# IE和FireFox中的childNodes区别!!

Javascript中，相信大家都试过用getElementsByTagName和childNodes来实现对节点的遍历。但是 getElementsByTagName对复杂的DOM结构遍历明显不如用childNodes，因为childNodes能更好的处理DOM的层 次结构,建议在需要进行了遍历时首先使用childNodes!!  
   但是不幸的是，在IE和FireFox中childNodes有点细微的差别:

**<head>**

**<script type="text/javascript">  
    function view(){   
    var childs1=$('FirstDiv').childNodes;  
    var childs2=$('SecondDiv').childNodes;     
    alert("length of FirstDiv: "+childs1.length+"--length of SecondDiv: "+childs2.length);  
    }  
    var $=function(id)  
    { return document.getElementById(id); }  
    </script>  
</head>  
<html >  
<body onload="view();">  
  
   <!--第一个遍历对象，节点之间留有空格和回车-->  
    <div id="FirstDiv">  
    <div>1</div>  
    <div>2</div>  
    <div>3</div>  
    </div>  
  
    <!--第二个遍历对象，除注释外，节点间无空格回车-->  
    <div id="SecondDiv"><div>first</div><div>second</div><div>third</div></div>  
</html>**

用IE和Firefox运行会有两个不同的结果：IE的结果是3:3；而Firefox则是7:3。怎么会有这种情况呢？  
在结构上，对象1和对象2不同的是对象1的子节点间有回车或者空格，而对象2则是一行写到尾。大家都应该想到了吧，IE是将一个完整标签作为一个节点。而 Firefox除了上述的的情况外，也把一个标签的结束符“>”到下一个标签的起始符“<”之间的内容（除注释外，包括任何的文字、空格、回 车、制表符）也算是一个节点了。而且这种节点也有它们自己独特的属性和值－－nodeName="#text"。

在实际运用中，Firefox在遍历子节点时，在for循环里不妨加上：  
if(childNode.nodeName=="#text") continue;  
或者nodeType == 1。  
这样，便跳过不需要的操作，使程序运行的更有效率。

附:

**Node.ELEMENT\_NODE == 1**

**Node.ATTRIBUTE\_NODE == 2**

**Node.TEXT\_NODE == 3**

**Node.CDATA\_SECTION\_NODE == 4**

**Node.ENTITY\_REFERENCE\_NODE == 5**

**Node.ENTITY\_NODE == 6**

**Node.PROCESSING\_INSTRUCTION\_NODE == 7**

**Node.COMMENT\_NODE == 8**

**Node.DOCUMENT\_NODE == 9**

**Node.DOCUMENT\_TYPE\_NODE == 10**

**Node.DOCUMENT\_FRAGMENT\_NODE == 11**

**Node.NOTATION\_NODE == 12**

**例**

**var node = document.documentElement.firstChild;  
if(node.nodeType != Node.COMMENT\_NODE)  
alert("You should comment your code well!");**