目录

[宽高client scroll offset 3](#_Toc533535826)

[top系列 3](#_Toc533535827)

[offsetTop 3](#_Toc533535828)

[scrollTop 4](#_Toc533535829)

[clientTop 4](#_Toc533535830)

[top 4](#_Toc533535831)

[posTop 4](#_Toc533535832)

[height系列 4](#_Toc533535833)

[offsetHeight 4](#_Toc533535834)

[scrollHeight 4](#_Toc533535835)

[clientHeight 5](#_Toc533535836)

[对ducument.body 5](#_Toc533535837)

[对document.documentElement 5](#_Toc533535838)

[对html控件 5](#_Toc533535839)

[实例 5](#_Toc533535840)

[滚动事件 5](#_Toc533535841)

[特殊对象 6](#_Toc533535842)

[document.documentElement 6](#_Toc533535843)

[操作css 6](#_Toc533535844)

[querySelector(All) 6](#_Toc533535845)

[.className 7](#_Toc533535846)

[使用 7](#_Toc533535847)

[封装 7](#_Toc533535848)

[setAttribute 8](#_Toc533535849)

[obj.style 9](#_Toc533535850)

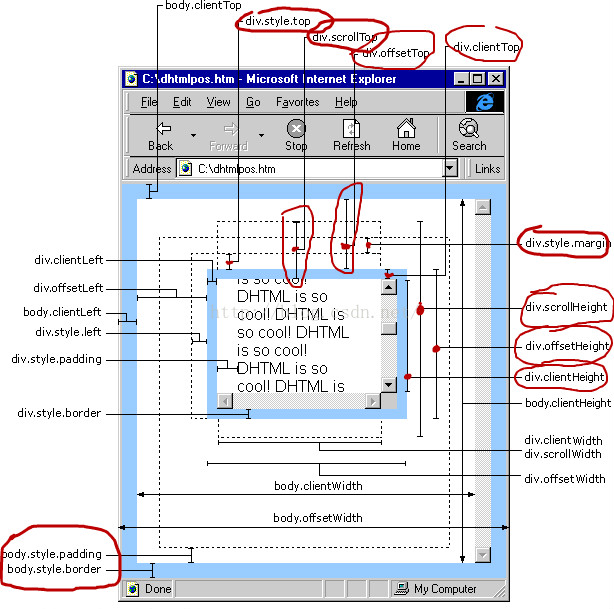
[obj.style.xxx/setProperty 9](#_Toc533535851)

[cssText 9](#_Toc533535852)

[js添加节点 9](#_Toc533535853)

[addRule/insertRule 9](#_Toc533535854)

# 宽高client scroll offset



三个高度都是只读的。

## top系列

### offsetTop

此属性可以获取元素的上外缘**距离最近采用定位父元素内壁的距离**，如果父元素中没有采用定位的，则是获取上外边缘距离文档内壁(body)的距离。所谓的定位就是position属性值为relative、absolute或者fixed。  
返回值是一个整数，单位是像素。  
此属性是**只读**的。

### scrollTop

此属性可以获取或者设置对象的最顶部到对象在当前窗口显示的范围内的顶边的距离，也就是元素滚动条被向下拉动的距离。  
返回值是一个**整数**，单位是像素。  
**此属性是可读写的。**

**特殊：**

document.body.scrollTop 始终为 0 （标准中）

document.documentElement.scrollTop： 实际滚动的高度。

### clientTop

f

### top

此属性仅仅在对象的定位(position)属性被设置时可用。否则，此属性设置会被忽略。

### posTop

posTop的数值其实和top是一样的，但区别在于，top固定了元素单位为px，而posTop只是一个数值,一般使用posTop来进行运算。

## height系列

### offsetHeight

此属性可以获取元素的宽度，宽度值包括:**元素内容+内边距+边框**。不包括外边距和滚动条部分。  
返回值是一个整数，单位是像素。

### scrollHeight

此属性获取对象的实际尺寸。

### clientHeight

此属性可以返回一个元素的高度值，值是:**元素的内容+内边距**。不包括边框、外边距和滚动条部分。  
返回值是一个整数，单位是像素。  
此属性是只读的。

### 对ducument.body

chrome里面三个相同，都是实际尺寸

### 对document.documentElement



两个是实际尺寸。

### 对html控件

offsetHeight是自身元素的高度，scrollHeight是 自身元素的高度+隐藏元素的高度。

## 经验

### 滚动高度

document：

document.documentElement.scrollTop || window.pageYOffset || document.body.scrollTop;

