

3组和大小，位置，相关的属性

offset client scroll,

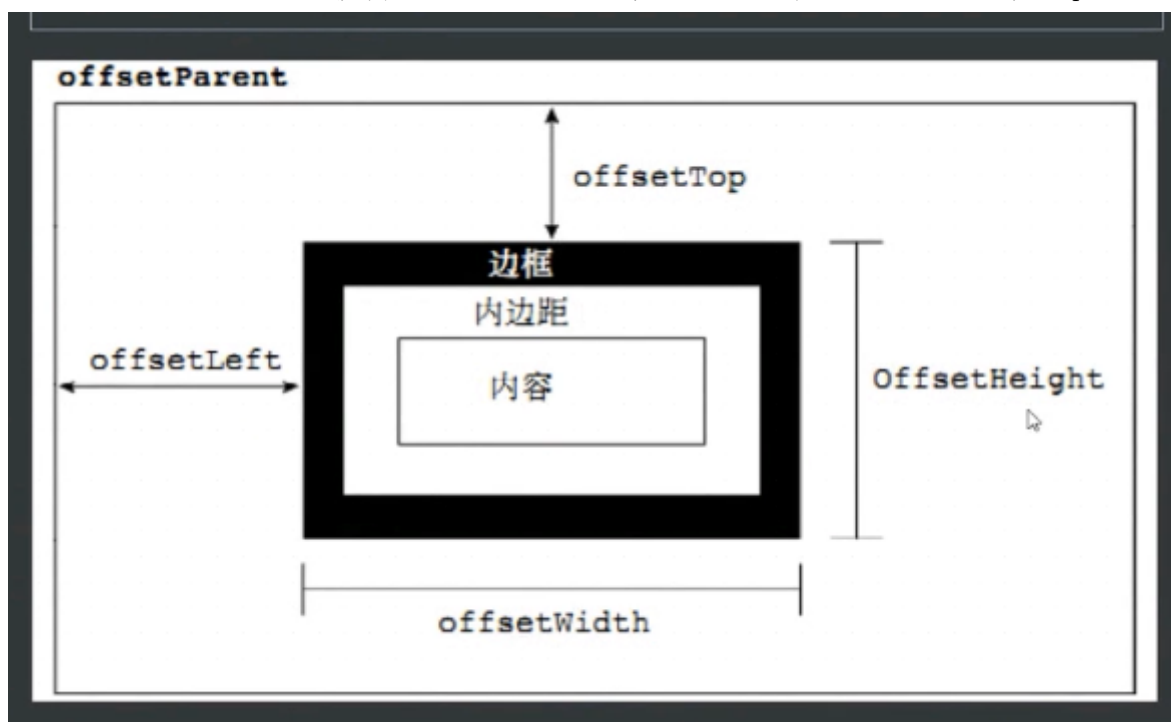
offsetTop 距离有定位的父级元素的top

offsetLeft 距离有定位的父级元素的左边有多少

offsetWidth 盒子的大小

offsetHeight 盒子的大小 这个大小是包括边框和padding的

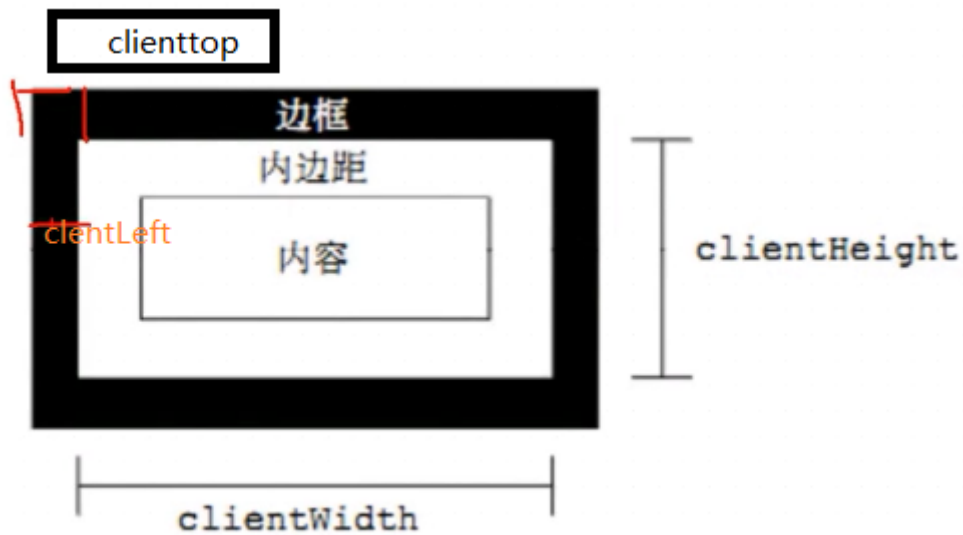
box.offsetParent 获得有定位的父级元素是谁 这个大小是包括边框和padding的



