- But if we use single element selector then we can't use li[0], li[1]

b[0] single element selectors selects only one element at a time.

DOM - Part-2

* parentNode → The parentNode is the element or Node in which the element present themselves.

~~parentNode = document.~~

item = document.querySelector('<ClassName>
<.idName>
<Element>')

item.parentNode ;
↳ points to parentNode.

→ We can change property or HTML of that Element.

Eg → item.parentNode.style.color = "red";

This will change the color of parent Node.

→ We can do chaining of this.

→ item.parentNode.parentNode.parentNode. ——— ——

- from the last → it points.

→ [NULL ← HTML tag ← body. ← parentNode]

* childNodes

qs is used to select child node but it count whitespaces also which makes it no use in general

Eg: item.childNodes

→ if many element with same class name & item then we can do like.

item.childNodes[0]
1
2
}

vs children

qs is used to select children element also but it counts HTML element no whitespaces which makes it efficient to use

Eg → item.children

same here..

item.children[0]
[1]
}

* firstChild vs firstElementChild.

Again this is same as childNode only it selects the first child [insignificant]
→ [No use]

This selects the 1st Element child
→ useful

→ item.firstChild

Also, if many with same [tabName] and we are selecting with multiple selector then we can access like an array.

→ item.firstElementChild

Same here

Eg → item.firstChild[0]
[1]
[2]
.
.
.

Eg → item.firstElementChild[1]
[1]
.
.
.

Also we can change property or add/edit/delete dom element or its attribute.

Eg → item.firstElementChild[0].style.color = 'red';

→ Similarly we have,

→ lastChild vs lastElementChild.

→ nextSibling vs nextElementSibling.

→ previousSibling vs previousElementSibling

→ Create Element

// create a div.

const newdiv = document.createElement('div')

// add a class

→ newdiv.className = "new-div" ;
(className)

// add id

→ newdiv.idName = "new-id" ;
(idName)

~~add attribute~~

// Set attribute.

→ newdiv.setAttribute('title', 'hello div');

// create a textNode

→ divTextNode = document.createTextNode('hello div
this.id.me');

// append ~~child~~ ns. textNode inside newdiv

→ newdiv.appendChild(divTextNode);

// Inserting to specific location

var container = document.querySelector('header .container')
var h1 = document.querySelector('header h1')

// insert before h1.

container.insertBefore(newdiv, h1)

newly created Element

Element which before newdiv will be inserted