ParentNode 接口，ChildNode 接口

节点对象除了继承 Node 接口以外，还会继承其他接口。`ParentNode`接口表示当前节点是一个父节点，提供一些处理子节点的方法。`ChildNode`接口表示当前节点是一个子节点，提供一些相关方法。

ParentNode 接口

如果当前节点是父节点，就会继承`ParentNode`接口。由于只有元素节点（element）、文档节点（document）和文档片段节点（documentFragment）拥有子节点，因此只有这三类节点会继承`ParentNode`接口。

ParentNode.children

`children`属性返回一个`HTMLCollection`实例，成员是当前节点的所有元素子节点。该属性只读。

下面是遍历某个节点的所有元素子节点的示例。

```javascript
for (var i = 0; i < el.children.length; i++) {
  // ...
}
```

注意，`children`属性只包括元素子节点，不包括其他类型的子节点（比如文本子节点）。如果没有元素类型的子节点，返回值`HTMLCollection`实例的`length`属性为`0`。

另外，`HTMLCollection`是动态集合，会实时反映 DOM 的任何变化。

ParentNode.firstElementChild

`firstElementChild`属性返回当前节点的第一个元素子节点。如果没有任何元素子节点，则返回`null`。

```javascript
document.firstElementChild.nodeName
// "HTML"
```

上面代码中，`document`节点的第一个元素子节点是`<HTML>`。

ParentNode.lastElementChild

`lastElementChild`属性返回当前节点的最后一个元素子节点，如果不存在任何元素子节点，则返回`null`。

```javascript
document.lastElementChild.nodeName
// "HTML"
```

上面代码中，`document`节点的最后一个元素子节点是`<HTML>`（因为`document`只包含这一个元素子节点）。

ParentNode.childElementCount

`childElementCount`属性返回一个整数，表示当前节点的所有元素子节点的数目。如果不包含任何元素子节点，则返回`0`。

```javascript
document.body.childElementCount // 13
```

ParentNode.append()，ParentNode.prepend()

`append`方法为当前节点追加一个或多个子节点，位置是最后一个元素子节点的后面。

该方法不仅可以添加元素子节点，还可以添加文本子节点。

```javascript
var parent = document.body;

// 添加元素子节点
var p = document.createElement('p');
parent.append(p);

// 添加文本子节点
parent.append('Hello');

// 添加多个元素子节点
var p1 = document.createElement('p');
var p2 = document.createElement('p');
parent.append(p1, p2);

// 添加元素子节点和文本子节点
var p = document.createElement('p');
parent.append('Hello', p);
```

注意，该方法没有返回值。

`prepend`方法为当前节点追加一个或多个子节点，位置是第一个元素子节点的前面。它的用法与`append`方法完全一致，也是没有返回值。

ChildNode 接口

如果一个节点有父节点，那么该节点就继承了`ChildNode`接口。

ChildNode.remove()

`remove`方法用于从父节点移除当前节点。

```javascript
el.remove()
```

上面代码在 DOM 里面移除了`el`节点。

ChildNode.before()，ChildNode.after()

`before`方法用于在当前节点的前面，插入一个或多个同级节点。两者拥有相同的父节点。

注意，该方法不仅可以插入元素节点，还可以插入文本节点。

```javascript
var p = document.createElement('p');
var p1 = document.createElement('p');

// 插入元素节点
el.before(p);

// 插入文本节点
el.before('Hello');

// 插入多个元素节点
el.before(p, p1);

// 插入元素节点和文本节点
el.before(p, 'Hello');
```

`after`方法用于在当前节点的后面，插入一个或多个同级节点，两者拥有相同的父节点。用法与`before`方法完全相同。

ChildNode.replaceWith()

`replaceWith`方法使用参数节点，替换当前节点。参数可以是元素节点，也可以是文本节点。

```javascript
var span = document.createElement('span');
el.replaceWith(span);
```

上面代码中，`el`节点将被`span`节点替换。