为了更精确的控制将哪些节点包含在范围中，还可以使用下列方法。

setStartBefore(refNode):将范围的起点设置在refNode之前，因此refNode也就是范围选区中的第一个子节点。同时会将startContainer属性设置为refNode.prentNode,将startOffset属性设置为refNode在其父节点的childNodes集合中的索引。

setStartAfter(refNode): 将范围的起点设置在refNode之后，因此refNode也就不在范围之内了，其下一个同辈节点才是范围选区中的第一个子节点。同时会将startContainer属性设置为refNode.prentNode,将startOffset属性设置为refNode在其父节点的childNodes集合中的索引加1。（因为第一个节点的索引是refNode的索引+1）。

setEndBefore(refNode):将范围的终点设置在refNode之前，因此refNode也就不在范围内了，上一个同辈节点才是范围选取中的最后一个子节点。同时会将endContainer属性设置为refNode.prentNode,将endOffset属性设置为refNode在其父节点的childNodes集合中的索引。

setEndAfter(refNode):将范围的终点设置在refNode之前，因此refNode也就是范围选区中的最后一个子节点。同时会将endContainer属性设置为refNode.prentNode,将endOffset属性设置为refNode在其父节点的childNodes集合中的索引加1（记成子元素的数量也可以）。

一些测试，和之前学的selectNode()以及selectNodeContexts()方法联合起来用

var range1 = document.createRange();  
var p1 = document.getElementById("p1");  
var p0 = document.getElementById("p0");  
var b = document.getElementById("b")  
var pOuter = document.getElementById("pOuter");  
alert(range1);//啥都没有  
range1.selectNode(p1);  
alert(range1);//some paragraphs Hello world!  
alert(range1.startContainer);//[object HTMLDivElement]  
range1.setStartBefore(b);  
alert(range1.startContainer);//[object HTMLParagraphElement]  
alert(range1.endContainer);//[object HTMLDivElement]  
alert(range1);//Hello world!  
range1.setStartAfter(b);  
alert(range1.startContainer)//[object HTMLParagraphElement]亲测就算把world!这个文本节点删除了，还是返回一样的结果，原因setStartAfter的介绍说明了。  
alert(b.nextSibling);//[object Text],删除world!显示null  
alert(range1);//world!亲测如果把world!这个文本节点删除了，就什么都不alert了。  
range1.setStartBefore(p1);  
alert(range1.startContainer);//[object HTMLDivElement]  
alert(range1)//some paragraphs Hello world!  
range1.setEndAfter(p1);  
alert(range1.endOffset)// 4。 因为有很多个空格算作文本节点，所以是4；如果把换行都删除，则是2，是父元素div的子节点数量。

range1.setStartBefore(pOuter);  
alert(range1.startContainer)// Argument 1 of Range.setStartBefore is not an object.

单独使用这几个方法的测试

var range1 = document.createRange();  
var p1 = document.getElementById("p1");  
var p0 = document.getElementById("p0");  
var b = document.getElementById("b")  
var pOuter = document.getElementById("pOuter");  
range1.setStartBefore(p1);  
range1.setEndBefore(p1);  
alert(range1);//alert空  
range1.setEndAfter(p1);  
alert(range1);//some paragraphs Hello world!  
alert(range1.endOffset);//4。 因为有很多个换行符也被dom算作文本节点，所以是4；如果把换行都删除，则是2，是父元素div的子节点数量。  
alert(range1.startOffset)//3。因为算上换行符和第一个文本节点，所以是3，如果把换行符都删除，则是1（第一个p0子节点是索引0，p1是索引1）