Week 5 Javascript Continue & CSS Framework Bootstrap

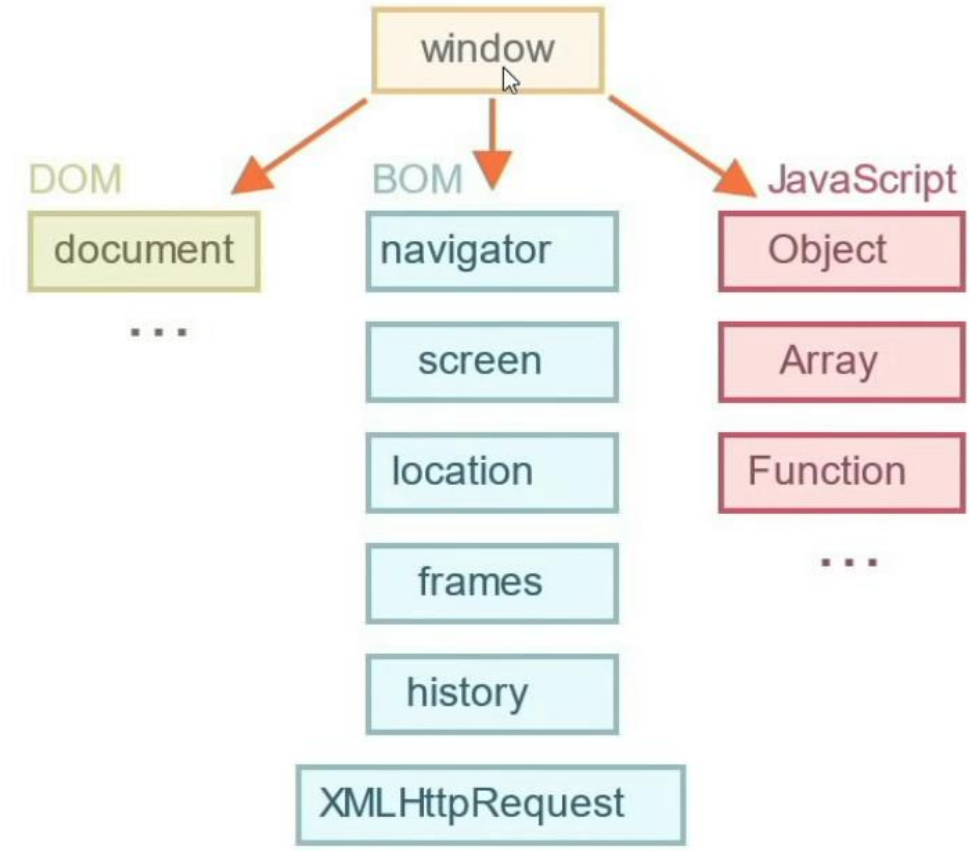
Topics we discussed:

- Window vs Document
- DOM vs BOM
- DOM Navigation
- Searching and getting Elements Reference

- **Window vs Document**

window	**document**
1. A window is the main container or the *global object*, and any operations related to the entire browser window can be a part of the window object. 2. All the members like objects, methods or properties. So if they are part of the window object, then we do not refer to the window object. 3. The window has methods, properties and objects. ex setTimeout()or setInterval()are the methods, whereas document is the Object of the Window, and It also has a screen Object with properties describing the physical display.	1. whereas the DOM is the child of a window object 2. Where in the DOM, we need to refer to the document, if we want to use the document object, methods or properties 3. Document is just the object of the global object, that is, the window that deals with the document and the HTML elements themselves.

Let's have an example: Goto any website and in console write window and all the related objects you can find. So if you write window, location, then you get all the included history, but if we have to do it through a document, we can't as a document can only access HTML elements to refer to.