clientLeft获取的是边框的宽度 左边框的宽度

clienttop获取的是边框的宽度 上边框的宽度

clientheight 包括内容和padding 不包括边框

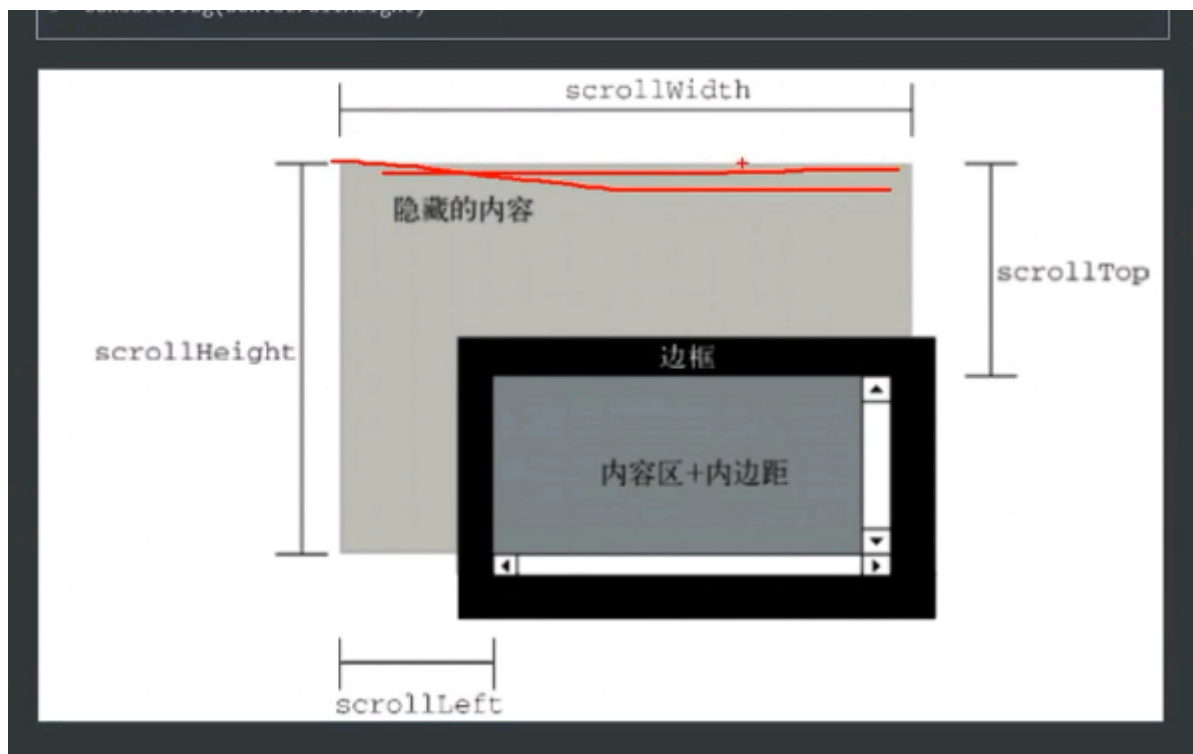


scrollWidth/scrollHeight :

内容的 包括padding 和 未显示的内容，不包括

scrollTop:内容滚动出去的距离

scrollLeft滚动条



onscroll事件:

当拖动滚动条的时候

```

var box = document.getElementById('box');
//获取盒子在页面上的距离

box.onclick = function (e) {
    console.log(this.offsetLeft);
    console.log(this.offsetTop);
    //获取鼠标在盒子中的位置 = 鼠标的位置 - 盒子的坐标
    var boxLeft = getPage(e).pageX - this.offsetLeft;
    var boxTop = getPage(e).pageY - this.offsetTop;
    console.log("盒子左边"+boxLeft);
    console.log("盒子上边"+boxTop);
}

```

```

//pageY = clientY + 页面滚动出去的距离
// 获取滚动出去的距离有兼容性问题：
// 兼容函数
function getScroll() {
    var scrollLeft = document.documentElement.scrollLeft || document.body.scrollLeft;
    var scrollTop = document.documentElement.scrollTop || document.body.scrollTop;
    return {
        scrollLeft: scrollLeft,
        scrollTop: scrollTop
    }
}

//获取鼠标在页面的位置 处理浏览器兼容问题
function getPage(e) {
    var pageX = e.pageX || e.clientX + getScroll().scrollLeft;
    var pageY = e.pageY || e.clientY + getScroll().scrollTop;
    return {
        pageX : pageX,
        pageY : pageY
    }
}

```