Node: textContent property

- Value
  - Differences from innerText
  - Differences from innerHTML
- Examples
- Specifications
- Browser compatibility
- See also

The **`textContent`** property of the Node interface represents the text content of the node and its descendants.

> **NOTE:** `textContent` and HTMLElement.innerText are easily confused, but the two properties are different in important ways.

Value

A string, or `null`. Its value depends on the situation:

- If the node is a document or a doctype, `textContent` returns `null`.

  > **NOTE:** To get *all* of the text and CDATA data for the whole document, use `document.documentElement.textContent`.

- If the node is a CDATA section, a comment, a processing instruction, or a text node, `textContent` returns, or sets, the text inside the node, i.e., the Node.nodeValue.
- For other node types, `textContent` returns the concatenation of the `textContent` of every child node, excluding comments and processing instructions. (This is an empty string if the node has no children.)

> **WARNING:** Setting `textContent` on a node removes *all* of the node's children and replaces them with a single text node with the given string value.

Differences from innerText

Don't get confused by the differences between `Node.textContent` and HTMLElement.innerText. Although the names seem similar, there are important differences:

- `textContent` gets the content of *all* elements, including script and style elements. In contrast, `innerText` only shows "human-readable" elements.
- `textContent` returns every element in the node. In contrast, `innerText` is aware of styling and won't return the text of "hidden" elements.
  - Moreover, since `innerText` takes CSS styles into account, reading the value of `innerText` triggers a reflow to ensure up-to-date computed styles. (Reflows can be computationally expensive, and thus should be avoided when possible.)

Differences from innerHTML

Element.innerHTML gets or sets HTML, as its name indicates. We advise against using `innerHTML` to get or set text inside an element because it deals with raw HTML rather than plain text and can be susceptible to XSS attacks. Even if you are sure that the text never contains HTML syntax, it is still less semantic and slower because it needs to invoke the HTML parser.

Examples

Start with this HTML fragment.

```
<div id="divA">This is <span>some</span> text!</div>
```

You can use `textContent` to get the element's text content:

```
let text = document.getElementById("divA").textContent;
// The text variable is now: 'This is some text!'
```

If you prefer to set the element's text content, you can do:

```
document.getElementById("divA").textContent = "This text is different!";
// The HTML for divA is now:
// <div id="divA">This text is different!</div>
```

Specifications

Browser compatibility

See also

- HTMLScriptElement.textContent and HTMLScriptElement.text
- HTMLElement.innerText
- Element.innerHTML

□□□ #3
□ □□□ □□□ 2026-10-31 03:27:53 UTC
□ cocobook-sync □□□ 2026-11-04 17:34:51 UTC