



传智播客 · 前端与移动开发学院

<http://web.itcast.cn>

前端基本功—javascript 第八天

目录

目录.....	2
1.2 复习.....	3
1.3 offset 家族.....	3
1.3.1 offsetWidth offsetHeight.....	4
1.3.2 offsetLeft offsetTop.....	4
1.3.3 offsetParent.....	6
1.3.4 offsetTop style.top 的区别.....	6
1.4 事件对象.....	7
1.4.1 event 常见属性.....	8
1.4.2 pageX clientX screenX 区别.....	9
1.5 常用事件.....	10

1.1 复习

1. 根据位置返回 字符 `asdfg.charAt(3)` f

根据字符返回位置 `asdfg.indexOf("f")` 3 从前面索引

`lastIndexOf("g")` 5 从后面索引

2. `slice`(起始位置, [结束位置])

`slice(3)` 从第 3 个开始选 `slice(3,6)` 从索引号 3 开始取, 数 6

个 但是从索引号 0 开始数

```
var arr = [1,3,5,7];
```

```
asdfghidk    fgh
```

`substr(3,6)` 从 3 开始选 从第三个开始数 往后 数 6 个 fghidk

```
toFixed(2)
```

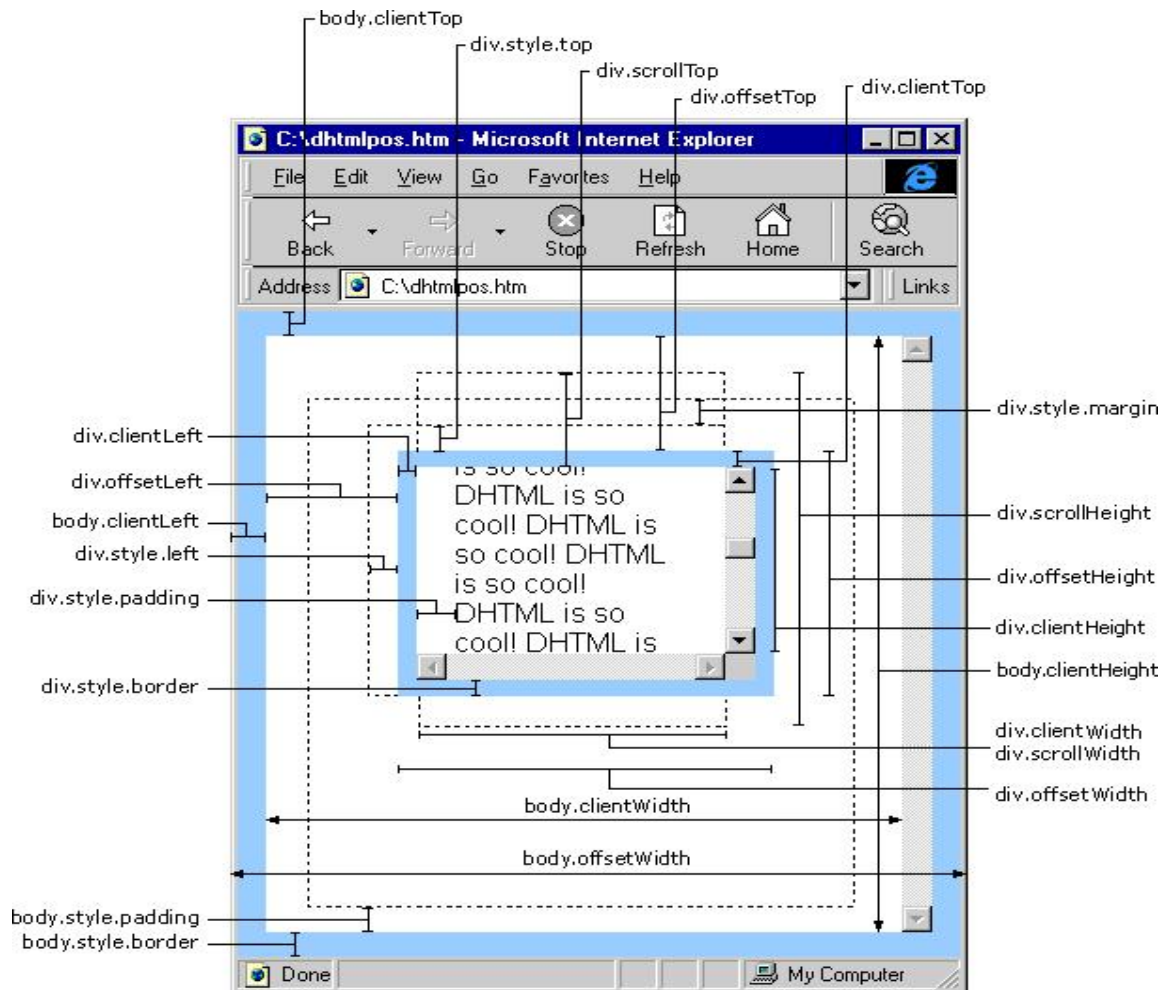
缓动公式 :

