Walking the DOM:

DOM tree refers to the HTML page where all the nodes are objects. There can be 3 main types of nodes in the DOM tree:

1. text nodes
2. element nodes
3. comment nodes

In an HTML page, < html > is at the root, and < head > and < body > are its children, etc.

A text node is always the leaf of the tree.

```
1. document.body

On writing this in the console, it will show the body of the html file.
```

```
2. document.body.childNodes

This will show all the child nodes in the body.

Ex:

#text, div.box, #text, div.box, #text, div.box, #text, div.box, #text, div.box

This tells that first there is a text i.e, newline, then there is a div. After closing of the
div there is a text,i.e, newline, then there is a div and so on...
```

```
3. document.body.childNodes[1]

This will show the child node at index 1, i.e, at 2nd position.
```

```
4. document.body.childNodes[1].childNodes

This will show the child nodes of the child node at index 1.
```

Note: A new line also considered a text node.

---

firstChild, lastChild, childNodes:

1. element.firstChild -> First child element.
2. element.lastChild -> Last child element.
3. element.childNodes -> All child nodes.

We can set "document.body.childNodes[1]" to a variable named cont, like this:

let cont = document.body.childNodes[1]

After this whenever we want to fetch document.body.childNodes[1], we can write cont and it will refer to the child node (present in the body) at index 1.

Remember that after reloading webpage, cont will remian undefined.

Now we can refer as cont.firstChild and so on.

On refering like this, getting text as output is very frustrating so we use firstElementChild.

Remember that text node is not an element.

Now on writing: cont.lastElementChild , we will get the elements say a div.

Suppose we have a div named container, inside it are 5 more divs. On writing the below code in console:

```
document.body.firstElementChild.children

we get -> 0: div.box, 1: div.box, 2: div.box, 3: div.box, 4: div.box, length: 5
```

Now on writing:

```
document.body.firstElementChild.children[4] , we get the child element at index[4].
```

Now write:

```
document.body.firstElementChild.children[3].nextElementSibling

we will get div at index 4.
```

Now write:

```
document.body.firstElementChild.children[3].previousElementSibling

we will get the div at index 2.
```

Parent:

```
document.body.firstElementChild.children[3].previousElementSibling.parentElement

Output will be div named container.
```