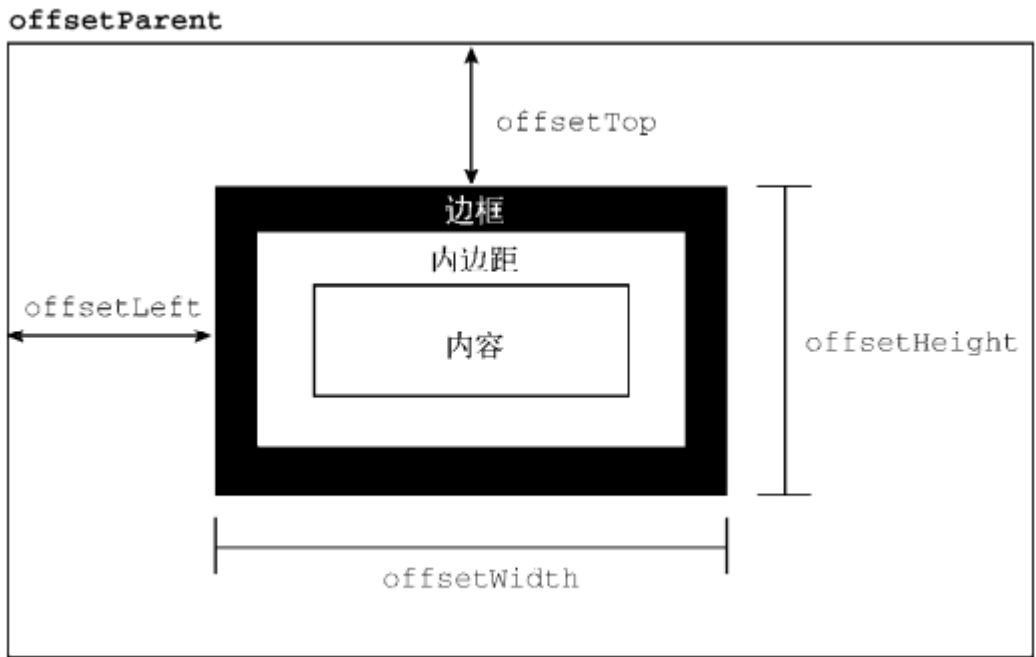


## 一、偏移尺寸

偏移尺寸的计算是相对于【包含元素】，包含元素指 `offsetParent`（`offsetParent` 不一定是 `parentNode`）



如何确定一个元素的offsetParent？

- 元素自身有fixed定位，父元素不存在定位，则offsetParent的结果为null（firefox中为：body，其他浏览器返回为null）
- 元素不是td元素，自身无fixed定位，且父元素也不存在定位，offsetParent为元素
- 元素是td元素，自身无fixed定位，且父元素也不存在定位，offsetParent为< table >元素
- 元素自身无fixed定位，且父元素存在定位，offsetParent为离自身最近且经过定位的父元素（position 为 absolute、relative 或者 fixed）
- <body>元素的offsetParent是null

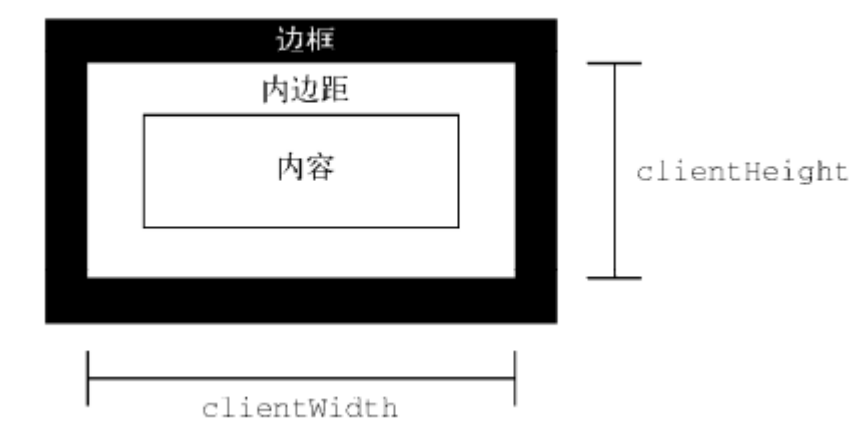
如何确定一个元素在页面中的偏移量？

可以把它的 offsetLeft 和 offsetTop 属性分别与 offsetParent 的相同属性相加，一直加到根元素

```
1 function getElementLeft(element) {
2   let actualLeft = element.offsetLeft;
3   let current = element.offsetParent;
4   while (current !== null) {
5     actualLeft += current.offsetLeft;
6     current = current.offsetParent;
7   }
8   return actualLeft;
9 }
```

