**1. document.getElementById(id)**

* Returns the element with the specified ID.

js

Copy code

const element = document.getElementById("myId");

**2. document.getElementsByClassName(className)**

* Returns an **HTMLCollection** of all elements with the given class name.

js

Copy code

const elements = document.getElementsByClassName("myClass");

**3. document.getElementsByTagName(tagName)**

* Returns an **HTMLCollection** of elements with the given tag (e.g., div, p).

js

Copy code

const paragraphs = document.getElementsByTagName("p");

**4. document.querySelector(selector)**

* Returns the **first element** that matches a **CSS selector**.

js

Copy code

const firstButton = document.querySelector(".btn");

**5. document.querySelectorAll(selector)**

* Returns **all elements** matching a CSS selector as a **NodeList**.

js

Copy code

const allButtons = document.querySelectorAll(".btn");

**6. document.createElement(tagName)**

* Creates a new HTML element dynamically.

js

Copy code

const newDiv = document.createElement("div");

**7. element.appendChild(childElement)**

* Appends a child to a parent element.

js

Copy code

document.body.appendChild(newDiv);

**8. element.innerHTML / element.textContent**

* innerHTML: Get or set HTML content.
* textContent: Get or set **text only**.

js

Copy code

element.innerHTML = "<strong>Hello</strong>";

element.textContent = "Hello";

**9. element.setAttribute(name, value) / getAttribute()**

* Set or get any attribute (like src, href, etc.).

js

Copy code

img.setAttribute("src", "image.jpg");

**10. element.style.property**

* Dynamically change the style of an element.

js

Copy code

element.style.color = "blue";

**✅ Event-Related**

**1. addEventListener()**

js

Copy code

document.getElementById("btn").addEventListener("click", () => {

alert("Button clicked!");

});

**2. removeEventListener()**

js

Copy code

function sayHello() {

alert("Hello!");

}

btn.addEventListener("click", sayHello);

// Later

btn.removeEventListener("click", sayHello);

**🔍 Traversal & Navigation**

**3. parentElement**

js

Copy code

const parent = document.getElementById("child").parentElement;

**4. children / childNodes**

js

Copy code

const list = document.getElementById("myList");

console.log(list.children); // HTML elements only

console.log(list.childNodes); // Includes text and comment nodes

**5. nextElementSibling / previousElementSibling**

js

Copy code

const next = document.getElementById("item1").nextElementSibling;

**6. closest(selector)**

js

Copy code

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

const form = button.closest("form");

**🧱 Node Manipulation**

**7. remove()**

js

Copy code

document.getElementById("toRemove").remove();

**8. replaceWith()**

js

Copy code

const oldNode = document.getElementById("old");

const newNode = document.createElement("p");

newNode.textContent = "New Element";

oldNode.replaceWith(newNode);

**9. insertBefore()**

js

Copy code

const list = document.getElementById("myList");

const newItem = document.createElement("li");

newItem.textContent = "New Item";

list.insertBefore(newItem, list.children[0]);

**10. cloneNode(deep)**

js

Copy code

const original = document.getElementById("card");

const copy = original.cloneNode(true);

document.body.appendChild(copy);

**🧠 Attributes & Classes**

**11. classList.add(), .remove(), .toggle(), .contains()**

js

Copy code

const box = document.getElementById("box");

box.classList.add("highlight");

box.classList.toggle("hidden");

**12. hasAttribute()**

js

Copy code

const hasHref = document.querySelector("a").hasAttribute("href");

console.log(hasHref); // true or false

**🖋️ Forms and Inputs**

**13. form.submit()**

js

Copy code

document.getElementById("myForm").submit();

**14. input.value**

js

Copy code

const val = document.getElementById("nameInput").value;

**15. input.checked**

js

Copy code

const isChecked = document.getElementById("checkbox").checked;

**📦 Document-Level Access**

**16. document.body, document.head**

js

Copy code

console.log(document.body);

console.log(document.head);

**17. document.title**

js

Copy code

document.title = "New Page Title";

**18. document.cookie**

js

Copy code

document.cookie = "username=JohnDoe; path=/;";

console.log(document.cookie);

**1. event.target vs event.currentTarget**

* **target** is the actual element clicked.
* **currentTarget** is the one the event listener is attached to.

js

Copy code

document.getElementById("container").addEventListener("click", (e) => {

console.log("Clicked on:", e.target);

});

**2. event.preventDefault()**

* Prevents the default behavior (e.g., stop a link from navigating).

js

Copy code

document.querySelector("a").addEventListener("click", (e) => {

e.preventDefault();

alert("Link disabled.");

});

**3. event.stopPropagation() / stopImmediatePropagation()**

* Stops the event from bubbling up or stopping other handlers.

js

Copy code

childDiv.addEventListener("click", (e) => {

e.stopPropagation();

alert("Only child clicked!");

});

**4. element.dataset**

* Access custom data attributes (e.g., data-user-id="123").

html

Copy code

<div id="user" data-user-id="42"></div>

js

Copy code

const userId = document.getElementById("user").dataset.userId;

**5. element.scrollIntoView()**

* Smoothly scrolls an element into the viewport.

js

Copy code

document.getElementById("section3").scrollIntoView({ behavior: "smooth" });

**6. element.focus() and blur()**

* Focuses or removes focus from an input.

js

Copy code

document.getElementById("email").focus();

**7. element.contentEditable**

* Makes any HTML element editable in the browser.

html

Copy code

<div contenteditable="true">Click and edit me!</div>

**8. element.getBoundingClientRect()**

* Returns size and position of the element relative to the viewport.

js

Copy code

const box = document.getElementById("box");

const pos = box.getBoundingClientRect();

console.log(pos.top, pos.left);

**9. MutationObserver**

* Observe DOM changes (like new elements added).

js

Copy code

const observer = new MutationObserver((mutations) => {

console.log("DOM changed:", mutations);

});

observer.observe(document.body, { childList: true, subtree: true });

**10. window.requestAnimationFrame()**

* For optimized animations.

js

Copy code

function animate() {

// update animation frame

requestAnimationFrame(animate);

}

animate();

These are very useful when working with **event handling**, **dynamic rendering**, **animations**, or **building custom UI components**