

# 一. 鼠标事件

## 1. 鼠标事件

**鼠标事件有下面这几种：** onscroll, onclick, onmousedown, onmousemove, onmouseup, onmouseover

### 1.1. onclick

鼠标点击事件

```
box.onclick = function(e){  
    console.log(e)  
}
```

### 1.2. onmousedown

鼠标按下事件

```
box.onmousedown = function(e){  
    console.log(e)  
}
```

### 1.3. onmouseup

鼠标松开事件

```
box.onmouseup = function(e){  
    console.log(e)  
}
```

### 1.4. onmousemove

鼠标移动事件

```
box.onmousemove = function(e){  
    console.log(e)  
}
```

## 1.5. onmouseover

鼠标经过事件

```
box.onmouseover = function(e){  
    console.log(e)  
}
```

## 1.6. onmouseout

鼠标划出事件

```
box.onmouseout = function(e){  
    console.log(e)  
}
```

同时需要熟悉 javascript 各种坐标

## 1.7. onscroll

解释：当元素的滚动条滚动时触发的事件。

onscroll 事件任何实体元素都可以绑定，这里的实体元素包括 DOM 元素、window 元素、document 元素。

用法即：element.onscroll=function({});

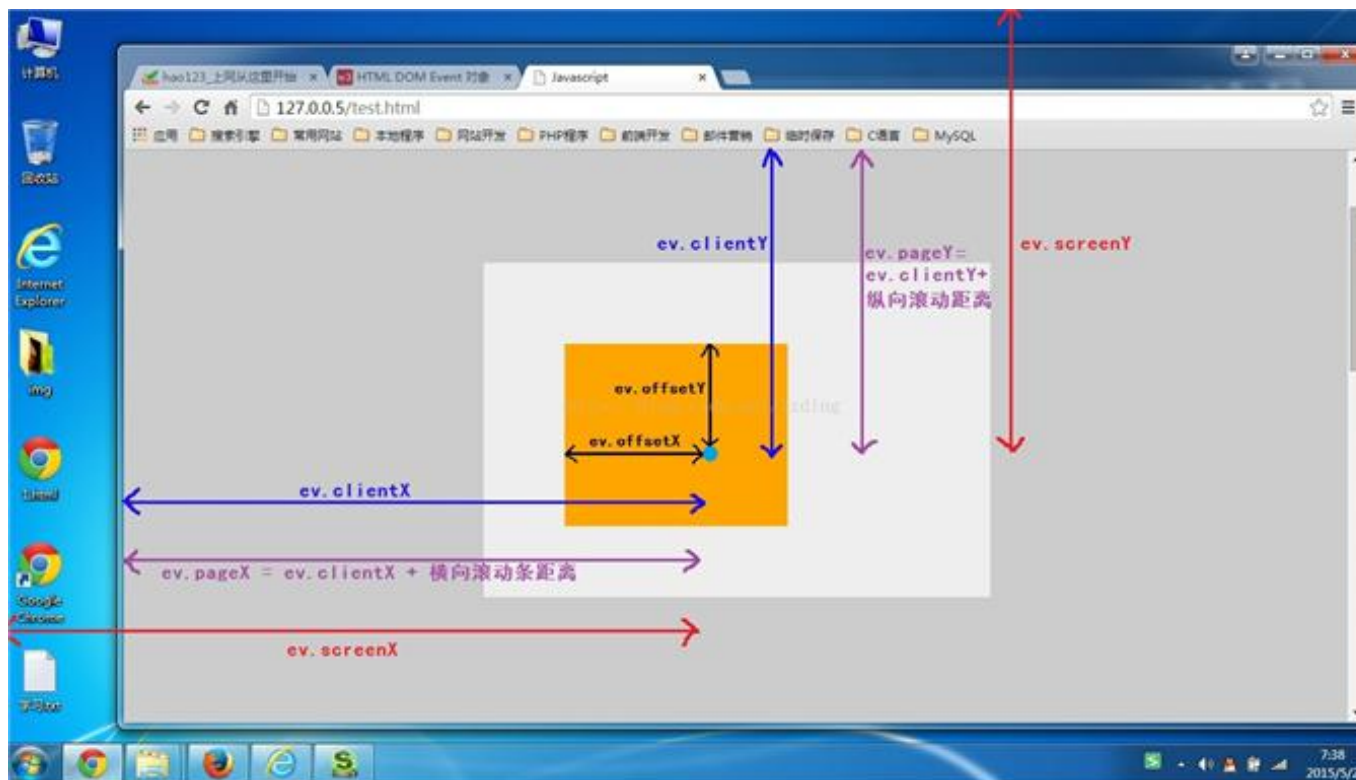
需要注意的是，滚动条一定要出现，而且滚动条是属于这元素的，例如：

```
<div id="wrap" style="height:100px;overflow:auto;">  
    <div id="inner" style="height:200px;">content</div>  
</div>
```

