**DOM (Document Object Model)**

What is DOM?

* Document Object Model is a hierarchical tree like structure of our webpage, where every node(element) is an object. Like document itself is an object, elements are objects, text inside elements are object.

BOM (Browser Object Model)

(complete the diagram)

Why it is needed?

* DOM connects our webpage and programming language, so if we want to modify our webpage dynamically or in real-time, it can only happen with the help of programming language and our programming language manipulate or target the DOM objects only. Because modifying DOM means modifying webpage.(as DOM is nothing but structure of our webpage).
* We can (create, access, update, delete)DOM objects through java script.



In this example on clicking button the text inside h1 will change. So here we are

manipulating DOM.

**DOM Methods**

To select a particular element

* document.getElementsById(“id”) - return the single element(datatype-Object)
* document.getElementsByClassName(“className”) – return the collection of elements(datatype -HTML collection of Objects)
* document.getElementsByTagName(“tagname”)- return collection of elements(datatype- HTML collection of Objects)
* document.querySelector(“id,class,tag”)- it will return single element.(datatype-Object) Note- if there will multiple elements, first element will be returned.
* document.querySelectorAll(“id,class,tag”)- it will return collection of Objects(datatype- Nodelist)

Difference between HTML Collection and NodeList

* HTML collection do not support forEach,map
* NodeList support forEach but map is not supported

Array Methods

To check a particular variable is Array or not

* Array.isArray(variableName) 🡪 it will return Boolean value true(means array) or false(not array)

To convert a variable into Array

* Array.from(VariableName) 🡪 return a new Array

Difference between innerHTML and innerText and textContent

innerText will only print text inside that element, and if any tag is present inside that tag it will ignore the tag.

For ex-   <h3>Brigadier Saurabh singh <span style="display: none;">Shekhawat</span></h3>

*let* a =document.querySelector("h3")

console.log(a.innerText)

Output- Brigadier Saurabh singh

innerHTML will print the text + tag (if any tag is present inside the element).

console.log(a.innerHTML)

Output- Brigadier Saurabh singh <span style="display:none;">Shekhawat</span>

textContent will print the text(even if that text is hidden by using css properties).

console.log(a.textContent)

Output- Brigadier Saurabh singh Shekhawat

**Event Propagation**

When one element event gets triggered along with that it also triggered all its super element’s event is called event propagation.

Two types

Event bubbling - in which event trigger start from inside. (like bubble)

Event capturing – in which event trigger start from outside. (like capturing of fort happens in war, first outside and then inside)

addEventListener(“event”, “function”{

}, true/false)

Third parameter will decide Event bubbling will happen or Event capturing will happen. By default if you not mention third argument it will be false by default.

false – means event bubbling.

true – means event capturing.