

盒子模型十三个属性

style

client 系列

offset系列

案例： client offset scroll 宽高 案例一样

scroll系列（内容+溢出）不加边框

scrollLeft / scrollTop 绝对位置案例（可以做模板）

盒子模型十三个属性

style

- style 为行内样式，样式表中的样式，style是获取不出来的。
 - 系统自己默认的是获取不出来的，自己写的可以获取出来，
 - 要获取（系统默认）样式表中的样式可以使用：getComputedStyle().attr 获取计算后的样式(获取修改后生效的样式)
- 注意：getComputedStyle 获取的值是带单位的（例如：100px）使用的时候需要转数字
- 小细节：有时候需要隐藏某个元素，但是又要存一下当前的元素的尺寸，这个时候就使用getComputedStyle
 - 例如：getComputedStyle(box).height （获取box的高）
 - let {width, height }=getComputedStyle(box)
 -
 -
 - attr () 方法设置或返回被选元素的属性值
 - attr(attribute) attribute 规定要获取其值的属性名。

client 系列

- 下面获取的属性不带单位的数字
- 支持盒子可视的宽高padding，不支持border
- clientWidth 内边宽度（内容+padding）
- clientHeight 内边高度（内容+padding）
- clientLeft 左边框宽度
- clientTop 左边框高度

offset系列

- 下面获取的属性不带单位的数字
- 支持盒子可视的宽高padding, 支持border
- offsetWidth (边框+内容的宽度)
- offsetHeight (边框+内容的高度)
-
- offsetParent 计算偏移量时候的参照物是父级, (定位看是哪个父级)
- offsetLeft/ offsetTop (当前元素的左外边框到父级的左内边框的距离)
 - 注意: 在低版本的IE版本下要触发hasLayout属性, 使用起来才不会出问题
 -

案例: client offset scroll 宽高 案例一样

```
<style>
#txt{
  width:100px;
  height:50px;
  border:1px solid #000;
}
</style>
</head>
<body>
  <!-- <textarea id="txt"></textarea> -->
  <div id="txt">dsadsdsadsadsadsa
    dsadsdsadsadsadsad
    sadsdsadsadsadsadsad
    sdsadsadsadsads
    adsdsadsadsadsadsa
    dsdsadsadsadsadsa
  </div>
  <button id="btn">获取txt的高度</button>
<script>
  btn.onclick = function(){
    console.log(txt.clientHeight); // 50 内容的高度, 不加文本框
    console.log(txt.offsetHeight); // 52 文本框的高度加上上下边框的
    高度
    console.log(txt.scrollHeight); //被内容撑开的高度, 不算边框的
  }
</script>
</body>
```

scroll系列 (内容+溢出) 不加边框

- 下面获取的属性不带单位的数字
- scrollWidth 如果内容不溢出, 则数值 = clientWidth

- 如果内容溢出，则数值 = clientWidth + padding的值
- scrollHeight 如果内容不溢出，则数值 =clientHeight
- 如果内容溢出，则数值 =clientHeight + padding的值
- scrollLeft 获取左右滚动条滚动的距离
- scrollTop 获取上下滚动条滚动的距离

注意： scrollLeft / scrollTop 在复杂的页面情况下，要慎用，获取的距离是根据定位父级来的

- 绝对位置才是最稳的一个距离（当前元素到页面顶部的距离）

scrollLeft / scrollTop 绝对位置案例（可以做模板）

复杂的页面情况下滚动条的定位父级用这个

思路：求绝对位置，就是把当前元素所有的祖先节点距离都计算一遍，求和即可

具体实施：

- 1.先定义一个元素（作为当前元素）
- 2.先求当前元素的距离（包含定位距离和边框距离）
- 3.把这次的定位父级变成下一次循环的当前

```
<style>
div{
    padding:100px;
}
#box{
    background: red;
    _position: relative;
}
#box1{
    background: green;
    position: relative;
    border:50px solid #000;
}
#box2{
    background: skyblue;
    position: absolute;
    top:150px;
    border:50px solid #000;
}
</style>
</head>
<body>
    <div id="box">
        <div id="box1">
            <div id="box2"></div>
```

```

    </div>
  </div>
<script>

let nowEle = box2;
let i=0;
while( nowEle ){ //只要父级里有这个元素就一直循环
    //当前元素的左边框值加上到父级左内边框的距离
    i += ( nowEle.offsetLeft + nowEle.clientLeft );
    //把上一轮的父级当成当前元素再循环
    //用nowEle不用i的原因是，下次循环使用父级当成当前元素，不是值
    nowEle = nowEle.offsetParent;
}
//减去box2的左边距的值，因为当时为了循环用了，最后计算要除去
i = i-box2.clientLeft
console.log(i)

```

1.对于内联样式，可以直接使用`ele.style.属性名`（当然也可以用键值对的方式）获得。注意在CSS中单词之间用-连接，在JS中要用驼峰命名法

```

<div id="dv" style="width: 100px;height: 200px;background-color: pink; border: 1px solid green;"></div>
<script>
    var dv = document.getElementById("dv");
    console.log(dv.style.width); //100px
    console.log(dv.style["height"]); //200px
    console.log(dv.style.backgroundColor); //pink
    console.log(dv.style.border); //1px solid green
</script>

```

2.对于外联样式表或者头部的`style`标签内的样式表里的样式内容，用上面的方法无法获得

可以使用js中的`window.getComputedStyle(element,pseudoElement)`.属性名的方法获得。获取的样式是元素在浏览器中最终渲染效果的样式。

其中`window.`可以省略

其中`pseudoElement`：可选，伪类元素，当不查询伪类元素的时候可以忽略或者传入 `null`。

```

<style>
    #dv2{
        width: 100px;

```

```

        height: 200px;
        background-color: #0086b3;
        border: 1px solid red;
    }
</style>
<div id="dv2" style="border-color: black"></div>
<script>
    var dv2 = document.getElementById("dv2");
    console.log(dv2.style["height"]); //空值，没有内联该样式无法获取
    console.log(dv2.style.backgroundColor); //空值，没有内联该样式无法获取
    console.log(window.getComputedStyle(dv2,null).width); //100px
    console.log(getComputedStyle(dv2,null).backgroundColor); //rgb(0, 134, 179)
    console.log(getComputedStyle(dv2,null).border); //1px solid rgb(255, 0, 0), 内联样式修改成了黑色，内联的权重更高。
    console.log(getComputedStyle(dv2,null)["border"]); //1px solid rgb(255, 0, 0) 键值对的方式当然也行

</script>

```

3. 修改CSS样式，只能通过`ele.style.`属性名的方式修改CSS样式，不能通过`getComputedStyle()`的方法修改。