsetParent){ 10. allTop += currentObj.offsetTop; 11. } 12. return allTop; 13. } |

## 2.3 页面的卷动值

页面被滚动条卷动走的量 ScrollTop



窗口的卷动事件onscroll，无论因为什么（比如鼠标拖拽滚动条、鼠标滚轮、键盘pagedown、键盘下箭头）导致了页面的卷动，触发这个事件。

|  |
| --- |
| 1. //当窗口卷动的时候发生这个事件 2. window.onscroll = function(){ 3. } |

高级浏览器认为scrollTop是body元素的属性。而IE6、7、8坚决认为是<html>标签的属性。

所以，我们要这么写：

|  |
| --- |
| 1. document.body.scrollTop || document.documentElement.scrollTop; |

document.documentElement就是<html>标签

事实上，很多东西都有这个区分，比如浏览器窗口的宽度。

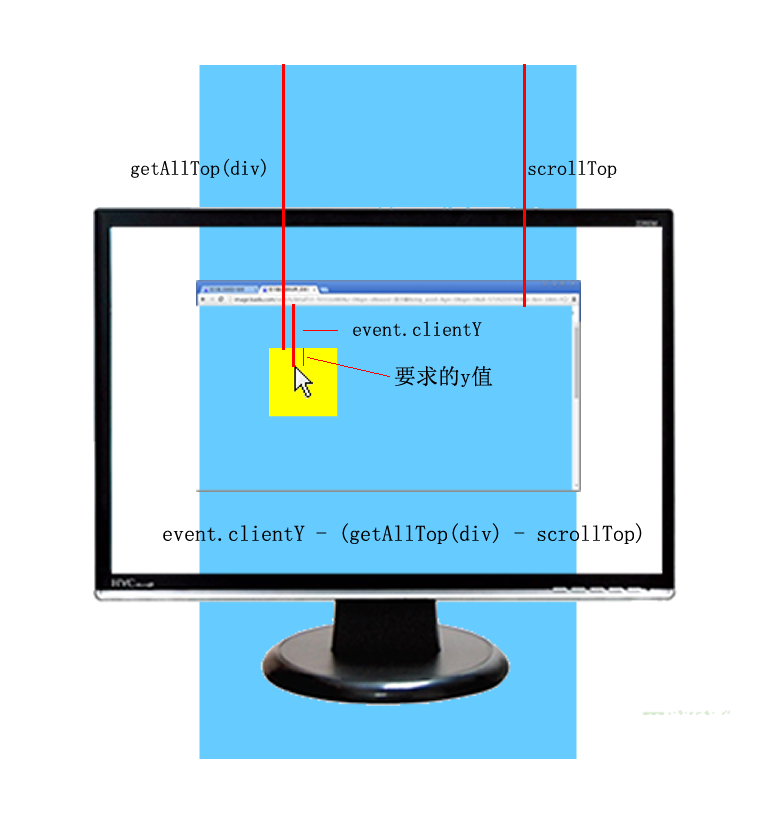
## 2.4 在盒子内的位置

event.offsetX

event.offsetY

表示鼠标在盒子的位置，火狐不兼容，但是我们发现现在的火狐嗷嗷兼容。

var y = event.clientX - (getAllLeft(smallPic) - scrollLeft)要想全线兼容，必须进行周转：



应该用event.clientY和getAllLeft(smallPic)和页面的scrollTop值进行减法比较。

|  |
| --- |
| var x = event.clientX - (getAllLeft(smallPic) - scrollLeft)  var y = event.clientY - (getAllLeft(smallPic) - scrollTop) |

# 三、窗口卷动

窗口卷动的时候触发的事件：

|  |
| --- |
| 1. window.onscroll = function(){ 2. } |

卷动值：

|  |
| --- |
| 1. document.body.scrollTop || document.documentElement.scrollTop |

卷动值不是可读的，也可以设置。甚至是动画。

复习：

元素的尺寸、位置：

oDiv.offsetLeft

oDiv.offsetTop

oDiv.clientWidth //width+padding

oDiv.clientHeight

oDiv.offsetWidth //width+padding+border

oDiv.offsetHeight

event对象的鼠标当前位置

event.clientX //鼠标到视口

event.clientY

event.screenX //鼠标到屏幕

event.screenY

event.offsetX //说是不兼容火狐，但现在火狐兼容，到最小盒子的左上角

event.offsetY

窗口的

document.body.scrollTop || document.documentElement.scrollTop //卷动值

document.body.clientWidth || document.documentElement.clientWidth //视口宽度