**学习目标：**

1、createElement

2、createTextNode

3、appendChild

4、insertBefore

5、removeChild

6、replaceChild

7、cloneNode

演示文档，即留言本，当点击按钮，把内容添加到ul当中。点击li中的删除，要能删除这第留言

<input type="text" id="text1" />

<input type="button" value="留言" id="btn" />

<ul id="ul1"></ul>

之前我们用过innerHTML，用的是累加的方式。如果留言一多，性能就不好了。这时，我们就需要动态创建。

一、document.createElement(标签名称); 创建元素

创建出来的元素同从页面中获取的元素本质上是一样的，页面中的元素能用什么方法，它也能用什么方法。

区别只是：一个已经存在于页面中，一个还只是存在于js内存中，没有在页面中存在的元素。

二、document.createTextNode("文本"); 创建文本节点，创建之后要添加到元素中，作为元素的内容。但是，元素的内容，还是推荐大家用innerHTML来添加。

三、父级.appendChild(要添加的元素) 方法，追加子元素，在父级的最后

四、父级.insertBefore(新元素，被插入的已有元素) 方法，在指定的元素前面插入一个新元素。

注意：在IE8及以下，如果第二个参数节点不存在，就报错。在其它浏览器下，如果第二个参数节点不存在，就以appendChild的方式插入。

所以兼容处理为：

if(ul.children[0] ){

ul.insertBefore(oLi,ul.children[0]);

}else{

ul.appendChild(oLi);

}

五、被删除元素的父级.removeChild(要删除的元素); 删除元素，这个方法返回被删除的元素。被删除的节点仍然存在于内存中，可以对其进行下一步操作。

六、被替换元素的父级.replaceChild(新节点，被替换节点); 替换节点。如果新节点原来就在页面中，还具有剪切替换的作用。返回被替换的元素。

七、被克隆的节点.cloneNode(true); 返回克隆后的节点。参数为true时，把子孙节点也克隆进去了，为空时，只克隆当前这个节点。

注意：appendChild,insertBefore,replaceChild都可以操作动态创建出来的节点，也可以操作已有节点

演示文档：div替换p标签

<div id="div1">div1</div>

<input type="button" value="按钮" id="btn" />

<hr />

<p id="p1">ppppp</p>

实现：

document.body.replaceChild( oDiv, oP );

留言本完整实现

window.onload = function() {

var oText = document.getElementById('text1');

var oBtn = document.getElementById('btn');

var oUl = document.getElementById('ul1');

oBtn.onclick = function() {

var oLi = document.createElement('li');

oLi.innerHTML = oText.value;

var oA = document.createElement('a');

oA.innerHTML = '删除';

oA.href = 'javascript:;';

oA.onclick = function() {

oUl.removeChild( this.parentNode );

}

oLi.appendChild( oA );

if ( oUl.children[0] ) {

oUl.insertBefore( oLi, oUl.children[0] );

} else {

oUl.appendChild( oLi );

}

}

}