

Лабораторна робота №7
Маніпуляція з DOM: розмір, прокрутка, координати.
Події браузера, їх властивості та шаблони обробки.

Мета: створення веб-сторінок з використанням HTML, CSS та JavaScript для закріплення знань, які включають роботу з подіями, анімацією, динамічним додаванням контенту та стилей. Кожне завдання спрямоване на розвиток навичок роботи з DOM (Document Object Model), подіями, анімацією, стилізацією та взаємодією користувача з веб-сторінкою.

Вимоги та самостійна робота:

- Event reference - <https://developer.mozilla.org/en-US/docs/Web/Events>
- Creating and triggering events - https://developer.mozilla.org/en-US/docs/Web/Events/Creating_and_triggering_events

Практичні завдання:

Завдання 1.

Створіть веб-сторінку з наступною структурою:

1. Зовнішній контейнер з фіксованою висотою та можливістю прокрутки.
2. У контейнері розмістіть набір блоків різного розміру та стилю.
3. Додати кнопку "Отримати розміри та координати елементів". При кліці на цю кнопку вивести в блок інформацію про кожен блок, зокрема:
 - розміри (ширина та висота) кожного блоку.
 - координати верхнього лівого кута кожного блоку відносно вікна браузера та відносно контейнера.
 - поточну позицію прокрутки контейнера.
4. Реалізувати можливість зміни розмірів блоків (тільки один обробник подій та prompt) та оновлення відповідної інформації про розміри та координати після змін.
5. Реалізувати можливість додавання елемента (кількість може бути б.я.) з атрибутом data-toggle-id. По кліку по якому необхідно показати/сховати елемент із заданим id (шаблон behavior toggler). Застосувати до контейнеру та блоку з інформацію про кожен блок у контейнері.

Завдання 2.

Створіть веб-сторінку, яка містить наступне:

1. Контейнер з елементами:
 - Створіть контейнер з фіксованою шириною та висотою, який може містити блоки.
 - Додайте можливість додавання та видалення блоків у цьому контейнері. Додавання блоку реалізувати через prompt. Для видалення блоку використати тільки один обробник подій.
 - Кожен блок має випадковий колір, розмір та текстовий вміст.
2. Анімація:
 - Додати анімацію при додаванні та видаленні блоків. Наприклад, блоки можуть з'являтися або зникають плавно за допомогою зміни прозорості або розмірів.

- Додати анімацію прокрутки контейнера до верху або донизу при додаванні або видаленні блоків, якщо контейнер переповнений.
3. Події:
 - Додайте обробку подій при наведенні миші на блоки. Наприклад, колір блоку може змінюватися при наведенні.
 - При кліку на блок вивести інформацію про його розміри та координати.
 4. Стилiзація:
 - Додайте можливість змінювати стилі (наприклад, колір, розмір шрифту тощо) блоків за допомогою випадаючого меню або іншого елемента керування.

Завдання 3.

1. Створити дерево, яке показує/приховує дочірні вузли при кліці (мінімальна кількість вузлів 3).
2. Використати тільки один обробник подій. Клік поза заголовком вузла (на порожньому місці) немає нічого робити.
3. Дизайн дерева на ваш розсуд.

Завдання 4.

1. Реалізувати простий інтерфейс закладок (tabs). Напишіть функцію createTabs, яка приймає вузол DOM, і створює інтерфейс, показуючи дочірні елементи цього вузла. Їй потрібно вставляти список елементів <button> у верхній частині вузла, по одному на кожен дочірній елемент, що містять текст, отриманий з атрибуту data-tabname. Всі, крім одного з дочірніх елементів, мають бути сховані (за допомогою display style none), а видимий поточний вузол можна вибирати натисканням кнопки.
2. Розширити функціонал, щоб активна кнопка мала свій стиль.
3. Додати анімацію.

Вимоги до виконання лабораторної роботи:

1. Використовуйте спосіб призначення обробників за допомогою спеціальних методів addEventListener та removeEventListener.
2. Зверніть увагу на коректність обчислення координат та розмірів при наявності прокрутки.
3. Забезпечте зручний інтерфейс користувачеві.
4. Зверніть увагу на анімацію та її плавність, використовуючи CSS або JavaScript анімації.

Контрольні запитання:

1. Що таке розмір і прокрутка елемента, і як це впливає на веб-розробку?
2. Як можна отримати розміри вікна та прокрутку сторінки у JavaScript?
3. Які існують координатні системи у веб-розробці, і як можна отримати координати елементів у відношенні до цих систем?
4. Які можливості надає знання про розміри, прокрутку та координати для розробки користувацького інтерфейсу та взаємодії з користувачем на веб-сторінці?

5. Що таке події браузера і яка їх роль у веб-розробці?
6. Які основні типи подій існують у браузері?
7. Що таке "бульбашковий механізм" у контексті подій браузера?
8. Що таке делегування подій і як воно використовується для оптимізації обробки подій?
9. Які типові дії браузера можуть викликати події?
10. Як можна ініціювати користувацькі події у JavaScript?