

Day 16



Element only Navigation:

What is Element-only Navigation?

In the DOM (Document Object Model), every part of an HTML page — including tags, text, and comments — is treated as a node. But sometimes, we only want to navigate between element nodes (like `<div>`, `<p>`, `<h1>`) and ignore text nodes (like spaces or line breaks).

This is called Element-only Navigation.

Example HTML

```
<div id="container">
  <h2>Heading</h2>
  <p>Paragraph</p>
</div>
```

Here, inside `<div>` there are:

- Two element nodes → `<h2>`, `<p>`
- Plus some text nodes (spaces and line breaks)

If you use general node navigation (`firstChild`, `nextSibling`), those text nodes will also be counted — which can cause confusion.

Why Use Element-only Navigation?

To skip text and comment nodes and only deal with real HTML elements. That's why we use the Element-only properties.

Element-only Navigation Properties

| Property | Description | Example |
|-------------------------------------|---------------------------------|---------------------------------------|
| <code>firstElementChild</code> | First child element | <code>div.firstElementChild</code> |
| <code>lastElementChild</code> | Last child element | <code>div.lastElementChild</code> |
| <code>nextElementSibling</code> | Next element in same parent | <code>h2.nextElementSibling</code> |
| <code>previousElementSibling</code> | Previous element in same parent | <code>p.previousElementSibling</code> |

| Property | Description | Example |
|----------|--------------------|--------------|
| children | All child elements | div.children |

Difference Between “Node” and “Element”

| Type | Includes | Example Property |
|-------------------------|----------------------------|---------------------------------------|
| Node Navigation | Elements + Text + Comments | firstChild, nextSibling |
| Element-only Navigation | Only HTML elements | firstElementChild, nextElementSibling |

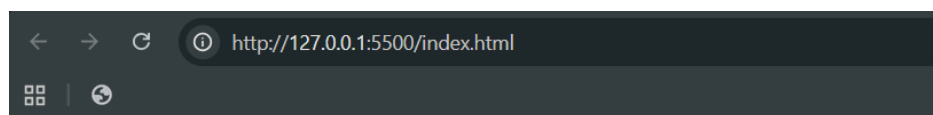
To understand this, we shall start from the basics:

```

Welcome  index.html X
index.html > ...
1  <!DOCTYPE html>
2  <html>
3  <head>
4  |   <title>Element-only Navigation Example</title>
5  </head>
6  <body>
7  |   <div id="container">
8  |       <h2>Heading</h2>
9  |       <p>Paragraph</p>
10 |   </div>
11 </body>
12 </html>

```

Output:



Heading

Paragraph

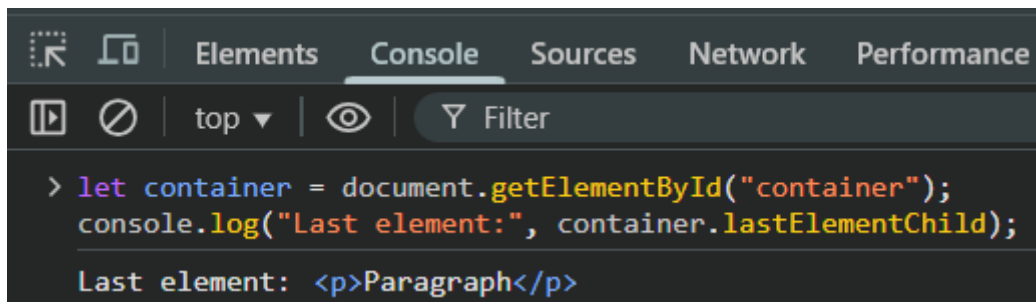
Console:

```

Elements  Console  Sources  Network  Performance  Memory  App
top  Filter
> let container = document.getElementById("container");
  console.log("First element:", container.firstElementChild);
First element: <h2>Heading</h2>

```

Also,



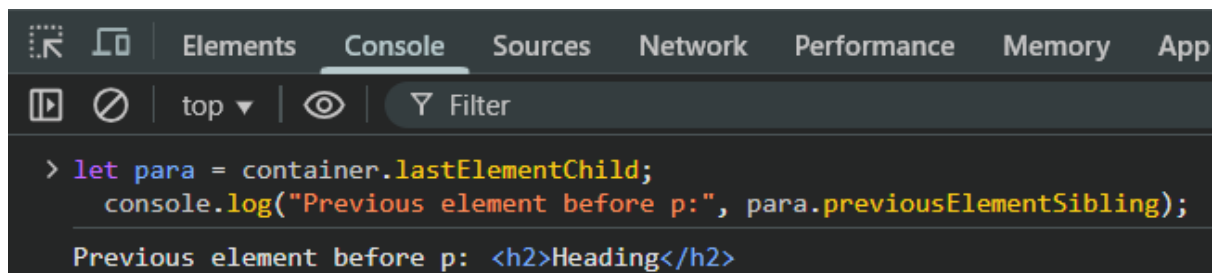
The screenshot shows the Chrome DevTools Console with the 'Console' tab selected. The code executed is: `> let container = document.getElementById("container"); console.log("Last element:", container.lastElementChild);`. The output is: `Last element: <p>Paragraph</p>`.

Also,



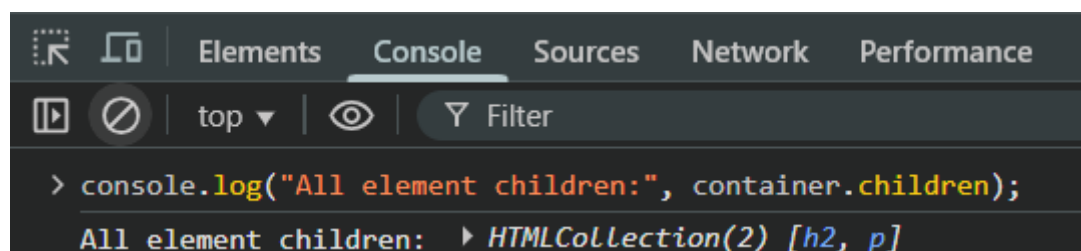
The screenshot shows the Chrome DevTools Console with the 'Console' tab selected. The code executed is: `> let heading = container.firstChild; console.log("Next element after h2:", heading.nextElementSibling);`. The output is: `Next element after h2: <p>Paragraph</p>`.

Also,



The screenshot shows the Chrome DevTools Console with the 'Console' tab selected. The code executed is: `> let para = container.lastElementChild; console.log("Previous element before p:", para.previousElementSibling);`. The output is: `Previous element before p: <h2>Heading</h2>`.

Also,



The screenshot shows the Chrome DevTools Console with the 'Console' tab selected. The code executed is: `> console.log("All element children:", container.children);`. The output is: `All element children: ▶ HTMLCollection(2) [h2, p]`.

--The End--