Document Object Model (DOM)

DOM targetting methods

```
console.log(document);
console.log(document.head);
console.log(document.title);
console.log(document.body);
console.log(document.documentURI);
```

Assessing the DOM

```
// 1. getElementsByTagName
console.log(document.getElementsByTagName("h1"));
console.log(document.getElementsByTagName("h1").length);

// 2. getElementById
console.log(document.getElementById("main"));

// 3. getElementsByClassName
console.log(document.getElementsByClassName("cls"));

// 4. querySelector
console.log(document.querySelector("#id-1"));

// 5. querySelectorAll
console.log(document.querySelectorAll("li"));

// Storing data in variables
const h1 = document.querySelector("h1");
console.log(h1);
```

innerText, textContent, inner HTML

```
const p = document.querySelector("p");
// Formatted & will not show script tag.
console.log(p.innerText);

// Not Formated and will show script tag.
console.log(p.textContent);

// Will Show All the content + HTML
console.log(p.innerHTML);
```

```
// Changing the values.
const h1 = document.querySelector("h1");
h1.innerText = "Text Changed";
h1.innerHTML = "<em>123</em>";
```

classlist, add(), remove(), toggle()

```
const h1 = document.querySelector("h1");
console.log(h1.classList);
// console.log((h1.style.color = "skyblue"));
// console.log((h1.style.fontWeight = "normal"));

// In EDITOR
h1.classList.add("styles");
h1.classList.remove("styles");
h1.classList.toggle("styles");
```

Get and Set methods

```
// href
// value
// type
// getAttribute(attrName)
// setAttribute(attrName, value)

const a = document.querySelector("a");
const input = document.querySelector("input");

// Getting Attribute
console.log(a);
console.log(a.href);
console.log(input.value);
console.log(input.type);

// Setting Attribute
a.href = "https://www.google.com";
console.log((input.value = "Hello"));
console.log((input.type = "password"));
console.log((input.placeholder = "Provide a strong password"));

// **********************************
// getAttribute(attrName)
console.log(input.getAttribute("type"));
```

```
// setAttribute(attrName, value)
console.log(input.setAttribute("placeholder", "Email"));
```

Siblings

```
let firstLi = document.querySelector("li");

console.log(firstLi.parentElement); // ul
console.log(firstLi.parentElement.parentElement); // body
console.log(firstLi.parentElement.parentElement.parentElement); // html
console.log(firstLi.parentElement.parentElement.parentElement.parentElemen
t); // null

let ul = document.querySelector("ul");

console.log(ul.children);         // HTMLCollection of <li> elements
console.log(ul.children[0]);      // First <li>
console.log(ul.children[1]);      // Second <li>
console.log(ul.children[2]);      // Third <li>
console.log(ul.children[2].innerText);   // Text inside third <li>
ul.children[2].innerText = "Apple";      // Changes third <li>'s text to
"Apple"

// Next Element Sibling
console.log(firstLi.nextElementSibling.textContent); // 2nd <li>
console.log(firstLi.nextElementSibling.nextElementSibling.textContent); //
3rd <li>
console.log(firstLi.nextElementSibling.nextElementSibling.nextElementSibli
ng.textContent); // 4th <li>
console.log(firstLi.nextElementSibling.nextElementSibling.nextElementSibli
ng.nextElementSibling.textContent); // 5th <li>


// Previous Element Sibling
let fourthLi = document.querySelector(".fourth"); // Suppose this is the
4th <li>

console.log(fourthLi.previousElementSibling.textContent); // 3rd <li>
console.log(fourthLi.previousElementSibling.previousElementSibling.textCon
tent); // 2nd <li>
```

🌐 JavaScript DOM Manipulation Cheat Sheet

✅ Common DOM Methods

- `createElement()`

- appendChild()
- append()
- prepend()
- insertBefore()
- insertAdjacentElement()
- removeChild()
- remove()

🛠️ Create & Append Element

```
const h1 = document.createElement("h1");
h1.textContent = "Hello World";
document.body.appendChild(h1);
```

➕ appendChild()

```
const ul = document.querySelector("ul");
const newLi = document.createElement("li");
newLi.innerText = "I'm li tag";
ul.appendChild(newLi);  // Adds <li> to the end of the <ul>
```

🔁 insertBefore()

```
const firstLi = document.querySelector("li");
ul.insertBefore(newLi, firstLi);  // Inserts newLi before the first <li>
```

🎯 insertAdjacentElement()

```
const firstP = document.querySelector("p");
const i = document.createElement("i");
i.innerText = "I'm italics";
i.style.color = "skyblue";

firstP.insertAdjacentElement("beforebegin", i);    // Before <p>
firstP.insertAdjacentElement("afterbegin", i);     // Inside <p> at the
beginning
firstP.insertAdjacentElement("beforeend", i);      // Inside <p> at the end
firstP.insertAdjacentElement("afterend", i);       // After <p>
```

🧩 append()

```
const section = document.querySelector("section");
section.append(i, firstLi);  // Adds <i> and <li> to <section>
```

🔼 prepend()

```
const newList = document.querySelector(".new-list");
const a = document.createElement("a");
a.textContent = "HuXn WebDev";
a.href = "https://www.youtube.com/@huxnwebdev";
newList.prepend(a);  // Adds <a> to the top of newList
```

❌ removeChild() & remove()

```
newList.removeChild(a);  // Removes the <a> from newList
newList.remove();        // Removes the entire newList element
```

🧠 Tip:

- `append()` and `prepend()` can accept **multiple nodes**.
- `appendChild()` and `insertBefore()` can only insert **one element at a time**.

---

4 Ways to add events

```
// ----------- BAD WAY-----------
const secondBtn = document.querySelector(".second-btn");
secondBtn.onclick = function () {
  console.log("Hello World");
};

// ----------- GREAT WAY-----------
const best = document.querySelector(".best");

best.addEventListener("click", () => {
  console.log("Yellow");
});

// ----------- Event (e) Object -----------
```

```
// It's an event object which contains information about perticular event.

const para = document.querySelector(".para");

para.addEventListener("click", (e) => {
  console.log(e);
});
```