



JavaScript textContent

Summary: in this tutorial, you will learn how to use the JavaScript `textContent` property to get the text content of a node and its descendants.

Reading textContent from a node

To get the text content of a node and its descendants, you use the `textContent` property:

```
let text = node.textContent;
```

Suppose that you have the following HTML snippet:

```
<div id="note">
  JavaScript textContent Demo!
  <span style="display:none">Hidden Text!</span>
  <!-- my comment -->
</div>
```

The following example uses the `textContent` property to get the text of the `<div>` element:

```
let note = document.getElementById('note');
console.log(note.textContent);
```

How it works.

- First, select the `div` element with the id `note` by using the `getElementById()` method.
- Then, display the text of the node by accessing the `textContent` property.

Output:

```
JavaScript textContent Demo!  
Hidden Text!
```

As you can see clearly from the output, the `textContent` property returns the concatenation of the `textContent` of every `child node`, excluding comments (and also processing instructions).

textContent vs. innerText

On the other hand, the `innerText` takes the CSS style into account and returns only human-readable text. For example:

```
let note = document.getElementById('note');  
console.log(note.innerText);
```

Output:

```
JavaScript textContent Demo!
```

As you can see, the hidden text and comments are not returned.

Since the `innerText` property uses the up-to-date CSS to compute the text, accessing it will trigger a reflow, which is computationally expensive.

A **reflow** occurs when a web browser needs to process and draw parts or all of a webpage again.

Setting textContent for a node

Besides reading `textContent`, you can also use the `textContent` property to set the text for a node:

```
node.textContent = newText;
```

When you set `textContent` on a node, all the node's children will be removed and replaced by a single text node with the `newText` value. For example:

```
let note = document.getElementById('note');
note.textContent = 'This is a note';
```

Summary

- Use the `textContent` property to return the concatenation of the `textContent` of every child node. You can use it to set a text for a node.
- The `innerText` returns the human-readable text that takes CSS into account.

Quiz