Node: firstChild property

- Value
- Example
- Specifications
- Browser compatibility
- See also

The read-only **`firstChild`** property of the Node interface returns the node's first child in the tree, or `null` if the node has no children.

If the node is a Document, this property returns the first node in the list of its direct children.

> **NOTE:** This property returns any type of node that is the first child of this one. It may be a Text or a Comment node. If you want to get the first Element that is a child of another element, consider using Element.firstElementChild.

Value

A Node, or `null` if there are none.

Example

This example demonstrates the use of `firstChild` and how whitespace nodes might interfere with using this property.

```
<p id="para-01">
  <span>First span</span>
</p>
```

```
const p01 = document.getElementById("para-01");
console.log(p01.firstChild.nodeName);
```

In the above, the console will show '#text' because a text node is inserted to maintain the whitespace between the end of the opening `<p>` and `<span>` tags. **Any** whitespace will create a `#text` node, from a single space to multiple spaces, returns, tabs, and so on.

Another `#text` node is inserted between the closing `</span>` and `</p>` tags.

If this whitespace is removed from the source, the #text nodes are not inserted and the span element becomes the paragraph's first child.

```
<p id="para-01"><span>First span</span></p>
```

```
const p01 = document.getElementById("para-01");
console.log(p01.firstChild.nodeName);
```

Now the console will show 'SPAN'.

To avoid the issue with `node.firstChild` returning `#text` or `#comment` nodes, Element.firstElementChild can be used to return only the first element node.

Specifications

Browser compatibility

See also

- Element.firstElementChild
- Node.lastChild

□□□ #3
□ □□□ □□□ 2026-10-31 03:27:53 UTC
□ cocobook-sync □□□ 2026-11-04 17:34:47 UTC