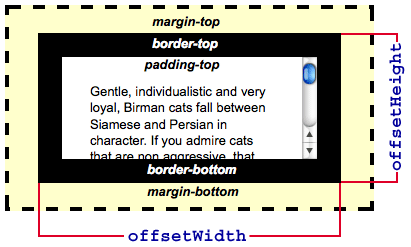
## 元素坐标

### offset (偏移：描述元素自身的大小和位置)

The HTMLElement.**offsetWidth** read-only property returns the layout width of an element. Typically, an element’s offsetWidth is a measurement which includes the element borders, the element horizontal padding, the element vertical scrollbar (if present, if rendered) and the element CSS width. 典型情况下，该值包括内容、边框、左右padding和垂直方向上的滚动条。

The HTMLElement.**offsetLeft** read-only property returns the number of pixels that the upper left corner of the current element is offset to the left within the [HTMLElement.offsetParent](https://developer.mozilla.org/en-US/docs/Web/API/HTMLElement/offsetParent" \o "The HTMLElement.offsetParent read-only property returns a reference to the object which is the closest (nearest in the containment hierarchy) positioned containing element. If the element is non-positioned, the nearest table, table cell or root element (html in standards compliant mode; body in quirks rendering mode) is returned by offsetParent. Returns null when the element has style.display set to \"none\". The offsetParent is useful because offsetTop and offsetLeft are relative to its padding edge.) node.

只读属性**offsetLeft**返回当前元素的**边框**外侧的左上角到定位了的父元素的左边框内侧的偏移量。

|  |  |
| --- | --- |
| offsetWidth  offsetHeight | 占位宽 = style.width（样式宽） + padding + border  占位高 = style.height + padding + border |
| offsetLeft  offsetTop | 当前元素到它的offsetParent之间的距离  边框到边框 = 子盒子定位用的left + margin |

**offsetLeft和style.left的区别**

1. offsetLeft**只读**，style.left可读可写
2. offsetLeft是**数值**，style.left是**字符串**并且有单位px
3. 如果没有加定位，style.left获取的数值可能是无效的
4. 最大区别在于offsetLeft以border左上角为基准，style.left以margin左上角为基准，相差一个小盒子的margin

### scroll (卷去：当内容超出容器时使用)

The **Element**.scrollWidth read-only property returns either the width in pixels of the content of an element or the width of the element itself, whichever is greater. If the element is wider than its content area (for example, if there are scroll bars for scrolling through the content), the scrollWidth is larger than the clientWidth.

只读属性**Element**.scrollWidth返回整个页面的实际宽度，包括隐藏的内容和展示出来的内容。Element.scrollLeft左边隐藏起来的宽度。

The Element.scrollLeft property gets or sets the number of pixels that an element’s content is scrolled to the left.

Note that if the element’s direction of the element is rtl (right - to - left) then scrollLeft is 0 when the scrollbar is at its rightmost position (at start of the scrolled content) and then increasingly negative as you scroll towards the end of the content.

|  |  |
| --- | --- |
| scrollWidth  scrollHeight | offsetoffsetWidth指的是**容器**的宽度，scrollWidth指的是容器里面的**内容**的宽度。 |
| scrollLeft |
| scrollTop |
| scrollX  scrollY | The read-only **scrollY** property of the [**Window**](https://developer.mozilla.org/en-US/docs/Web/API/Window) interface returns the number of pixels that the **document** is currently scrolled vertically.  window.scrollY返回文档在垂直方向已滚动的像素值。  pageYOffset 是 scrollY 的别名，为了跨浏览器兼容，请用 window.pageYOffset 代替 window.scrollY |

要获取当前页面的滚动条纵坐标位置，用document.documentElement.scrollTop，而不是：document.body.scrollTop，前者对应的是html标签，而body对应的是body标签。

* 如果滚动条出现在浏览器窗口最右侧，那么

window.onscroll = function() {

console.log(window.pageYOffset); //等价于window.scrollY

}

* 如果在某个盒子上加上overflow: scroll; 的话

element.onscroll = function() {

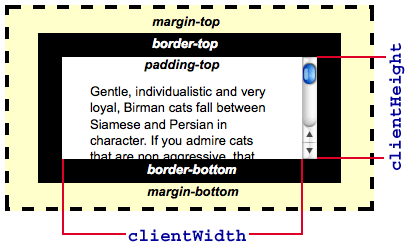
console.log(container.scrollTop);

