**1st Video**

DOM – Document Object Model

It converts the whole html document into an object.

We can access the elements using *document*

We can also access the body of the document using *document.body*

DOM means the whole document is a tree and it have many sub sub parts.

*<!-- We can use script tag in two way -->*

*<!-- Inside the HTML -->*

    <script>

        console.log('I am JS')

    </script>

*<!-- Different JS File with connection to HTML -->*

    <script *src*="first.js"></script>

**2nd Video**

* To select the items of the document or html file we can use *getElementsByTagName*. This will select the Tag and can provide you the html line of that part. Like we can select using h1 tag.
* This will return a HTMLCollection which is array like object. Not totally array or object.

    <h1>Some Fruits Names</h1>

    <ul>

        <li>Banana</li>

        <li>Apple</li>

        <li>Orange</li>

        <li>Strawberry</li>

        <li>Mango</li>

    </ul>

<script>

        const fruitsName = document.getElementsByTagName('li'); *//Stored the list items inside the fruitsName as a collection. Here document means the whole HTML file.*

*for*(const item of fruitsName){  *//taking all the items from the unordered list*

            console.log(item.innerText); *//to retrieve the inner text of the html code*

        }

    </script>

**3rd Video**

getElementByID will work on IDs or single element. And it will return only the element or if there is no such named id, it will return empty array.

getElementsByClassName will work on Classes or multiple elements and will return a collection if it has one or more than elements and even for nothing.

    <h2 *id*="drink">My Favorite Drinks</h2>

    <ul>

        <li>Pepsi</li>

        <li>Coke</li>

        <li>Mountain Deo</li>

        <li>Speed</li>

        <li>Power</li>

    </ul>

    <h2>My Favorite Places</h2>

    <ul>

        <li *class*="places">Bandarban</li>

        <li *class*="places">Sylhet</li>

        <li *class*="places">Coxs Bazar</li>

        <li *class*="places">Kuakata</li>

        <li>Saint Martin</li>

    </ul>

    <script>

        const drinks = document.getElementById('drink');

        drinks.innerText = "Popular Drink Brands";

        const places = document.getElementsByClassName('places');

*for*(const place of places){

            console.log(place.innerText);

        };

    </script>

**4th Video**

querySelectorAll(): This works like other getElementByID, getElementByClassName and TagName but here it requires #idname inside the ‘’ and .classname inside the ‘’. This results in a nodelist not collection.

querySelector(): This will result the first one of all the elements that can be selected. Results nodelist.

*//QuerySelectorAll*

        const places = document.querySelectorAll('.places-container li');

*for*(const place of places){

            console.log(place.innerText);

        }

*//QuerySelector*

        const onePlace = document.querySelector('.places-container li');

        console.log(onePlace.innerText);

**5th Video**

*Style:* In JS styling can be done using style. But before that you have to select it using any of the selectors such as queryselector, getElementByID/Class/Tag. Then use ‘.’ And then style.

Remember then again you will use ‘.’ And then the style you want to set.

In CSS we may have used *background-color* but here if you want to use that, you have to write backgroundColor. That means DOM follow camel casing.

*//We can style using js*

*//Changing Background Color*

        document.getElementById('drink').style.backgroundColor = 'Red'; *//In JS property name should be in camel case, no hyphen*

*//Changing color*

        const items = document.getElementsByClassName('places-container');

*for*(const item of items){

            item.style.color = 'red';

        }

*//Center Align text*

        document.getElementById('drink').style.textAlign = 'center';

*getAttribute():* Before using it please select the tag, class or id or query. They you can use this to retrieve the attribute value. Such as if you want to retrieve the value of class you will write getAttribute(‘class’)

*//Get Attribute*

        document.getElementById('drink').getAttribute('width');

setAttribute(): Before using it please select the tag, class or id or query. Then you can set any Attribute using it, but it takes two values. First one is the attribute name you want to set such as title and the second one is the value of the attribute you want to set.

*//Set Attribute*

*for*(const item of items){

            item.setAttribute('title', 'This is a tourist place');

        }

innerHTML: This will return you the html part inside of the tag/class/id you mentioned. You can change the part using =

*//innerHTML*

        document.getElementById('drink').innerHTML = '<h1>These are prohibited for health</h1>';

innerText: This will return you the text part inside of the tag/class/id you mentioned. You can change the part using =

*//innerText*

        document.getElementById('drink').innerText = 'They are actually prohibited';

*//classList*

*//the first element with the clasname places has been picked using 0 index and all the classes inside the tag has been retrieved using the classList*

        document.getElementsByClassName('places')[0].classList;

*//We can add another class inside that indexed element using add*

        document.getElementsByClassName('places')[0].classList.add('tourist-place');

*//We can remover a class from that indexed element using remove*

        document.getElementsByClassName('places')[0].classList.remove('chittagong');

**Extra**

To work on the className or apply style or anything to class using for of and then apply. Otherwise it doesn’t work

*//Changing color*

        const items = document.getElementsByClassName('places-container');

*for*(const item of items){

            item.style.color = 'red';

        }

\n means line break if you get that

**6th Video**

        const sections = document.querySelectorAll('section');

*for*(const section of sections){

            section.style.backgroundColor = 'tomato';

            section.style.padding = '10px';

            section.style.margin = '10px';

            section.style.borderRadius = '20px'

        }

        const classes = document.getElementById('places-container');

        classes.classList.add('Mirza', 'Mahbub');

        console.log(classes.classList);

        classes.classList.remove('Mirza');

        console.log(classes.classList);

using querySelector we selected the section and then provided style to the elements using JS.

Also we can add and remove classes by selecting the item using getElementByID/Class/Tag

**7th Video**

        const child = document.getElementById('third');

        console.log(child.childNodes);  *//NodeList(5) [text, h2, text, ul#places-container, text]*

        console.log(child.firstChild);  *//#text*

        console.log(child.lastChild);   *//#text*

        console.log(child.firstChild.nextSibling);  *//h2*

        console.log(child.firstChild.nextSibling.previousSibling);  *//#text*

firstChild selects the first child of the parent, lastChild selets the last child of the element. nextSibling provides the next item in the list or collections and preiousSibling provides previous item.

*//Adding items dynamically*

        const addItems = document.getElementById('places-container');

        const listItem = document.createElement('li');

        listItem.innerText = "Khagrachari";

        addItems.append(listItem);

        const sikim = document.createElement('li');

        sikim.innerText = "Sikim";

        addItems.append(sikim);

items can be added dynamically from JS. First you have to select the tag where you want to add. Then create and element using createElement and then append that using appendChild.