

# 【前端面试】childNodes 和 children 的区...

---

## 前言

childNodes 和 children 算是一个细节问题，在当下框架火热的环境中，几乎很少有人会去直接操作 DOM 节点了。但是如果你比较深入的去学习框架之后，会发现它们的底层原理还是逃脱不了 DOM 的操作，所以说，学习这些关于 DOM 的细节问题还是很重要的，面试官也常常会以这些细节来判断一个人的知识是否掌握到位！

## 1.Node 和 Element

想要学习这两个属性，我们必须要先知道 Node 和 Element 的区别。

在 JS 操作 DOM 的时候，我们有时候会操作 Node 节点，有时候也会才做 Element，这两者的区别有些人傻傻分不清，其实它们两个的区别很简单。

- **Node:**

在 DOM 树中，所有的节点都是 Node，包括 Element，也就是说 Node 包含了 HTML 元素标签、text、以及注释等等内容，它是所有 DOM 的基类。

- **Element:**

在 DOM 树中，Element 是所有 HTML 元素的基类，也就是说，Element 只包含 HTML 元素标签。

综上两点可以得出：Node 和 Element 两者是包含关系，Node 包含 Element。从而衍生出了两个集合：NodeList 和 HTMLCollection。NodeList 是 Node 的集合，HTMLCollection 是 Element 的集合。

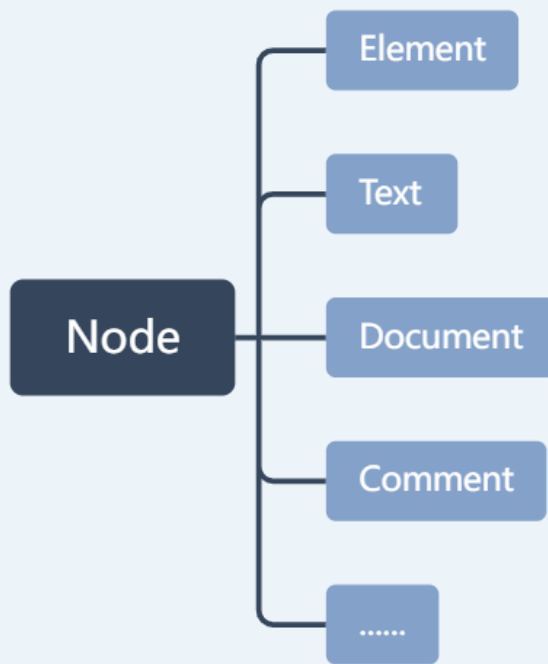
## 2.childNodes理解

先来看下官网的解释：

Node.childNodes 返回包含指定节点的子节点的集合，该集合为即时更新的集合。

官网说的也比较通俗易懂，childNodes 属性返回的是 DOM 节点的子集合，它属于 NodeList 集合。

如下图所示：



总结：

childNodes会返回所有节点，包括HTML、Text、注释等等内容。

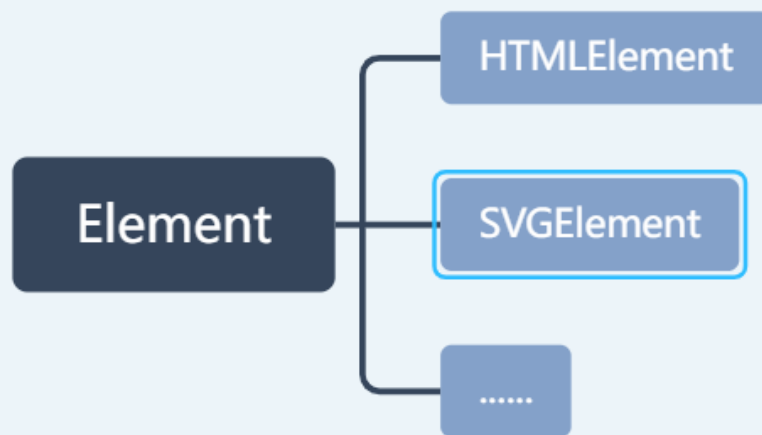
### 3.children理解

同样先来看下官网解释：

Element.children 是一个只读属性，返回 一个Node的子elements ， 是一个动态更新的 HTMLCollection 。

通过官网的解释可以得出：children属于HTMLCollection集合，它返回的是节点的子elements，通俗的说，children返回的是HTML元素标签。

如下图所示：



总结：

children只返回元素节点。

## 4.代码演示

### 4.1 childNodes

我们任意编写一些HTML代码，然后通过JS获取。

代码如下：

```
1 <body>
2   <div id="box1">
3     <p>小猪课堂</p>
4     <span>小猪课堂</span>
5     <b>小猪课堂</b>
6     <em>小猪课堂</em>
7     <!-- 我是注释 -->
8   </div>
9 </body>
10 <script>
11   const box1 = document.getElementById("box1");
12   console.info("-----childNodes-----");
13   console.info(box1.childNodes);
14 </script>
```

输出结果：

```
NodeList(11) [text, p, text, span, text, b, text, em, text, comment,
text] ⓘ
  ▶ 0: text
  ▶ 1: p
  ▶ 2: text
  ▶ 3: span
  ▶ 4: text
  ▶ 5: b
  ▶ 6: text
  ▶ 7: em
  ▶ 8: text
  ▶ 9: comment
  ▶ 10: text
      length: 11
  ▶ [[Prototype]]: NodeList
```

我们会发现输出的节点长度为11，其中多了很多的文本节点，和我们编写的代码产生了冲突，问题就在于childNodes会返回所有节点，包括注释、标签、换行文本等等。

我们把HTML放到一行在执行一遍，代码如下：

```
1 <body>
2   <div id="box1"><p>小猪课堂</p><span>小猪课堂</span><b>小猪课堂</b><em>小猪课堂
3 </body>
```

输出结果：

-----childNodes-----

```
▼ NodeList(5) [p, span, b, em, comment] ⓘ
  ▶ 0: p
  ▶ 1: span
  ▶ 2: b
  ▶ 3: em
  ▶ 4: comment
      length: 5
  ▶ [[Prototype]]: NodeList
```

这个时候输出结果就和我们的理想结果一样了，输出中包含了节点和注释。

通过上面的代码实例可以看出，如果我们使用childNodes来进行获取子节点，那么换行也会被统计进去，这不是我们想要的结果。为了解决这个问题，我们可以通过nodeType这个属性来判断该节点是什么类型，它的取值有3个：

- 1：元素节点
- 2：属性节点
- 3：文本节点

代码如下：

```
1 <body>
2   <div id="box1">
```

```

3     <p>小猪课堂</p>
4     <span>小猪课堂</span>
5     <b>小猪课堂</b>
6     <em>小猪课堂</em>
7     <!-- 我是注释 -->
8 </div>
9 </body>
10 <script>
11     const box1 = document.getElementById("box1");
12     console.info("-----childNodes-----");
13     console.info(box1.childNodes);
14     let NodeList = box1.childNodes
15     for (let i = 0; i < NodeList.length; i++) {
16         if (NodeList[i].nodeType === 1) {
17             console.info("元素节点", NodeList[i])
18         } else if (NodeList[i].nodeType === 2) {
19             console.info("属性节点", NodeList[i])
20         } else {
21             console.info("文本节点", NodeList[i])
22         }
23     }
24 </script>

```

输出结果:

NodeList(11) [text, p, text, span, text, b, text, em, text, comment, text]		
文本节点 ▶ #text		<a href="#">index.html:68</a>
元素节点 <p>小猪课堂</p>		<a href="#">index.html:64</a>
文本节点 ▶ #text		<a href="#">index.html:68</a>
元素节点 <span>小猪课堂</span>		<a href="#">index.html:64</a>
文本节点 ▶ #text		<a href="#">index.html:68</a>
元素节点 <b>小猪课堂</b>		<a href="#">index.html:64</a>
文本节点 ▶ #text		<a href="#">index.html:68</a>
元素节点 <em>小猪课堂</em>		<a href="#">index.html:64</a>
文本节点 ▶ #text		<a href="#">index.html:68</a>
文本节点 <!-- 我是注释 -->		<a href="#">index.html:68</a>
文本节点 ▶ #text		<a href="#">index.html:68</a>

## 4.2 children

我们通过children的方式来看看能获取到哪些元素。

代码如下:

```

1 <body>
2   <div id="box1">
3     <p>小猪课堂</p>
4     <span>小猪课堂</span>
5     <b>小猪课堂</b>
6     <em>小猪课堂</em>

```

```
7      <!-- 我是注释 -->
8      </div>
9  </body>
10 <script>
11     const box1 = document.getElementById("box1");
12     console.info("-----children-----");
13     console.info(box1.children);
14 </script>
```

输出结果:

-----children-----

---

▼ *HTMLCollection*(4) [*p*, *span*, *b*, *em*] ⓘ

- ▶ 0: *p*
- ▶ 1: *span*
- ▶ 2: *b*
- ▶ 3: *em*

length: 4

▶ [[Prototype]]: *HTMLCollection*

此时输出的就只有HTML元素节点了，长度也正是我们的标签元素的个数。

## 总结

两个属性的区别到这里应该就很清楚了，它们有着各自的有点和缺点，需要根据不同的场景下使用不同的属性，总结如下两点：

- `childNodes`属于`NodeList`集合，它会返回所有的子节点，包括文本、标签、注释等等。
- `children`数据`HTMLCollection`集合，它会返回所有HTML元素节点。
- `childNodes`包含`children`。

如果觉得文章不够详细，可以观看B站视频：[小猪课堂](#)