# 响应函数和for的i

当页面中有多个超链接，都需要绑定单击响应函数，这时可以通过for循环来为每一个超链接去绑定单击响应函数，这里在页面加载完毕以后for循环会立即执行，而响应函数需要在事件触发以后才执行，响应函数执行时，for循环早就执行完了，所以在响应函数中，访问到的i一定是for循环执行的最后一次的i，而不能访问到当前元素的索引

# 样式的操作

## 修改元素的样式

\* 语法：元素.style.样式名 = 样式值; 样式值为字符串

\* 注意样式名中带有-不能直接写

\* 像

\* background-color

\* border-width

\* border-left-style

\* - 这种带有-的样式名，需要将其修改为驼峰命名法，然后再操作

\* 将-去掉，然后将-后的字母改大写

\* backgroundColor

\* borderWidth

\* borderLeftStyle

\* float由于是JS中的保留字，所以不能直接写，需要在前面加css，并使用驼峰命名法，即cssFloat

\* - 通过style修改的样式都是元素的内联样式，而内联样式的优先级最高的，所以我们通过JS修改的样式往往会立即显示，但是如果在样式后添加了!important，则无法通过内联样式进行覆盖，所以我们要尽量去避免使用!important

## 读取元素的样式

### 元素.style.样式名

\* - 通过style只可以读取内联样式，不能读取样式表中的样式

\* 如果没有内联样式则不能读取

读取元素的当前生效的样式

### 元素.currentStyle.样式名

\* currentStyle是一个对象，在这个对象中封装了元素当前的样式

\* currentStyle只有IE浏览器支持，在其他的浏览器中不好使

正常浏览器需要使用一个getComputedStyle()这个方法来获取当前的样式

\* 这个方法就是window的方法，可以直接使用

\* 参数：

\* 1.要获取样式的元素

\* 2.要获取的元素的伪类，如果不需要可以传null

### getComputedStyle(元素,null).样式名

\* - 这个方法会返回一个对象，在这个对象中封装了元素的当前样式

\* - 注意此方法IE8及以下的浏览器不支持

\* - 通过currentStyle和getComputedStyle()读取到的样式都是**只读**的，不能修改，如果需要修改，则需要使用style

为了可以兼容所有的浏览器，我们需要自定义一个函数，

### 用来获取任意元素的当前样式

\* 参数：

\* obj 要获取样式的元素

\* name 要获取的样式的名字

function getStyle(obj , name){

//判断浏览器中是否含有getComputedStyle这个方法

if(window.getComputedStyle){

//支持正常的浏览器

return getComputedStyle(obj,null)[name];

}else{

//只支持IE

return obj.currentStyle[name];

}

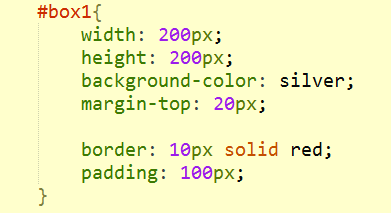
}

getComputedStyle前加window：当不加window时，IE中没有getComputedStyle这个方法，会报错，当加了window时，不会报错，会返回undefined，undefined转换为布尔值是false，会执行else中的语句

或者使用三元运算符return window.getComputedStyle ? getComputedStyle(obj,null)[name] : obj.currentStyle[name]; （空格只是为了让语句看的清楚）

在读取元素的样式时，如果元素没有设置样式，则火狐、Chrome等浏览器会自动计算元素的样式值，而IE浏览器，有时候不会自动计算，而是返回默认值，比如宽度会返回auto，所以尽量获取已有的样式，如果没有就写一个

为box1设置样式：



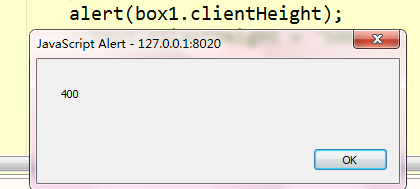
### clientWidth

### clientHeight

\* - 获取元素的可见宽度和高度

\* - 获取到的值包括元素的内容区和内边距

\* - 这些属性都是只读，只能看不能修改



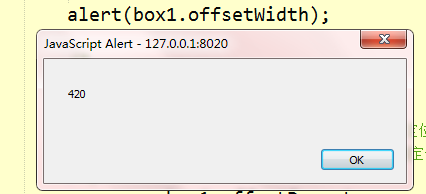
### offsetWidth

### offsetHeight

\* - 可以用来获取元素整个的可见框的大小

\* - 它获取到的值，包括内容区、内边距和边框

\* 只读

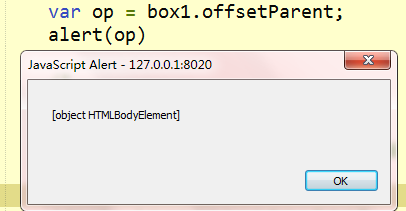


### offsetParent

\* - 可以返回当前元素的定位父元素

\* - 定位父元素，指离当前元素最近的开启了定位的祖先元素

\* 如果所有的祖先元素都没有开启定位，则返回body



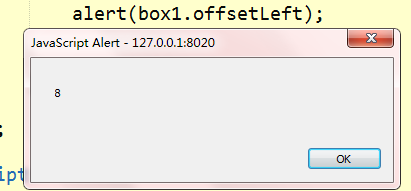
### offsetLeft（水平偏移量）

### offsetTop（垂直偏移量）

\* - 获取当前元素相对于其定位父元素的偏移量

\* - 定位父元素，指离当前元素最近的开启了定位的祖先元素

\* 如果所有的祖先元素都没有开启定位，则返回相对于body的偏移量



8为body默认的内边距

### scrollHeight

\* - 可以获取当前元素滚动区域的高度

### scrollWidth

\* - 可以获取当前元素滚动区域的宽度

### scrollTop

\* - 垂直滚动条滚动的距离

### scrollLeft

\* - 水平滚动条滚动的距离

上面4个一般针对子元素超出父元素的情况，scrollHeight、scrollWidth是给父元素设置的，获取滚动元素的高度、宽度；scrollTop、scrollLeft也是给父元素设置的，获取滚动条距父元素上、左滚动的距离

当 scrollHeight - scrollTop = clientHeight时，则说明垂直滚动条滚动到底了

当 scrollWidth - scrollLeft = clientWidth时，则说明水平滚动条滚动到底了

都是给父元素设置的

利用上面两条，可以设置滚动条滚动到底后的触发事件，例如阅读协议完后，才能点注册

onscroll

\* - 滚动条滚动的事件，当元素的滚动条滚动时会触发

disabled属性用来设置表单项是否禁用

设置为true，表示禁用，设置为false表示可用