DOM Manipulation Cheat Sheet

Selecting Elements

`getElementById("id")`

- Selects **one element** by ID.
- Returns a **single element** (or `null`).
- Old but still widely used.

```
document.getElementById("main-title");
```

`getElementsByClassName("class")`

- Selects **all elements** with a class.
- Returns an **HTMLCollection** (not an array).

```
document.getElementsByClassName("card");
```

`getElementsByTagName("tag")`

- Selects **all elements** with a tag name (`div`, `p`, `ul`, etc.).
- Returns an **HTMLCollection**.

```
document.getElementsByTagName("li");
```

`querySelector("selector")`

- Selects the **first matching element**.
- Uses **CSS selectors**, very flexible.

```
document.querySelector("#main-title");   // by ID
document.querySelector(".card");         // by class
document.querySelector("p");             // first <p>
document.querySelector("ul li.active");  // nested selection
```

`querySelectorAll("selector")`

- Selects **all matching elements**.
- Returns a **NodeList** (can use `forEach`, unlike HTMLCollection).

```
document.querySelectorAll(".card");
document.querySelectorAll("p");
```

⚡ **When to use what?**

- **Use `getElementById`** → single element by ID.
- **Use `querySelector`** → first match with CSS-like selector.
- **Use `querySelectorAll`** → all matches (easy looping).

Updating Existing Content

Property	Description
`.innerHTML`	Updates HTML inside the element
`.outerHTML`	Updates the element itself including tags
`.textContent`	Updates only text content, ignores HTML
`.innerText`	Updates visible text (ignores hidden text)

Adding New Elements / Content

```
let box = document.querySelector(".box");

let newPara = document.createElement('p');          // create <p>
let newText = document.createTextNode('sample-text'); // create text
node

newPara.append(newText);   // add text inside <p>
box.append(newPara);        // add <p> to .box
```

Adding to multiple elements

```
let boxes = document.querySelectorAll(".box");

boxes.forEach(box => {
  box.append("hello");
});
```

Using `insertAdjacentHTML`

```
box.insertAdjacentHTML("beforebegin", "<p>Hello before box</p>");
box.insertAdjacentHTML("afterbegin", "<p>Hello at start</p>");
box.insertAdjacentHTML("beforeend", "<p>Hello at end</p>");
box.insertAdjacentHTML("afterend", "<p>Hello after box</p>");
```

Styling Elements

```
box.style.color = 'red';                     // single style
box.style.cssText = 'color:red; background-color:green'; // multiple
styles
box.setAttribute("style", "color:orange");  // alternative
```

Working with Classes & IDs

```
box.setAttribute("id", "heading-id");  // set ID

// Class name as string
console.log(box.className);
```

```
// Class list (like array, easy to manipulate)
box.classList.add("new-class");
box.classList.remove("old-class");
box.classList.toggle("active"); // add if missing, remove if exists
```