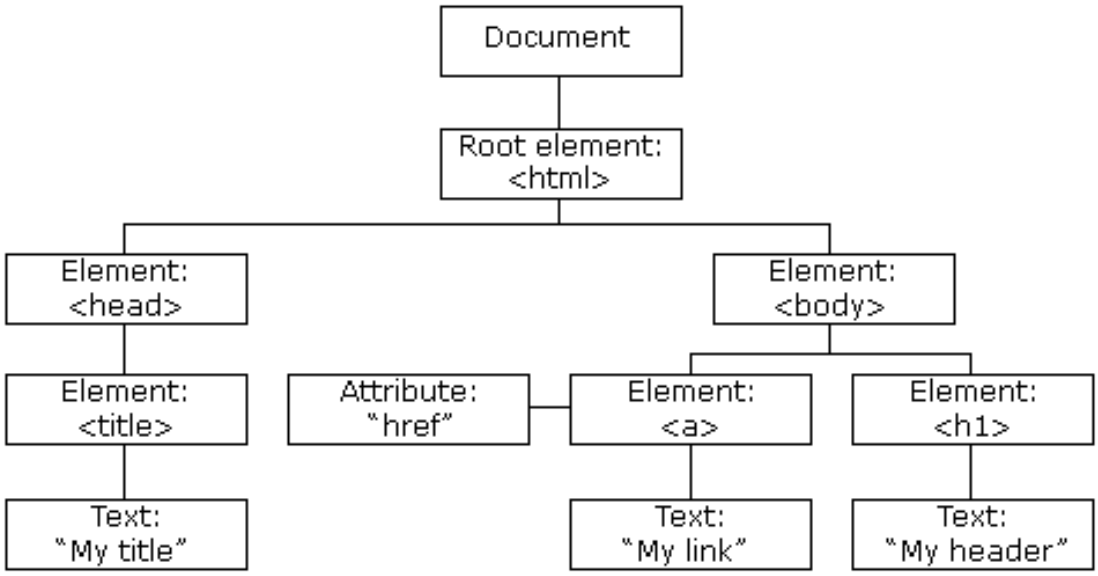


- **DOM vs BOM**

```
 👉 The DOM is the Document Object Model, which deals with the document,
 the HTML elements themselves, e.g. document and all traversal you
 would do in it, events, etc. DOM stands for Document Object Model: it is how
the the browser stores your HTML and CSS.

We can access this from JavaScript using the `document` global variable.
Javascript can use this to:

- read HTML

- manipulate HTML

 For Ex: 
 change the background color to red
 document.body.style.background = "red";
```

```
👉 The BOM is the Browser Object Model, which deals with browser components
aside from the document, like history, location, navigator and screen
(as well as some others that vary by browser). OR
In simple meaning, all the Window operations which come under BOM are
performed  using BOM


Let's see more practical on History object

Functions alert/confirm/prompt are also a part of BOM:
they are directly not related to the document,
but represent pure browser methods of communicating with the user.

alert(location.href); // shows current URL
if (confirm("Want to Visit FrontMentorChallenge?")) {
  location.href = "https://www.frontendmentor.io/"; // redirect the browser to
another URL
}
```

- **DOM Navigation**



```
1: document.documentElement
      returns the Element that is the root element of the document.
2: document.head
3: document.body
4: document.body.childNodes (include tab,enter and whiteSpace)
  list of the direct children only
5: document.children (without text nodes, only regular Elements)
6: document.childNodes.length
```

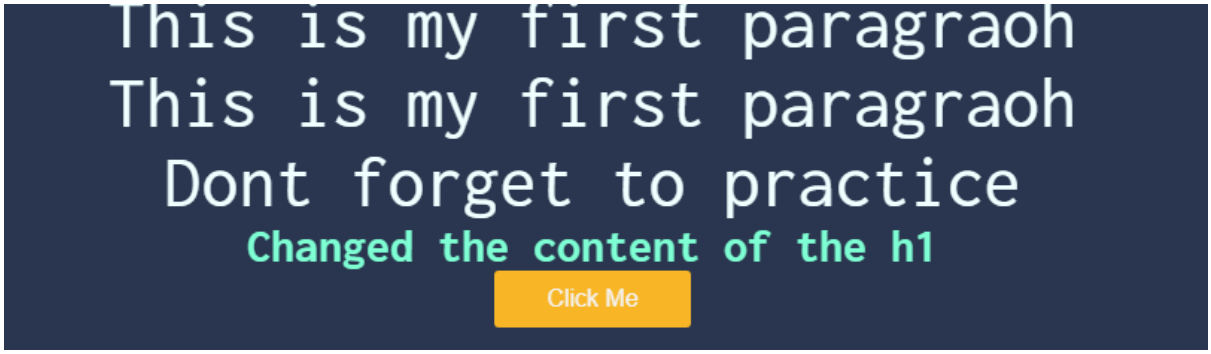
Let's do practical

Open DOMNavigation.html file and let's discover all the above Dom navigation structure.

```
👉 Practice Time
How to check whether an element has child nodes or not?
we will use hasChildNodes()

👉 Practice Time
How to find the child in DOM tree
firstChild vs firstElementChild
lastChild vs lastElementChild

firstChild vs firstElementChild
lastChild vs lastElementChild


Example
const firstChild = document.body.firstElementChild.firstElementChild;
firstChild
firstChild.style.color='red'; //CSS Document Style
```

- **Searching and getting Elements Reference**

Open search.html file and now we are going to change the heading by clicking on button :



HOW?

1.change content //search

```
const changeContent = () => {

         document.getElementById('heading').innerHTML ="Welcome to Web
development Evening Bootcamp"
        }
```

2.change content //by reference

```
 const headingChange=  document.getElementById('heading')
       headingChange.innerHTML="Welcome to Web development Evening Bootcamp"
```

3.Search element //class property

```
console.log (document.getElementsByClassName('para'));
```

4.Search element // by Tag nam

```
console.log(document.getElementsByTagName('p'));
```

```
Now the same we have is the query selectors
querySelector() returns the first matching value whereas
querySelectoral1()will return all the elements
```

`document.querySelector("selector goes here")`

The query selector takes a CSS selector. Anything you can use in CSS to target HTML you can use in this function/method - don't forget to use `.` for classes and `#` for ids!

Returns the FIRST (and only the first) item that matches the selector. This is called an Element.

5.querySelector

```
document.querySelector('#heading').innerHTML="I changed again";
```

6.querySelectorAll()

`document.querySelectorAll("selector")`

Returns a special kind of array called a NodeList, that contains every Element that matches the selector (for example, every p tag on a page).

```
 console.log(document.querySelectorAll('.para'));
```

Take Away Homework/ Project Light bulb ON or OFF: