endOffset:范围在endContainer中终点的偏移量（与startOffset遵循相同的取值规则）

一直不知道怎么算！之前包括元素节点的时候，就假装range.endOffset的值是子元素索引数+1或者直接从1（而不是0）数到所需子元素，但是也是知其然而不知其所以然。

现在正儿八经的选文本节点的范围，这个问题就又来了，结果在我苦思冥想之际，竟然发现计算endOffset的真正（更好用的）方法！

比如，这样一个P元素节点中包含了一个hello的文本节点，如果我取

range1.setStart(p1Text,1);  
range1.setEnd(p1Text,4);

会如何?

HTML：

<**p** id=**"p1"**>**hello**</**p**>

js:

var range1 = document.createRange();  
var p1 = document.getElementById("p1");  
var p1Text = p1.firstChild;  
range1.setStart(p1Text,1);  
range1.setEnd(p1Text,4);  
alert(range1)//ell

这样一看，我觉得startOffset和endOffset应该这么理解（先卖个关子让我再分析分析），先放上书中原句

startOffset:范围在startContainer中起点的偏移量。如果startContainer是文本节点、注释节点或CDATA节点，那么startOffset就是范围起点之前跳过的字符数量。否则，startOffset就是范围中第一个子节点的索引（注意索引是从0开始算的）。

重点是：那么startOffset就是范围起点之前跳过的字符数量。

字符数量指的应该是是从1开始而不是0，但是红宝书335页又是从0开始标记的，所以还是以实践的335页为标准吧。

正式写我的理解：

我之后是这么理解的，其实在range这一章，不应该以一个字符为一个索引（这么说有点歧义，总之我认为在range这一章不应该使用索引这个词，感觉这与真正想报答的概念有点不太一样），范围的索引应该是和slice（）方法类似的，是在字符前后的，如：

0 h 1 e 2 l 3 l 4 o 5

这样就很好理解了，范围就是用开始和结尾两个界限框起来的范围嘛，那么我要找一个1到4的范围还不容易?

很显然是1 e 2 l 3 l 4

去掉这些人为标注的1 2 3 4，答案不就是ell了嘛

而包含元素子节点的也可以这么算，但是要注意一下换行符的问题。

比如：html

<**p** id=**"p1"**>  
 <**b**></**b**><**span**></**span**>  
</**p**>

js:

var range1 = document.createRange();  
var p1 = document.getElementById("p1");  
range1.selectNodeContents(p1)  
alert(range1.endOffset)//4  
alert(range1.startOffset)//0

范围框定的是从p元素的开始标签的右边括号>到p元素的结束标签的左边括号<

一共框定了这些东西：

一个换行符、b元素、span元素、一个换行符

写成前面文本的形式就是

0 换行符 1 b元素 2 span元素 3 换行符 4

所以是startOffset和endOffset返回0和4没毛病啊！的确是0和4把这些东西框了进去！