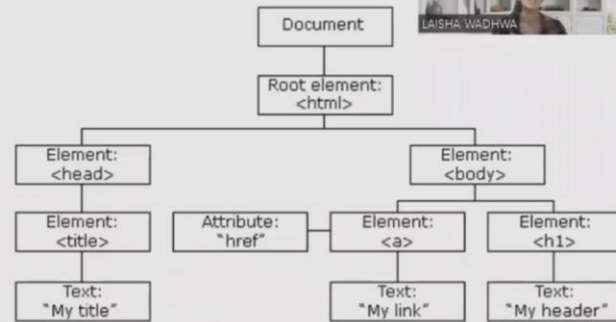


DOM Nodes

- The entire document is a document node
- Every HTML element is an element node
- The text inside HTML elements are text nodes
- Every HTML attribute is an attribute node (deprecated)
- All comments are comment nodes



```

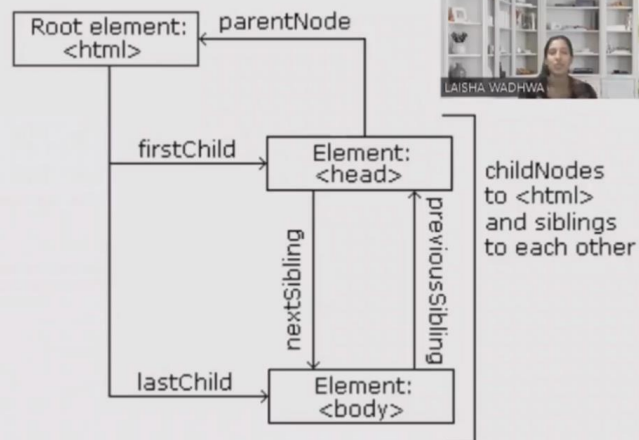
<html>

<head>
  <title>DOM Tutorial</title>
</head>

<body>
  <h1>DOM Lesson one</h1>
  <p>Hello world!</p>
</body>

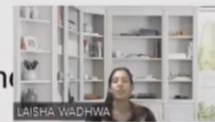
</html>

```



The node method

You can access any element by using the following properties with the object:



- `node.childNodes` – accesses the child nodes of a selected parent
- `node.firstChild` – accesses the first child of a selected parent
- `node.lastChild` – accesses the last child of a selected parent.
- `node.parentNode` – accesses the parent of a selected child node.
- `node.nextSibling` – accesses the next consecutive element (sibling) of a selected element.
- `node.previousSibling` – accesses the previous element (sibling) of a selected element

Ways to access inner text /text node

```
<title id="demo">DOM Tutorial</title>
```

```
myTitle = document.getElementById("demo").innerHTML;
```

```
myTitle = document.getElementById("demo").firstChild.nodeValue;
```

```
myTitle = document.getElementById("demo").childNodes[0].nodeValue;
```

```

<!DOCTYPE html>
<html>
<body>

<!-- This is a comment node! -->

<h1>The Element Object</h1>
<h2>The childNodes Property</h2>
<p>Whitespace between elements are text nodes. Comments are also nodes.</p>

<p>The body element's child nodes are:</p>
<p id="demo"></p>

<script>
const nodeList = document.body.childNodes;

let text = "";
for (let i = 0; i < nodeList.length; i++) {
  text += nodeList[i].nodeName + "<br>";
}
document.getElementById("demo").innerHTML = text;
</script>

</body>
</html>

```

The Element Object

The childNodes Property

Whitespace between elements are text nodes. Comments are also nodes.

The body element's child nodes are:

```

#text
#comment
#text
H1
#text
H2
#text
p
#text
p
#text
p
#text
SCRIPT

```

So we seen these nodes when we acces usng childnodes !

Types of Nodes



Node	Type	Example
ELEMENT_NODE	1	<h1 class="heading">W3Schools</h1>
ATTRIBUTE_NODE	2	class = "heading" (deprecated)
TEXT_NODE	3	W3Schools
COMMENT_NODE	8	<!-- This is a comment -->
DOCUMENT_NODE	9	The HTML document itself (the parent of <html>)
DOCUMENT_TYPE_NODE	10	<!Doctype html>

While when using children to access childs

When using the `children` property in JavaScript, text nodes are not included in the collection. The `children` property specifically returns an **HTMLCollection** of the element's child **element nodes** (i.e., elements like `<div>`, `<p>`, ``, etc.), excluding text nodes and comment nodes.

```
test.html > html > body
2  <html>
3  <head>
4
5  </head>
6  <body>
7      <h2>Favorite Beverages</h2>
8      <ul id="beverageList">
9          <li>Coffee</li>
10         <li>Tea</li>
11         <li>Milk</li>
12     </ul>
13
14     <button id="changeSecondItem">Change Second Item to Juice</button>
15     You, 10 hours ago • first commit
16     <script>
17         // Get the unordered list element
18         const list = document.getElementById("beverageList");
19
20         // Access all child <li> elements
21         const items = list.children;
22
23         // Log the text content of each child element
24         for (let i = 0; i < items.length; i++) {
25             console.log("Item " + (i + 1) + ": " + items[i].textContent);
26         }
27
28         // Change the text of the second item using the 'children' property
29         const changeSecondItemButton = document.getElementById("changeSecondItem");
30         changeSecondItemButton.addEventListener("click", function() {
31             // Access the second child (index 1, since children is zero-indexed)
32             if (items.length > 1) { // Check if there is a second item
33                 items[1].textContent = "Juice";
34             }
35         });
36     </script>
37 </body>
38 </html>
39
```

Favorite Beverages

- Coffee
- Tea
- Milk

Change Second Item to Juice

⌵

⌵

Elements

Console

Sources

Network

Performance

Memory

Application

Security

Lighthouse

Recorder

⌵

⌵

top ▾

⌵

Filter

▶

⌵

4 messages

▶

⌵

4 user mes...

⊗

No errors

⚠

No warnings

▶

⌵

4 info

⌵

No verbose

Item 1: Coffee

Item 2: Tea

Item 3: Milk

uBOL: Generic cosmetic filtering stopped because no more DOM changes

Use of nodesiblings

```
1 <!DOCTYPE html>
2 <html>
3 <body>
4 <div class="list">
5   <h2 class="title">Items</h2>
6   <ul class="items">
7     <li class="item">Item 1</li>
8     <li class="item">Item 2</li>
9     <li class="item">Item 3</li>
10  </ul>
11  <p>Hello world</p>
12 </div>
13 <script type="text/javascript">
14   const ul = document.querySelector(".items");
15   // parentNode
16   console.log(ul.parentNode)
17   ul.parentNode.style.color = "red";
18
19   //childNodes
20   // console.log(ul.children[1])
21   // console.log(ul.previousElementSibling)
22 </script>
23
24 </body>
25 </html>
```


UI children access and ul sibling

```
<!DOCTYPE html>
<html>
<body>
<div class="list">
  <h2 class="title">Items</h2>
  <ul class="items">
    <li class="item">Item 1</li>
    <li class="item">Item 2</li>
    <li class="item">Item 3</li>
  </ul>
  <p>Hello world</p>
</div>
<script type="text/javascript">
  const ul = document.querySelector(".items");
  // parentNode
  console.log(ul.parentNode)
  ul.parentNode.style.color = "red";

  //childnode
  console.log(ul.children[1])
  console.log(ul.previousElementSibling)
</script>

</body>
</html>
```

Items

- Item 1
- Item 2
- Item 3

Hello world

