dom节点操作：

1、获取节点：

- getElementById　对象
- getElementsByTagName 标签名称　集合
- getElementsByClassName　类名 集合　IE678不支持，需要兼容 取每个element的class，再对传入的买个className用indexOf判断是否存在参数中的className
- querySelector / All 选择器

querySelector：获取第一个符合条件的元素，querySelectorAll：所有符合元素的节点的列表

list不是动态的，一旦获取之后就不会再变化，后面做的修改不会有影响

例子：

```
<table border="10" id="users">
    <tr>
        <td colspan=2 align="center">员工号</td>
    </tr>
    <tr class="user" id="last">
        <td rowspan=2 align="center">学历</td>
        <td align="center">专业</td>
    </tr>
    <tr>
        <td colspan=2 align="center">毕业学校</td>
    </tr>
</table>
```

```
alone = document.querySelector('tr');
console.log(alone);
all = document.querySelectorAll('tr');
console.log(all);

console.log(document.querySelectorAll('#users .user'));
```

```
▶ <tr>…</tr>
▶ [tr, tr, tr, tr, tr, tr#last.user, tr]
▶ [tr#last.user]
```

结果中会有该元素的信息，例如类、id

IE6、7、8不支持quetySelector
node.childNodes 获取子节点列表
node.firstChild
node.lastChild
node.parentNode 获取父节点
node.ownerDocument 获取祖先节点(整个document)
node.previousSibling 返回前一个同胞节点，没有则返回null
node.nextSibling 返回后一个同胞节点
2、判断节点
element.hasChildNodes() 返回true或者false，是否有后代节点
H5的classList，element.classList.contains(className) 判断是否有类
element.contains(child) 判断是否有child这个后代节点
3、创建节点
element = document.createElement(tagName)
element = document.createDocumentFragment(tagName) （为避免频繁刷新DOM，可以先创造代码片段，完成所有节点操作之后统一添加到DOM中）
4、修改节点

- textContent
- innerText
- innerHTML 节点的HTML内容 innerHTML仅建议用于新节点，且内容是可控的，不是用户填写的内容，如果是用户填写的也要确保里面没有标签。

element.textContent：ie9以下不支持
element.textContent，获取节点及其后代节点的文本内容
element.textContent = '' 修改节点的内容
element.innerText：用法一致，不规范，firefox不支持，需要兼容方案。
5、插入节点：

- appendChild
- insertBefore

appendChild：
var achild = element.appendChild(achild) 在指定的元素下追加一个节点
insertBefore
var achild = element.insertBefore(achild，referenceChild) 在指定的元素的指定的子节点前面插入一个节点
插入HTML代码：

node.insertAdjacentHTML(‘beforeBegin’,html)　在该元素之前插入代码

node.insertAdjacentHTML(‘afterBegin’,html)　在该元素的第一个子元素之后插入代码

node.insertAdjacentHTML(‘beforeEnd’,html)　在该元素的最后一个子元素之后插入代码

node.insertAdjacentHTML(‘afterEnd’,html)　在该元素之后插入代码

6、删除节点：

removeChild：

child = document.removeChild(child) 删除element指定的子元素

7、替换节点：

replaceChild(newChild, oldChild)

eplaceChild()接收的两个参数是要插入的节点和要替换的节点，要替换的节点将由这个方法返回并从文档树中移除，同时由要插入的节点占据其位置.

8、复制节点：

cloneNode()

cloneNode方法用于克隆一个节点。它接受一个布尔值作为参数，表示是否执行深复制。在参数为true时，执行深复制，也就是复制节点及整个子节点树。在参数为false的情况下，执行浅复制，即复制节点本身。复制后返回的节点副本属于文档所有，但并没有为它指定父节点。若参数为空，也相当于false