

## 第 22 章 DOM 元素尺寸和位置

学习要点:

1. 获取元素 CSS 大小
2. 获取元素实际大小
3. 获取元素周边大小

主讲教师: 李炎恢

合作网站: <http://www.ibeifeng.com>

讲师博客: <http://hi.baidu.com/李炎恢>

本章, 我们主要讨论一下页面中的某一个元素它的各种大小和各种位置的计算方式, 以便更好的理解。

### 一. 获取元素 CSS 大小

1. 通过 style 内联获取元素的大小

```
var box = document.getElementById('box'); //获取元素
box.style.width; //200px、空
box.style.height; //200px、空
```

PS: style 获取只能获取到行内 style 属性的 CSS 样式中的宽和高, 如果有获取; 如果没有则返回空。

2. 通过计算获取元素的大小

```
var style = window.getComputedStyle ?
    window.getComputedStyle(box, null) : null || box.currentStyle;
style.width; //1424px、200px、auto
style.height; //18px、200px、auto
```

PS: 通过计算获取元素的大小, 无关你是否是行内、内联或者链接, 它经过计算后得到的结果返回出来。如果本身设置大小, 它会返回元素的大小, 如果本身没有设置, 非 IE 浏览器会返回默认的大小, IE 浏览器返回 auto。

3. 通过 CSSStyleSheet 对象中的 cssRules(或 rules)属性获取元素大小

```
var sheet = document.styleSheets[0]; //获取 link 或 style
var rule = (sheet.cssRules || sheet.rules)[0]; //获取第一条规则
rule.style.width; //200px、空
rule.style.height; //200px、空
```

PS: cssRules(或 rules)只能获取到内联和链接样式的宽和高, 不能获取到行内和计算后的样式。

总结: 以上的三种 CSS 获取元素大小的方法, 只能获取元素的 CSS 大小, 却无法获取元素本身实际的大小。比如加上了内边距、滚动条、边框之类的。

## 二. 获取元素实际大小

### 1.clientWidth 和 clientHeight

这组属性可以获取元素可视区的大小，可以得到元素内容及内边距所占据的空间大小。

```
box.clientWidth; //200
```

```
box.clientHeight; //200
```

PS: 返回了元素大小，但没有单位，默认单位是 px，如果你强行设置了单位，比如 100em 之类，它还是会返回 px 的大小。(CSS 获取的话，是照着你设置的样式获取)。

PS: 对于元素的实际大小，clientWidth 和 clientHeight 理解方式如下：

- 1.增加边框，无变化，为 200；
- 2.增加外边距，无变化，为 200；
- 3.增加滚动条，最终值等于原本大小减去滚动条的大小，为 184；
- 4.增加内边距，最终值等于原本大小加上内边距的大小，为 220；

PS: 如果说没有设置任何 CSS 的宽和高度，那么非 IE 浏览器会算上滚动条和内边距的计算后的大小，而 IE 浏览器则返回 0。

### 2.scrollWidth 和 scrollHeight

这组属性可以获取滚动内容的元素大小。

```
box.scrollHeight; //200
```

```
box.scrollWidth; //200
```

PS: 返回了元素大小，默认单位是 px。如果没有设置任何 CSS 的宽和高度，它会得到计算后的宽度和高度。

PS: 对于元素的实际大小，scrollWidth 和 scrollHeight 理解如下：

- 1.增加边框，不同浏览器有不同解释：
  - a) Firefox 和 Opera 浏览器会增加边框的大小，220 x 220
  - b) IE、Chrome 和 Safari 浏览器会忽略边框大小，200 x 200
  - c) IE 浏览器只显示它本来内容的高度，200 x 18
- 2.增加内边距，最终值会等于原本大小加上内边距大小，220 x 220，IE 为 220 x 38
- 3.增加滚动条，最终值会等于原本大小减去滚动条大小，184 x 184，IE 为 184 x 18
- 4.增加外边距，无变化。
- 5.增加内容溢出，Firefox、Chrome 和 IE 获取实际内容高度，Opera 比前三个浏览器获取的高度偏小，Safari 比前三个浏览器获取的高度偏大。

### 3.offsetWidth 和 offsetHeight

这组属性可以返回元素实际大小，包含边框、内边距和滚动条。

```
box.offsetWidth; //200
```

```
box.offsetHeight; //200
```

PS: 返回了元素大小，默认单位是 px。如果没有设置任何 CSS 的宽和高度，他会得到计算后的宽度和高度。

PS: 对于元素的实际大小，offsetWidth 和 offsetHeight 理解如下：

- 1.增加边框，最终值会等于原本大小加上边框大小，为 220;
- 2.增加内边距，最终值会等于原本大小加上内边距大小，为 220;
- 3.增加外边据，无变化;
- 4.增加滚动条，无变化，不会减小;

PS: 对于元素大小的获取，一般是块级(block)元素并且以设置了 CSS 大小的元素较为方便。如果是内联元素(inline)或者没有设置大小的元素就尤为麻烦，所以，建议使用的时候注意。

### 三. 获取元素周边大小

#### 1.clientLeft 和 clientTop

这组属性可以获取元素设置了左边框和上边框的大小。

```
box.clientLeft;           //获取左边框的长度  
box.clientTop;           //获取上边框的长度
```

PS: 目前只提供了 Left 和 Top 这组，并没有提供 Right 和 Bottom。如果四条边宽度不同的话，可以直接通过计算后的样式获取，或者采用以上三组获取元素大小的减法求得。

#### 2.offsetLeft 和 offsetTop

这组属性可以获取当前元素相对于父元素的位置。

```
box.offsetLeft;          //50  
box.offsetTop;          //50
```

PS: 获取元素当前相对于父元素的位置，最好将它设置为定位 position:absolute; 否则不同的浏览器会有不同的解释。

PS: 加上边框和内边距不会影响它的位置，但加上外边据会累加。

```
box.offsetParent;       //得到父元素
```

PS: offsetParent 中，如果本身父元素是<body>，非 IE 返回 body 对象，IE 返回 html 对象。如果两个元素嵌套，如果上父元素没有使用定位 position:absolute，那么 offsetParent 将返回 body 对象或 html 对象。所以，在获取 offsetLeft 和 offsetTop 时候，CSS 定位很重要。

如果说，在很多层次里，外层已经定位，我们怎么获取里层的元素距离 body 或 html 元素之间的距离呢？也就是获取任意一个元素距离页面上的位置。那么我们可以编写函数，通过不停的向上回溯获取累加来实现。

```
box.offsetTop + box.offsetParent.offsetTop; //只有两层的情况下
```

如果多层的话，就必须使用循环或递归。

```
function offsetLeft(element) {  
    var left = element.offsetLeft;           //得到第一层距离  
    var parent = element.offsetParent;       //得到第一个父元素  
  
    while (parent !== null) {                 //如果还有上一层父元素
```

```
left += parent.offsetLeft;           //把本层的距离累加
parent = parent.offsetParent;       //得到本层的父元素
}                                     //然后继续循环
return left;
}
```

### 3.scrollTop 和 scrollLeft

这组属性可以获取滚动条被隐藏的区域大小，也可设置定位到该区域。

```
box.scrollTop;                       //获取滚动内容上方的位置
box.scrollLeft;                      //获取滚动内容左方的位置
```

如果要想滚动条滚动到最初始的位置，那么可以写一个函数：

```
function scrollStart(element) {
    if (element.scrollTop != 0) element.scrollTop = 0;
}
```

# 感谢收看本次教程！

本课程是由北风网([ibeifeng.com](http://ibeifeng.com))

瓢城 **Web** 俱乐部([yc60.com](http://yc60.com))联合提供：

本次主讲老师：李炎恢

我的博客：[hi.baidu.com/李炎恢/](http://hi.baidu.com/李炎恢/)

我的邮件：[yc60.com@gmail.com](mailto:yc60.com@gmail.com)