ParentNode 接口，ChildNode 接口

目录

- ParentNode 接口
  - ParentNode.children
  - ParentNode.firstElementChild
  - ParentNode.lastElementChild
  - ParentNode.childElementCount
  - ParentNode.append()，ParentNode.prepend()
- ChildNode 接口
  - ChildNode.remove()
  - ChildNode.before()，ChildNode.after()
  - ChildNode.replaceWith()

节点对象除了继承 Node 接口以外，还拥有其他接口。`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，就会混入了（mixin）`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拥有 `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。

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

[illegible] `children` [illegible]
`HTMLCollection` [illegible] `length` [illegible] `0` [illegible]

[illegible] `HTMLCollection` [illegible] DOM [illegible]

ParentNode.firstElementChild

`firstElementChild` [illegible] `null` [illegible]

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

[illegible] `document` [illegible] `<HTML>` [illegible]

ParentNode.lastElementChild

`lastElementChild` [illegible] `null` [illegible]

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

[illegible] `document` [illegible] `<HTML>` [illegible] `document` [illegible]

ParentNode.childElementCount

`childElementCount`
[illegible] `0` [illegible]

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

ParentNode.append()[illegible] ParentNode.prepend()

[illegible] 1[illegible] ParentNode.append()

`append()` [illegible]

[illegible]

```
var parent = document.body;

// [illegible]
var p = document.createElement('p');
parent.append(p);

// [illegible]
parent.append('Hello');

// [illegible]
var p1 = document.createElement('p');
var p2 = document.createElement('p');
parent.append(p1, p2);

// [illegible]
var p = document.createElement('p');
parent.append('Hello', p);
```

[illegible]

[illegible] `Node.prototype.appendChild()` [illegible]

- `append()` [illegible] `appendChild()` [illegible]
- `append()` [illegible] `appendChild()` [illegible]
- `append()` [illegible] `appendChild()` [illegible]

[illegible] 2[illegible] ParentNode.prepend()

`prepend()` [illegible] `append()` [illegible]

ChildNode [illegible]

[illegible] `ChildNode` [illegible]

ChildNode.remove()

`remove()`方法用于从父节点移除当前节点。

```
el.remove()
```

上面代码在 DOM 里面移除了 `el`节点。

ChildNode.before()，ChildNode.after()

（1）ChildNode.before()

`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。

```
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');
```

（2）ChildNode.after()

`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。

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

上面代码中，`el`节点将被`span`节点替换。

版本号 #2

由 壳壳虫 创建于 2018-03-02 15:25:57 UTC

由 壳壳虫 更新于 2020-12-02 04:29:00 UTC