

u5

JavaScript: Module ⑧ [DOM]

[check code in GIT REPO]

ex ①: Accessing DOM Elements

concept: Getting Elements From HTML

```
<section id = "example-1">
```

```
<h2> Accessing DOM Elements </h2>
```

```
<p id = "myParagraph">
```

This is a paragraph. (click the button to change me!)

```
</p>
```

```
<button id = "changeTextButton"> ChangeText </button>
```

```
</section>
```

JS:

```
document.getElementById("changeTextButton") -
```

```
  .addEventListener("click", function () {
```

```
    document.getElementById("myParagraph").textContent
```

```
      = "New Text";
```

```
  });
```

↙ gets element by its ID attribute

↙ attaches event like (click) to elements

↘ changes text inside the element

ex ②: DOM Traversal

concept: Finding Element using CSS Selector

```
<section id = "example-2">
```

```
<h2> Traversing the DOM </h2>
```

```
<ul id = "citiesList">
```

```
<li class = "char"> New York </li>
```

```
<li> Tokyo </li> <li> Paris </li> </ul>
```


//_

```
<button id = "highlightFirstCity">highlight first city</button> </section>
```

JS :

```
document.getElementById("highlightFirstCity").addEventListener("click", function() {  
    const firstCity = document.querySelector("#citiesList  
    li");  
    firstCity.classList.add("highlight");
```

Uses querySelector - to grab first in #citiesList

→ Adds the .highlight class → yellow background

ex ③ : Manipulating Element Properties
concept : Changing both content & style

```
<section id = "example-3">  
    <h2> Manipulating DOM Elements </h2>  
    <div id = "coffeeOrder"> Order : <span id =  
    "coffeeType"> Latte </span> </div>  
    <button id = "changeOrder"> Change Order to  
    Char </button> </section>
```

changes
the text

JS :

```
document.getElementById("changeOrder").addEventListener("click", function() {  
    document.getElementById("coffeeType").textContent = "Char";
```


ex ④: Creating & Inserting new Element
 concept: Add new HTML elements Dynamically

```
<section id = "example - 4" >
  <h2> Creating & Inserting New Elements </h2>
  <button id = "addNewItem" > Add a New
    Item to the Shopping List </button>
  <ul id = "Shopping List">
    <li> Bread </li>
    <li> Milk </li> </ul> </section>
```

```
JS: document.getElementById ("AddNewItem").addEventListener (
  "mouseover", function () {
    document.getElementById ("Shopping List").append (child
      (document.createElement ("li")).textContent =
        "Eggs" . });
```

→ on mouse hover over button - created new ("Eggs")
 and appends to # Shopping List.

ex ⑤: Removing Elements

```
<section id = "example - 5">
  <h2> Removing DOM Elements </h2>
  <ul id = "taskList">
    <li> Task 1 </li>
    <li> Task 2 </li>
    <li> Task 3 </li> </ul>
  <button id = "removeLastTask" > Remove </button>
</section>
```


//_

JS: `document.getElementById("removeLastTask").addEventListener("click", function () {
document.querySelector("#taskList li:last-child").
remove();
});`

- Removes the `` (last) inside `#taskList` each time button is clicked

ex ⑥: Event Handling

concept: Responding to User Action

`<section id = "example - 6">`

`<h2> Event Handling in DOM </h2>`

`<button id = "clickMeButton"> Click Me ! </button>`

`</section>`

JS: `document.getElementById("clickMeButton").addEventListener("mouseover", function () {
alert("Button clicked!");
});`

- on hovering over the button, an alert box pops up

ex ⑦: Event Delegation

concept: Handling Events on Multiple Similar Elements

`<section id = "example-7">`

`<h2> Event Delegation </h2>`

`<ul id = "teaList">`

`<li class = "teaItem"> Green Tea `

`<li class = "teaItem"> Chai `

` </section>`


```
JS: document.getElementById("tealList").addEventListener("click",
      function (event) {
        if (event.target.tagName === "LI") {
          event.target.classList.toggle("highlight");
        }
      });
```

- Instead of adding click to , it listens to <ul id="tealList">
- when any is called, it toggles highlight to element

ex (8): Form Handling

concept: Processing User Input

```
<section id = "example-8">
  <h2> Form Handling </h2>
  <form id = "feedback form">
    <label for = "FeedbackInput"> Your feedback : </label>
    <input type = "text" id = "FeedbackInput" placeholder
      = "Enter Feedback" />
    <button type = "Submit"> Submit Feedback </button>
  </form>
  <p id = "Feedback Display"> </p> </section>
```

```
JS: document.getElementById("feedback form").addEventListener(
      "submit", function (event) {
```

stops from
refreshing
page

```
event.preventDefault();
```

```
const feedback = document.getElementById("FeedbackInput")
  .value;
```

Takes input
text & display
it below
form

```
document.getElementById("FeedbackDisplay").textContent
  = `Thank you! Your feedback: ${feedback} has
    been submitted.`;
  });
```


ex @: DOM Content Loaded

```
<section id = "example-9">  
<h2> DOM content Loaded </h2>  
<p id = "domStatus"> waiting For DOM to load...  
</p> </section>
```

```
JS: document.addEventListener("DOMContentLoaded", function  
    () {  
        document.getElementById("domStatus").textContent =  
            "DOM Fully Loaded";  
    });
```

↪ ensures code runs only after DOM exists