因为外层 wrap 的高度小于内层 inner 的高度，所以当设置 overflow:auto 时会出现滚动条，当拖动滚动条时就会触发 wrap 的 onscroll 事件，而不是 inner 的 onscroll 事件，即这滚动条属于 wrap 而不是属于 inner，明白这点十分重要，对下面理解的 scrollTop、scrollHeight 一样道理。

## 二. clientX、offsetX、screenX、pageX、x 的区别

区别：



## 浏览器的兼任情况

	chrome	firefox	ie8 -	ie9	ie10 +
offsetX	✓	✗	✓	✓	✓
clientX	✓	✓	✓	✓	✓
pageX	✓	✓	✗	✓	✓
screenX	✓	✓	✓	✓	✓
layerX	✓	✓	✗	✓	✓
x	✓	✗	✓	✓	✓

## scrollTop

解释：获取元素滚动后的距离文档顶部的距离，也就是滚动条滚动的距离（或者可以理解为 元素滚动条内的顶部隐藏部分的高度）。

scrollTop 属性只有 DOM 元素才有，window/document 没有。

用法 1：获取值 `var top = element.scrollTop;` //返回数字，单位像素

用法 2: 设置值 `element.scrollTop = 200;`

对上面的例子来说, 控制滚动条的位置是 `wrap.scrollTop=xx`; 而不是 `inner.scrollTop`, 道理同上。

兼容性问题: 获得整个文档 `scrollTop`, IE 是 `document.documentElement.scrollTop`, FF/CH 则是 `document.body.scrollTop`.

## scrollTop

**解释: 元素滚动条内的内容高度。**

`scrollTop` 同 `scrollTop` 属性一样, 只有 DOM 元素才有, `window/document` 没有。

不同的是 `scrollTop` 是只读, 不可设置。

兼容性问题: 获取整个文档 `scrollTop`, IE/FF/CH 都可以通过

`document.documentElement.scrollTop` 或 `document.body.scrollTop` 获得。

此外还有 `scrollLeft`, `scrollWidth`, 道理是一样的。

## 关于 `window.scroll()`, `window.scrollBy()`, `window.scrollTo()`

这 3 个是全局函数, 最新的 IE/FF/CH 都支持。

`window.scroll(x,y)` 是让 `window` 滚动条滚动到那个 `x,y` 坐标。// `x` 是水平坐标, `y` 是垂直坐标。

`window.scrollBy(-x,-y)` 是让 `window` 滚动条 **相对滚动到某个坐标**, `-10` 即相对向左/向上滚动 10 像素。

`window.scrollTo(x,y)` 和 `window.scroll(x,y)` 一样。

## `clientHeight`, `scrollTop`, `offsetHeight`

`clientHeight`: 可理解为内部可视区高度, 样式的 `height`+上下 `padding`

`scrollTop`: 内容的实际高度+上下 `padding` (如果没有限制 `div` 的 `height`, 即 `height` 是自适应的)

`offsetHeight`: 可理解为 `div` 的可视高度, 样式的 `height`+上下 `padding`+上下 `border-width`。

有个公式: `element.scrollTop - element.scrollTop === element.clientHeight`

用于判断滚动条是否滚动到底。