不同浏览器可能放在不同的值上面。

### 距离父元素顶部距离

static getElementTop(*element*) {

let actualTop = element.offsetTop;

let current = element.offsetParent;

while(current !== null) {

actualTop += current.offsetTop;

current = current.offsetParent;

}

return actualTop;

}

# 特殊对象

## document.documentElement

会返回文档对象（document）的根元素的只读属性（如HTML文档的 <html> 元素）。

# 操作css

## querySelector(All)

querySelector返回的是一个对象，querySelectorAll返回的一个集合(NodeList)。

querySelectorAll 在文档内找全部符合选择器**描述的节点**包括Element本身  
jQuery(element).find(selector) 在文档内找全部符合选择器描述的节点不包括Element本身

## .className

### 使用

page[ne].className += ' swipe';

### 封装

function addClass(obj, cls){

var obj\_class = obj.className,//获取 class 内容.

blank = (obj\_class != '') ? ' ' : '';//判断获取到的 class 是否为空, 如果不为空在前面加个'空格'.

**added = obj\_class + blank + cls;//组合原来的 class 和需要添加的 class.**

obj.className = added;//替换原来的 class.

}

function removeClass(obj, cls){

var obj\_class = ' '+obj.className+' ';//获取 class 内容, 并在首尾各加一个空格. ex) 'abc bcd' -> ' abc bcd '

obj\_class = obj\_class.replace(/(\s+)/gi, ' '),//将多余的空字符替换成一个空格. ex) ' abc bcd ' -> ' abc bcd '

removed = obj\_class.replace(' '+cls+' ', ' ');//在原来的 class 替换掉首尾加了空格的 class. ex) ' abc bcd ' -> 'bcd '

removed = removed.replace(/(^\s+)|(\s+$)/g, '');//去掉首尾空格. ex) 'bcd ' -> 'bcd'

obj.className = removed;//替换原来的 class.

}

function hasClass(obj, cls){

var obj\_class = obj.className,//获取 class 内容.

obj\_class\_lst = obj\_class.split(/\s+/);//通过split空字符将cls转换成数组.

x = 0;

for(x in obj\_class\_lst) {

if(obj\_class\_lst[x] == cls) {//循环数组, 判断是否包含cls，可以用**indexOf(cls)**

return true;

}

}

return false;

}

## setAttribute

整体设置。

element.setAttribute('style', 'height: 100px !important');

element.style = " height: 100px !important "; //这个不支持ios9, ios10。

部分属性（支持直接写在节点上的属性）

element.setAttribute('height', 100);

## obj.style

var obj = document.getElementById("btnB");

obj.style.cssText ="background-color:black; display:block;color:White;

### obj.style.xxx/setProperty

第一种：

element.style.height = '100px';

注意要写成驼峰形式。

第二种：

element.style.setProperty('height', '300px', 'important');

### cssText

element.style.cssText = 'height: 100px !important';

### js添加节点

styleElement.appendChild(document.createTextNode(str))

传入string，通过添加文本节点的方式。

### addRule/insertRule

// 在原有样式操作

document.styleSheets[0].addRule('.box', 'height: 100px');

document.styleSheets[0].insertRule('.box {height: 100px}', 0);

// 或者插入新样式时操作

var styleEl = document.createElement('style'),

styleSheet = styleEl.sheet;

styleSheet.addRule('.box', 'height: 100px');

styleSheet.insertRule('.box {height: 100px}', 0);

document.head.appendChild(styleEl);

# 其他对象

## CSSStyleSheet

通过document.stylesheets或者style节点的sheet属性获取。

https://developer.mozilla.org/zh-CN/docs/Web/API/CSSStyleSheet