Laborator 4
DOM-uri și ELEMENTE

1. Ce este DOM-ul?

DOM (Document Object Model) este reprezentarea internă a paginii HTML în browser, sub forma unui **arbore de noduri**.

- Fiecare tag HTML devine un **nod** în DOM.
- Fiecare element (ex. <p>, , <img>) este un **nod de tip element**.
- Textul din interiorul tag-urilor este un **nod text**.

Exemplu HTML:

```html
<body>
  <h1>Titlu</h1>
  <p>Un paragraf <strong>important</strong>.</p>
</body>
```

Arbore DOM (simplificat):

- body
  - h1
    - #text („Titlu”)
  - p
    - #text („Un paragraf ”)
    - strong
      - #text („important”)

‼ JavaScript poate **citi, modifica, crea sau șterge** aceste noduri, făcând pagina dinamică.

2. Accesarea elementelor DOM

În JavaScript, accesăm elementele prin obiectul global document.

Exemplu definire element:

```html
<label for="textI">Text input</label>
<input type="text" id="textI" name="textI"><br>
```

2.1 Metode de selectare

```javascript
// după id
const titlu = document.getElementById("titlu");

// după clasă (returnează o colecție)
const casute = document.getElementsByClassName("casuta");

// după numele tag-ului
const paragrafe = document.getElementsByTagName("p");

// cu selector CSS (primul găsit)
const primulButon = document.querySelector("button");

// cu selector CSS (toate elementele)
const toateElementele = document.querySelectorAll(".item");
```

3. Modificarea conținutului și a stilului

3.1 Text și HTML

În HTML:

```html
<h1 id="titlu">Salut!</h1>
<p id="descriere">Text inițial.</p>
```

În JS:

```javascript
const titlu = document.getElementById("titlu");
const descriere = document.getElementById("descriere");

titlu.textContent = "Bine ai venit!";   // modifică doar textul
descriere.innerHTML = "Acesta este un <strong>laborator DOM</strong>."; // interpretează și tag-uri HTML
```

3.2 Modificarea stilului direct din JS

```javascript
titlu.style.color = "blue";
titlu.style.fontSize = "32px";
titlu.style.textAlign = "center";
```

- **titlu** – reprezintă elementul HTML selectat anterior în JavaScript
- **style** – proprietatea prin care JavaScript oferă acces la stilurile *inline* ale elementului.
- **color, fontSize, textAlign** – sunt proprietăți CSS exprimate în sintaxă JavaScript

3.3 Lucru cu clase

În HTML:

```html
<p id="mesaj" class="info">Mesaj inițial</p>
```

În CSS:

```css
.info {
  color: #333;
}

.eroare {
  color: red;
  font-weight: bold;
}
```

În JS:

```javascript
const mesaj = document.getElementById("mesaj");

// adăugăm o clasă
mesaj.classList.add("eroare");

// eliminăm o clasă
mesaj.classList.remove("info");

// comutăm o clasă (dacă există o scoate, altfel o pune)
mesaj.classList.toggle("eroare");

//înlocuim o clasă cu alta
mesaj.classList.replace("info", "eroare");
```

Laborator 4- Tehnologii Web
Anul III-AB

4. Crearea, adăugarea și ștergerea elementelor

4.1 Crearea unui nou element

În HTML:

```html
<ul id="lista">Listă 1</ul>
```

În JS:

```javascript
const lista = document.getElementById("lista");

// creăm un li nou
const elementNou = document.createElement("li");
elementNou.textContent = "Element nou";

// îl adăugăm în listă
lista.appendChild(elementNou);
```

4.2 Ștergerea unui element

```javascript
// ștergem primul copil al listei
lista.removeChild(lista.firstElementChild);

// sau direct pe element
elementNou.remove();
```

5. Evenimente (click, input, focus etc.)

Evenimentele permit codului JS să „reacționeze” la acțiunile utilizatorului.

Exemplu simplu:

În HTML:

```html
<button id="butonSalut">Apasă-mă</button>
<p id="rezultat"></p>
```

În JS:

```javascript
const buton = document.getElementById("butonSalut");
const rezultat = document.getElementById("rezultat");

buton.addEventListener("click", function () {
  rezultat.textContent = "Butonul a fost apăsat!";
});
```

Alte exemple de evenimente:

- input (când se tastează într-un câmp)
- focus / blur (când câmpul este selectat / deselectat)
- mouseover / mouseout (când mouse-ul trece peste un element)

