

Table of Contents

- [DOM](#)

DOM

What's DOM defines?

- JavaScript Dom/document এর সাহায্য HTML এর প্রতিটা element কে manipulate করে |
- HTML এর element গুলিকে DOM এর সাহায্য Objects বানিয়ে JavaScript manipulate করে
 - Example:

Input:

```
const liCollection = document.getElementsByTagName("li");
console.log(liCollection);
```

output : Array Like object

```
► HTMLCollection(14) [li, li, li, li, li, li, li, li.important-places, li.important-places, li.important-places, li, li, li, li, li]
```

- In More:

Input::

```
const liCollection = document.getElementsByTagName("li");
console.log(liCollection);
for (const li of liCollection) {
  console.log(li);
}
```

Output:

```

▶ HTMLCollection(14) [li, li, li, li, li, li, li, li.important-places, li.important-places, li.important-places, li, li, li, li, li] index.html:70
▶ <li> ... </li> index.html:72
▶ <li> ... </li> index.html:72
▶ <li> ... </li> index.html:72
▶ <li> ... </li> index.html:72
▶ <li> ... </li> index.html:72
▶ <li> ... </li> index.html:72
▶ <li> ... </li> index.html:72
▶ <li class="important-places"> ... </li> index.html:72
▶ <li class="important-places"> ... </li> index.html:72
▶ <li class="important-places"> ... </li> index.html:72
▶ <li> ... </li> index.html:72
▶ <li> ... </li> index.html:72
▶ <li> ... </li> index.html:72
▶ <li> ... </li> index.html:72
▶ <li> ... </li> index.html:72

```

- In More More Details:

Input::

```

const liCollection = document.getElementsByTagName("li");
console.log(liCollection);
for (const li of liCollection) {
  console.log(li.innerText);
}

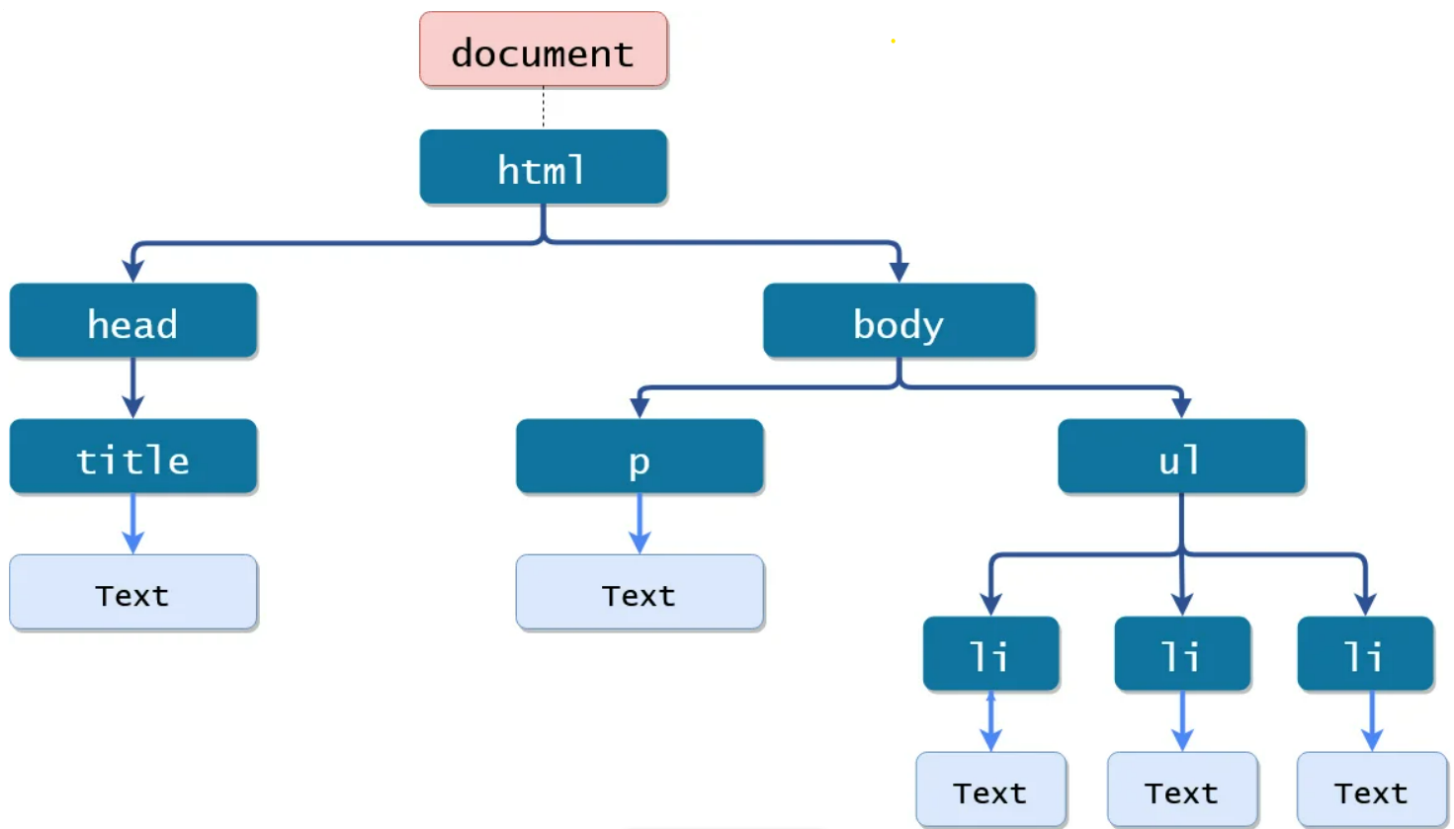
```