$$\text{leader} = \text{leader} + (\text{target} - \text{leader}) / 10$$

1.2 offset 家族

`offset` 自己的

目的: js 中有一套方便的获取元素尺寸的办法就是 `offset` 家族;



1.2.1 offsetWidth offsetHeight

得到对象的宽度和高度(自己的，与他人无关)

$\text{offsetWidth} = \text{width} + \text{border} + \text{padding}$

```
div { width:220px; border-left:2px solid red; padding:10px;}
```

$\text{div.offsetWidth} = 220 + 2 + 20$

为什么不用 `div.style.width` 因为东西 只能得到行内的数值

1.2.2 offsetLeft offsetTop

返回距离上级盒子（最近的带有定位）左边的位置

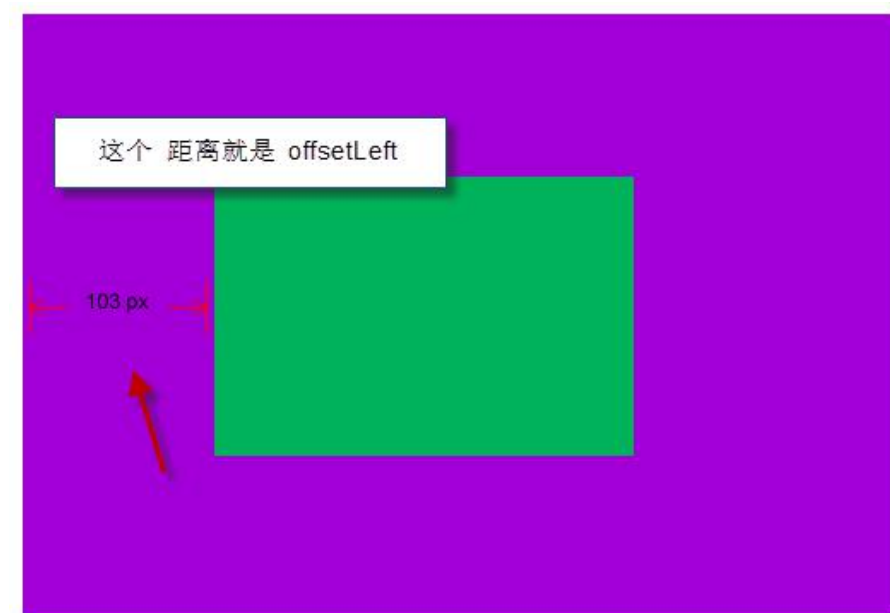
如果父级都没有定位则以 `body` 为准

这里的父级指的是所有上一级 不仅仅指的是 父亲 还可以是 爷爷 曾爷爷 曾曾爷爷。。。。



`offsetLeft` 从父级的 `padding` 开始算 父亲的 `border` 不算

总结一下： 就是子盒子到定位的父盒子边框到边框的距离



1.2.3 offsetParent

返回改对象的父级 （带有定位） 不一定是亲的爸爸

前面学过一个返回父亲(亲的) `parentNode` 有所区别

如果当前元素的父级元素没有进行 CSS 定位（`position` 为 `absolute` 或 `relative`），`offsetParent` 为 `body`。