6. Preluare și formatarea datei curente

Exemplu pentru preluarea datei și afișarea acesteia la apăsarea unui buton

```html
<!DOCTYPE html>
<html lang="ro">
<head>
    <meta charset="UTF-8">
    <title>Afișează Data</title>
</head>
<body>

    <button id="butonData">Afișează data</button>
    <p id="afisareData"></p>

    <script>
        document.getElementById("butonData").addEventListener("click", function () {

            const d = new Date();

            const zi = d.getDate();
            const luna = d.getMonth() + 1; // lunile încep de la 0
            const an = d.getFullYear();

            document.getElementById("afisareData").textContent = zi + "/" + luna + "/" + an;
        });
    </script>

</body>
</html>
```

Exemplu pentru preluarea datei și formatarea acesteia

```html
<!DOCTYPE html>
<html lang="ro">
<head>
    <meta charset="UTF-8">
    <title>Data Formatată</title>
</head>
<body>

    <button id="butonData">Afișează data formatată</button>
    <p id="afisareData"></p>

    <script>
        // lista lunilor în limba română
        const luni = [
            "Ianuarie", "Februarie", "Martie", "Aprilie", "Mai", "Iunie",
            "Iulie", "August", "Septembrie", "Octombrie", "Noiembrie", "Decembrie"
        ];

        document.getElementById("butonData").addEventListener("click", function () {

            const d = new Date();

            const zi = d.getDate();
            const luna = luni[d.getMonth()];  // obținem luna scrisă
            const an = d.getFullYear();

            const dataFormatata = zi + " " + luna + " " + an;

            document.getElementById("afisareData").textContent = dataFormatata;
        });
    </script>

</body>
</html>
```

Laborator 4- Tehnologii Web
Anul III-AB

Exercițiul 1 – Listă dinamică de activități (HTML + CSS + JS + dată)
Realizați o pagină web numită listaActivitati.html în care să implementați o „listă de activități zilnice”.

Pagina trebuie să conțină:

- un titlu sugestiv, de exemplu: „Lista mea de activități”;
- un paragraf scurt cu instrucțiuni pentru utilizator;
- un container (de exemplu un cu rol de „card”) care să includă:
  - un câmp de tip text pentru introducerea activității (id="inputActivitate");
  - un buton „Adaugă activitate” (id="btnAdauga");
  - o listă neordonată <ul id="listaActivitati"> în care vor fi afișate activitățile.
- Stilizare utilizând un fișier CSS extern

Comportamentul dinamic al paginii va fi realizat într-un fișier JavaScript extern script.js, atașat la listaActivitati.html:

- la apăsarea butonului „Adaugă activitate”:
  - se citește textul introdus în câmpul inputActivitate;
  - dacă textul nu este gol:
    - se creează un nou element <li>;
    - se obține data curentă folosind obiectul Date; luna va fi afișată în format text (de exemplu „Noiembrie”), folosind un tablou de șiruri de caractere pentru lunile anului în limba română;
    - textul elementului <li> va conține atât activitatea, cât și data la care a fost adăugată, de forma:
      Activitate – adăugată la: 16 Noiembrie 2025;
    - elementul <li> este adăugat în interiorul listei <ul id="listaActivitati">;
    - câmpul de input este golit după adăugare;

Exercițiul 2 – Afișare/ascundere de detalii produs (HTML + CSS + JS + dată)
Realizați o pagină web numită detaliiProdus.html care să afișeze un „card de produs” cu detalii ce pot fi afișate sau ascunse la cererea utilizatorului.

Pagina trebuie să conțină:

- un titlu general, de exemplu „Detalii produs”;
- un container de produs (ex. un cu o clasă adecvată) care să includă:
  - titlul produsului, de exemplu <h2>Telefon X Pro</h2>;
  - opțional, o imagine a produsului;
  - un buton cu textul „Afișează detalii” (id="btnDetalii");
  - un care conține:
    - un paragraf cu descrierea produsului;
    - un paragraf care va conține data curentă, de forma:
      „Informațiile sunt valabile la data: 16 Noiembrie 2025.”

(data propriu-zisă poate fi plasată într-un).

- Stilizare utilizând un fișier CSS extern

Comportamentul dinamic va fi implementat într-un fișier JavaScript extern script.js, atașat la detaliiProdus.html:

- la încărcarea paginii:
  - selectați elementul cu id="detalii" și adăugați-i clasa .ascuns, astfel încât secțiunea de detalii să nu fie vizibilă inițial;
  - obțineți data curentă cu obiectul Date; folosiți un tablou de luni pentru a afișa luna în format text (ex. „Noiembrie”), iar apoi injectați în elementul #dataProdus un șir de forma 16 Noiembrie 2025;
- la click pe butonul cu id="btnDetalii":
  - comutați vizibilitatea secțiunii de detalii prin classList.toggle("ascuns") aplicat pe div-ul detalii;
  - în funcție de starea actuală, modificați textul butonului:
    - dacă detaliile sunt vizibile → textul butonului trebuie să fie „Ascunde detalii”;
    - dacă detaliile sunt ascunse → textul butonului trebuie să redevină „Afișează detalii”;