## 二、客户端尺寸

元素内容及其内边距所占用的空间， 不包含滚动条占用的空间



```
element.clientWidth = width + paddingLeft + paddingRight
element.clientHeight = height + paddingTop + paddingBottom
```

### 三、 滚动尺寸

**scrollHeight**

没有滚动条时， 等同于内容的总高度（ clientHeight ）

**scrollLeft**

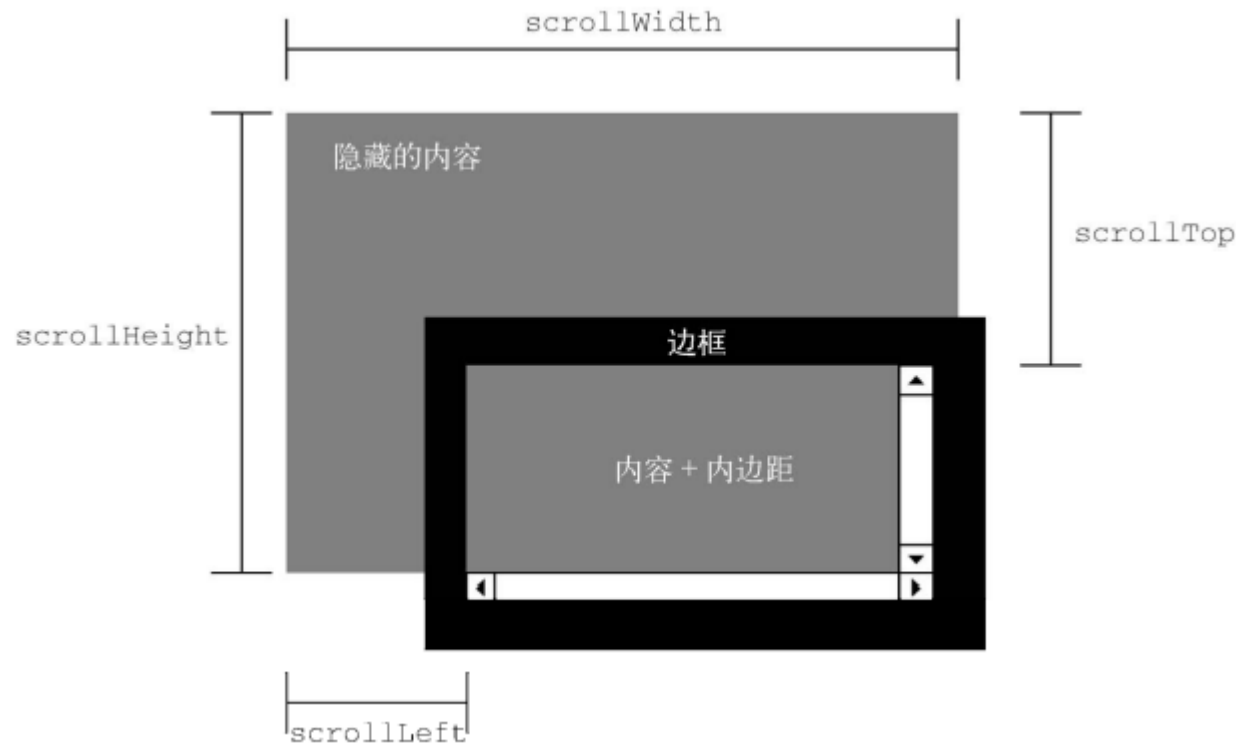
元素左侧的隐藏宽度， 设置这个属性可以改变元素的滚动位置

**scrollTop**

元素顶部的隐藏宽度， 设置这个属性可以改变元素的滚动位置

**scrollWidth**

没有滚动条时， 等同于内容的总宽度（ clientWidth ）



scrollWidth 和 scrollHeight 可以用来确定给定元素内容的实际尺寸。

## 四、 确定元素尺寸

getBoundingClientRect()返回值示意图：