Output:

```

▶ HTMLCollection(14) [li, li, li, li, li, li, li, li.important-places, li.important-places, li.important-places, li, li, li, li, li] index.html:70
Jalali set index.html:72
Shafayet index.html:72
Bonobash index.html:72
Apple index.html:72
Guava index.html:72
Banana index.html:72
bandarban index.html:72
khataban index.html:72
shundarban index.html:72
shalbon index.html:72
pahartolibon index.html:72
T-shirt index.html:72
Lungi index.html:72
pajama index.html:72

```



What can JavaScript do with DOM?

- JavaScript can change all the HTML elements in the page.
- JavaScript can change all the HTML attributes in the page.
- JavaScript can change all the CSS styles in the page.
- JavaScript can remove existing HTML elements and attributes
- JavaScript can new HTML elements and attributes
- JavaScript can react to all existing HTML events in the page.
- The events for all HTML elements
- The Properties of all HTML elements
- The methods to access all HTML elements
- All Elements as objects

Find the HTML element in 4 ways

1. getElementById
2. getElementsByClassName
3. getElementsByTagName
4. querySelector & querySelectorAll

Dynamic style, getAttribute, setAttribute, innerText, innerHTML

- style: document.getElementById("myH1").style.color = "red";
- getAttribute: let text = element.getAttribute("class");
- setAttribute: element.setAttribute("class", "democlass");
- innerText: let text = element.innerText;
- innerHTML: document.getElementById("demo").innerHTML = "I have changed!";
- innerHTML: document.getElementById("demo").innerHTML = "
I have changed!
";

Styling dom properties, add and remove css class using JS

```
const sections= document.querySelectorAll('section');
for (const section of sections){
    section.style.border= '3px solid blue';
    section.style.marginBottom='5px';
    section.style.borderRadius='15px';
    section.style.paddingLeft='7px';
    section.style.background='lightgray';
}
/* const placesContainer=document.getElementById('places-container');
placesContainer.style.background='blue'; */

/* Add,remove class */
const placesContainer=document.getElementById('places-container');
placesContainer.classList.add('yellow-bg');
placesContainer.classList.remove('large-text');
```

Event, addEventListener, Event bubble

1. What's event in JS? Find buttons with event handlers

- An Event handler is a routine that deals with the event, allowing a programmer to write code that is executed when the event occurs with the help of event attributes.

2. event bubbling, event capturing, stop propagation and event delegation

- event capturing is the event starts from top element to the target element.
- The stop propagation() method of the event interface prevents further propagation of the current event in the capturing and bubbling phases.
-

2. some javascript events:

- ONCHANGE
- ONCLICK
- ONMOUSEOVER
- ONKEYDOWN
- ONBLUR
- ONLOAD

The following