2、如果当前元素的父级元素中有 CSS 定位(`position` 为 `absolute` 或 `relative`)，`offsetParent` 取最近的那个父级元素。

```
var son = document.getElementById("son");  
//alert(son.parentNode.id);  
alert(son.offsetParent.tagName); // tagName 标签的名字
```

1.2.4 offsetTop style.top 的区别

一、最大区别在于 `offsetLeft` 可以返回没有定位盒子的距离左侧的位置。
而 `style.top` 不可以 只有定位的盒子 才有 `left top right`

二、offsetTop 返回的是数字，而 style.top 返回的是字符串，除了数字外还带有单位：px。

```
style.left = 300px      parseInt (300px)      结果  300  
parseInt (style.left) + parseInt (style.left)
```

三、offsetTop 只读，而 style.top 可读写。

四、如果没有给 HTML 元素指定过 top 样式，则 style.top 返回的是空字符串。

五、最重要的区别 style.left 只能得到 行内样式 offsetLeft 随便

1.3 事件对象

我们学过一些事件： onmouseover onmouseout onclick

```
btn.onclick = function(event) { 语句 }
```

event 单词翻译过来 事件 的意思

event 就是事件的对象 指向的是 事件 是 onclick

再触发 DOM 上的某个事件时，会产生一个事件对象 event，这个对象中包含着所有与事件有关的信息。所有浏览器都支持 event 对象，但支持的方式不同。

比如鼠标操作时候，会添加鼠标位置的相关信息到事件对象中。

普通浏览器支持 event

ie 678 支持 window.event

所以我们 采取兼容性的写法：

```
var event = event || window.event;
```

1.3.1 event 常见属性

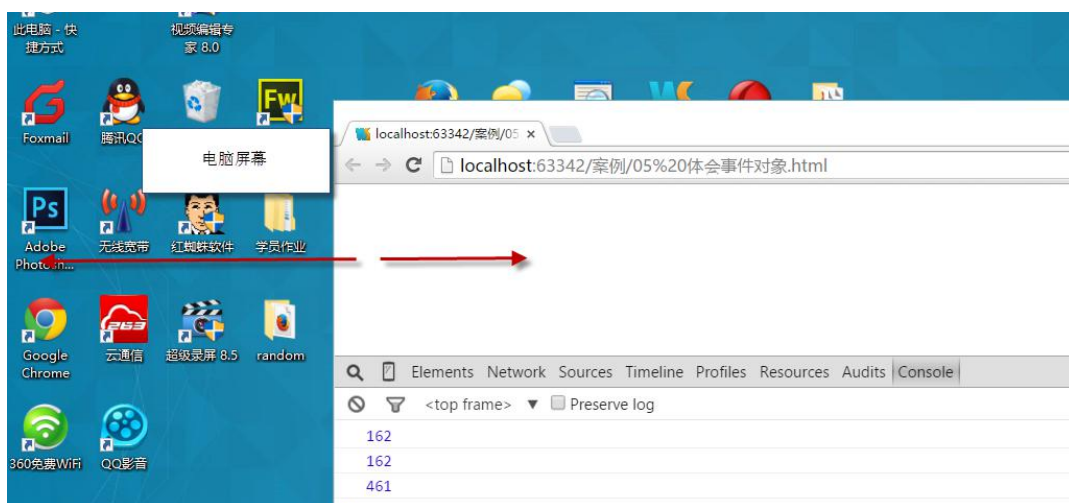
属性	作用
data	返回拖拽对象的 URL 字符串（dragDrop）
width	该窗口或框架的高度
height	该窗口或框架的高度
pageX	光标相对于该网页的水平位置（ie 无）
pageY	光标相对于该网页的垂直位置（ie 无）
screenX	光标相对于该屏幕的水平位置
screenY	光标相对于该屏幕的垂直位置
target	该事件被传送到的对象

