

Лабораторна робота № 8

Тема: Робота з вузлами

Мета: Освоїти додавання, видалення та переміщення вузлів у DOM.

КОРОТКІ ТЕОРЕТИЧНІ ВІДОМОСТІ

DOM (Document Object Model) це ієрархічне представлення HTML-документа, яке дозволяє динамічно змінювати його структуру за допомогою JavaScript.

1. Додавання вузлів у DOM

Для додавання нових елементів у документ використовуються методи:

`document.createElement(tagName)` – створює новий вузол (без додавання в документ).

`parentNode.appendChild(node)` – додає `node` у кінець `parentNode`.

`parentNode.insertBefore(newNode, referenceNode)` – додає `newNode` перед `referenceNode`.

`element.innerHTML += 'Новий текст'` – додає HTML-код всередину елемента (небезпечний спосіб через XSS).

Приклад:

```
let newItem = document.createElement("li");
newItem.textContent = "Новий пункт";
document.querySelector("ul").appendChild(newItem);
```

2. Видалення вузлів у DOM

Для видалення елементів використовуються:

`parentNode.removeChild(childNode)` – видаляє `childNode` із `parentNode`.

`element.remove()` – сучасний метод для видалення елемента.

`element.innerHTML = ''` – очищає весь вміст елемента.

Приклад:

```
let listItem = document.querySelector("li");
listItem.remove();
```

3. Переміщення вузлів у DOM

DOM-елементи можна переміщати між батьківськими вузлами:

`parentNode.insertBefore(movingNode, referenceNode)` – вставляє `movingNode` перед `referenceNode`.

`parentNode.appendChild(movingNode)` – переміщує `movingNode` у кінець `parentNode`.

Приклад (переміщення `li` вгору):

```
let item = document.querySelector("li");
let parent = item.parentNode;
if (item.previousElementSibling) {
    parent.insertBefore(item,
item.previousElementSibling);
}
```

ЗАВДАННЯ ДЛЯ САМОСТІЙНОГО ВИКОНАННЯ

Прості завдання

1. **Додавання нового елемента до списку**
 - Створіть ul із кнопкою "Додати елемент".
 - При натисканні додається новий li з текстом "Новий елемент".
2. **Видалення останнього елемента зі списку**
 - Додайте кнопку "Видалити останній".
 - При натисканні останній li видаляється (якщо список не порожній).
3. **Очищення всіх елементів списку**
 - Додайте кнопку "Очистити список".
 - При натисканні ul стає порожнім.
4. **Додавання заголовка перед списком**
 - Додайте кнопку "Додати заголовок".
 - При натисканні перед ul вставляється h3 з текстом "Мій список".
5. **Додавання елемента на початок списку**
 - При натисканні кнопки "Додати на початок" новий li додається першим у ul.

Завдання середньої складності

6. **Додавання елемента з інпуту**
 - Додайте поле введення (input) і кнопку "Додати".
 - При натисканні текст із input додається як li.
7. **Видалення конкретного елемента зі списку**
 - Кожен li має кнопку "Видалити".
 - При натисканні цей li видаляється.
8. **Переміщення елемента вгору та вниз**
 - Додайте кнопки "Вгору" та "Вниз" для кожного li.
 - При натисканні li змінює свою позицію у списку.
9. **Заміна одного вузла іншим**
 - Додайте p з текстом.
 - При натисканні кнопки "Замінити" він замінюється на h2 з тим самим текстом.
10. **Додавання та видалення карток**

- Кожна картка (div.card) містить текст і кнопку "Закрити".
- При натисканні кнопки картка видаляється.
- Додайте кнопку "Додати картку", яка створює нову картку.

Додаткові завдання

Завдання 11. Дано таблицю.

```
<table>
<thead>
  <tr>
    <th>Ім'я</th><th>Прізвище</th><th>Вік</th>
  </tr>
</thead>
<tbody>
  <tr>
    <td>Сергій</td><td>Петренко</td><td>10</td>
  </tr>
  <tr>
    <td>Юрій</td><td>Даниленко</td><td>15</td>
  </tr>
  <tr>
    <td>Анна</td><td>Ткаченко</td><td>5</td>
  </tr>
  <tr>
    <td>...</td><td>...</td><td>...</td>
  </tr>
</tbody>
</table>
```

Створити сценарій для:

1. Сортування таблиці за колонкою Ім'я.
2. До кожної колонки додати кнопку сортувати та при її натисканні проводити сортування за даною колонкою.

Завдання 12.

Дано масив прізвищ. Створити форму де кожне прізвище поміщено як значення відповідного поля input

Масив: article=["Качурівський В.", "Калиній І.", "Роман Б.", "Ваше прізвище та ім'я"]