

element.outerHTML



概述

`element` DOM接口的 `outerHTML`属性, 获取描述包括其后代的元素的序列化HTML片段。它可以用从给定字符串解析的节点设置为替换元素。

语法



```
1 | let content = element.outerHTML;
```

返回时，内容包含描述元素及其后代的序列化HTML片段。

```
1 | element.outerHTML = content;
```

将元素替换为通过使用元素的父作为片段解析算法的上下文节点解析字符串内容生成的节点。

例子

获取一个元素的`outerHTML`属性的值：

```
1 | // HTML:
2 | /*
3 | <div id="d">
```

```

4      <p>Content</p>
5      <p>Further Elaborated</p>
6  </div>
7  */
8
9  const d = document.getElementById("d");
10 console.log(d.outerHTML);
11
12 /*
13  字符串 '<div id="d"><p>Content</p><p>Further Elaborated</p></div>'
14  被显示到控制台窗口
15  */

```

通过设置outerHTML属性来替换节点：

```

1  // HTML:
2  /*
3  <div id="container">
4      <div id="d">This is a div.</div>
5  </div>
6  */
7
8  let container = document.getElementById("container");
9  let d = document.getElementById("d");
10
11 console.log(container.firstChild.nodeName);
12 // logs "div"
13
14 d.outerHTML = "<p>This paragraph replaced the original div.</p>";
15
16 console.log(container.firstChild.nodeName);
17 // logs "P"

```

```
18
19  /*
20     #d div不再是文档树的一部分，新段替换了它。
21     (不在页面中显示,但仍然在内存中)
22  */
```

注意事项

如果元素没有父元素，即如果它是文档的根元素，则设置其outerHTML属性将抛出一个带有错误代码 *NO_MODIFICATION_ALLOWED_ERR* 的 [DOMException](#)。例如：

```
1 document.documentElement.outerHTML = "test";
2 // 抛出一个 DOMException
```

此外，虽然元素将在文档中被替换，设置了outerHTML属性的变量仍将保持对原始元素的引用：

```
1 let p = document.getElementsByTagName("p")[0];
2 console.log(p.nodeName);
3 // 显示: "P"
4 p.outerHTML = "<div>This div replaced a paragraph.</div>";
5
6 console.log(p.nodeName);
7 // 仍然为: "P";
```

规范

Specification	Status	Comment
---------------	--------	---------

浏览器兼容性

Desktop

Mobile

Feature	Firefox (Gecko)	Chrome	Internet Explorer	Opera	Safari
Basic support	11 (11)	0.2	4.0	7	1.3

相关链接

- MSDN: [outerHTML](#) 属性