type	事件的类型
clientX	光标相对于该网页的水平位置（当前可见区域）
clientY	光标相对于该网页的水平位置

1.3.2 pageX clientX screenX 区别

screen X screenY

是以我们的电脑屏幕 为基准点 测量



pageX pageY

以我们的 文档 （绝对定位） 的基准点 对齐

ie678 不认识

clientX clientY

以 可视区域 为基准点 类似于 固定定位

1.4 常用事件

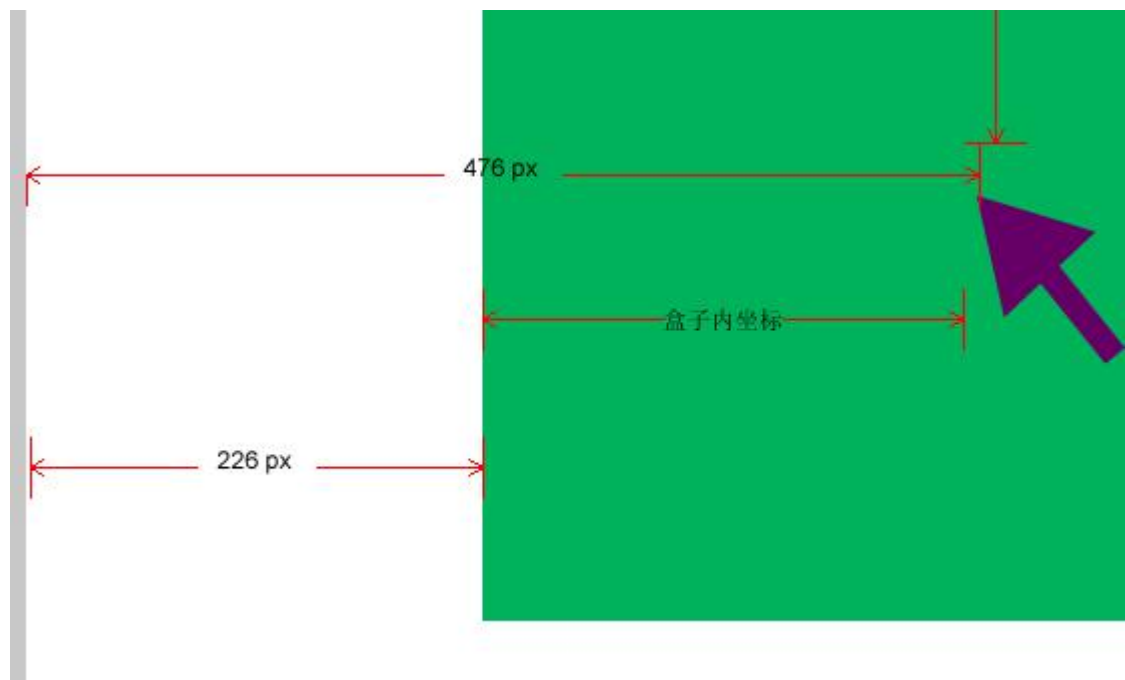
`onmouseover` `onmouseout` `onclick`

`onmousemove` 当鼠标移动的时候 就是说，鼠标移动一像素就会执行的事件

```
div.onmousemove = function() { 语句 }
```

当鼠标再 `div` 身上移动的时候，就会执行。

得到在某个盒子内的坐标：



`div.onmouseover` 和 `div.onmousemove` 区别

他们相同点 都是 经过 `div` 才会触发

`div.onmouseover` 只触发一次

`div.onmousemove` 每移动一像素，就会触发一次

onmouseup 当鼠标弹起

onmousedown 当鼠标按下的时候

1. 拖动 原理 == 鼠标按下 接着 移动鼠标 。

```
bar.onmousedown = function(){  
    document.onmousemove = function(){  
    }  
}
```

2. 当我们按下鼠标的时候，就要记录当前 鼠标 的位置 - 大盒子的位置
算出 bar 当前 在 大盒子内的距离 。

1.4.1 防止选择拖动

我们知道 按下鼠标然后拖拽可以选择文字 的。

清除选中的内容

```
window.getSelection ? window.getSelection().removeAllRanges() :  
document.selection.empty();
```