}

这时scrollTop的值不断变化，但是window.pageYOffset一直为0，因为html并没有滚动。

pageYOffset 和scrollTop 的区别在使用jQuery开发跨平台移动应用时可以清晰地表现出来，因为多个页面可以定义在单独一个html页面内。在大多数情况下，这两个属性的表现是一样的。但是window.pageYOffset 针对的仅仅是整个html页面，而scrollTop属性所有的元素都可以用（但是兼容性很差）。

### client (可视区)

The Element.clientWidth read-only property is zero for elements with no CSS or **inline** layout boxes, otherwise it’s the inner width of an element in pixels. It includes padding but not the vertical scrollbar (if present, if rendered), border or margin.

只读属性clientWidth | clientHeight包括content + padding，不包括滚动条、边框、margin

|  |  |
| --- | --- |
| clientWidth  clientHeight | 可视区宽 = 样式宽 + padding  可视区高 = 样式高 + padding 不包括滚动条和边框 |
| clientLeft  clientTop | clientLeft和tclientTop没什么用，他们就是borderTop和borderLeft（如果有滚动条会包含滚动条的宽度） |

在IE678中，用document.body.clientWidth代替window.innerWidth来获取视口的宽度（Width (in pixels) of the browser window viewport including, if rendered, the vertical scrollbar.）

## 事件坐标

Event对象代表事件的状态，比如事件在其中发生的元素、键盘按键的状态、鼠标的位置、鼠标按钮的状态。事件通常与函数结合使用，函数不会在事件发生前被执行。

|  |  |  |  |
| --- | --- | --- | --- |
| Event Handlers | | | |
| onblur | onmouseup | onclick | onresize |
| onfocus | onmousedown | ondblclick | onselect |
| onkeyup | onmouseover | onchange | onsubmit |
| onkeydown | onmouseout | onload | onerror |
| onkeypress | onmousemove | onunload |  |

|  |  |
| --- | --- |
| 标准Event属性 | 标准Event方法 |
| target触发事件的元素 | initEvent() 初始化 |
| currentTarget返回其事件监听器触发该事件的元素 | preventDefault() 通知浏览器不要执行与事件关联的默认动作 |
| type返回Event对象表示的事件的名称 | stopPropagation() 阻止冒泡 |

|  |  |  |  |
| --- | --- | --- | --- |
| Width / Height / Left / Top 针对的是DOM元素 | | | |
| X / Y针对的是**Event对象** | | | |
| offsetX/offsetY | an experimental technology |

### clientX / clientY (相对于当前窗口)

The MouseEvent.clientX read-only property provides the horizontal coordinate within the application’s client area at which the event occurred (as opposed to the coordinates within the page). For example, clicking in the top-left corner of the client area will always result in a mouse event with a clientX value of 0, regardless of whether the page is scrolled horizontally.

Event.clientX / Event.clientY 是鼠标指针对于浏览器（或客户区）的水平 / 垂直坐标。客户区指的是当前窗口。该坐标**不考虑**文档的滚动。如果事件发生在窗口的顶部，无论文档滚了多远，clientY 的值都是 **0**。

### screenX / screenY

screenX / screenY 是鼠标指针相对于屏幕的水平 / 垂直坐标。

### offsetX \*[experimental technology](https://developer.mozilla.org/en-US/docs/MDN/Contribute/Guidelines/Conventions_definitions#Experimental)

The MouseEvent.offsetX read-only property provides the offset in the X coordinate of the mouse pointer between that event and the padding edge of the target node.

offsetX / offsetY发生事件的地点在事件源元素的坐标系统中的x / y坐标。非标准属性，慎用！

### JS和JQ的区别

1. js中的scrollTop/ scrollLeft 与jQ中的scrollTop / scrollLeft的区别：

在js中是属性，在jq中是方法。

1. js中的offsetTop / offsetLeft 与jQ中的offset的区别：

|  |  |
| --- | --- |
| js | jQ |
| offset指当前元素到它的offsetParent之间的距离 | offset() 不管该元素如何定位，也不管其父元素如何定位，获取的都是该元素相对于当前视口的偏移。当执行拖拽操作时用这个方法最合适。  position() 获取匹配元素相对父元素的偏移。 |