

Лабораторна робота №5

Дата та час.

Планування: `setTimeout` та `setInterval`.

Map та Set. WeakMap та WeakSet

Мета: ознайомлення студентів з роботою з датою та часом у JavaScript, використанням методів `setTimeout` та `setInterval` для планування виконання коду, а також з особливостями роботи із структурами даних Map, Set, WeakMap та WeakSet.

Вимоги та самостійна робота:

- Ознайомитися з основними методами та властивостями об'єкту [Date](#) для роботи з датою та часом у JavaScript.
- Розібратися у принципах роботи з методами [setTimeout](#) та [setInterval](#) для планування виконання функцій в майбутньому.
- Здійснити практичні вправи з використання структур даних [Map](#) та [Set](#) для зберігання унікальних значень та пар ключ-значення.
- Дослідити особливості роботи з [WeakMap](#) та [WeakSet](#), зокрема їхні можливості та обмеження.
- Ознайомитися з можливістю [додавання та видалення класів у HTML елементи](#)
- Розробити власні приклади використання усіх цих засобів у практичних завданнях, що демонструють їхню потужність та ефективність.

Практичні завдання:

Завдання 1 – робота з `className` та `classList`

1. Напишіть програму, яка моделює включення та виключення лампочки за допомогою кліків на кнопку. При кліку на кнопку "Включити", зображення лампочки має змінювати свій стан на ввімкнено, і навпаки.
2. Реалізуйте можливість вибору типу лампочки (наприклад, звичайна лампочка, енергозберігаюча лампочка, світлодіодна лампочка) із наступним включенням та виключенням.
3. Додайте можливість змінювати яскравість лампочки (якщо це можливо для вибраного типу лампочки) через введення користувачем значень через `prompt`.
4. Створіть можливість автоматичного вимкнення лампочки через певний час бездіяльності користувача (наприклад, через 5 хвилин).

Завдання 2 - `setTimeout` та `setInterval`

1. Напишіть програму, яка імітує роботу світлофора з трьома станами: червоний, жовтий і зелений. Кожен стан має тривати певну кількість часу: червоний - 5 секунд, жовтий - 3 секунди, зелений - 7 секунд.
2. Реалізуйте алгоритм світлофора таким чином, щоб після закінчення циклу (зеленого світла) він знову переходив на жовтий і потім на червоний.
3. Послідовність переключень: червоний → жовтий → зелений → миготливий жовтий (мигає 3 рази) → червоний.
4. Додайте можливість змінювати тривалість кожного стану світлофора (червоний, жовтий, зелений) через введення користувачем значень через `prompt`.
5. Розширте програму, додавши анімацію зміни кольору світлофора, виводячи на екран словесний опис поточного стану (наприклад, "червоний", "жовтий", "зелений").
6. Створіть можливість вручну переключати світлофор на наступний стан (наприклад, зелений → жовтий → червоний).

Завдання 3 - Дата та час

Створіть функції, які мають наступні функціональності:

1. Цифровий годинник із анімацією: має показувати годинник, який оновлюється кожную секунду, з мигаючим індикатором секунд.
2. Таймер зворотного відліку: Додайте можливість встановлення таймера зворотного відліку. Користувач може вказати дату і час, коли таймер має закінчитися, і веб-додаток має відображати залишковий час до цього моменту.
3. Календар: Додайте календар, який відображає поточний місяць та рік. Користувач може переглядати різні місяці та роки, використовуючи `input` відповідного типу.
4. Користуючись календарем задайте свій день народження і порахуйте, через скільки часу воно настане (місяців, днів, годин, хвилин, секунд).

Завдання 4 - `Map` та `Set`, `WeakMap` та `WeakSet`

Створіть програму для обліку продуктів у вашому інтернет-магазині використовуючи `Map`, `Set`, `WeakMap`, та `WeakSet`. Програма повинна мати такі функціональні можливості:

1. Додавання продукту: Користувач може додавати нові продукти у каталог, вказуючи назву продукту, його ціну та кількість на складі.
2. Видалення продукту: Користувач може видаляти продукти з каталогу за допомогою їхнього ідентифікатора.
3. Оновлення інформації про продукт: Користувач може оновлювати інформацію про продукт, таку як ціна та кількість на складі.
4. Пошук продукту за назвою: Користувач може шукати продукти за їхнім ім'ям і отримувати повну інформацію про них.
5. Відстеження замовлень: Програма повинна вести облік замовлень і автоматично зменшувати кількість продуктів на складі після кожного замовлення.

6. Використання Map, Set, WeakMap, та WeakSet: Використайте структури даних Map та Set для зберігання інформації про продукти та замовлення. Використайте WeakMap та WeakSet для зберігання додаткової інформації, наприклад, історії змін у продуктах або посилань на користувачів.

Контрольні питання:

1. Що таке об'єкт Date у JavaScript і які методи він надає для роботи з датою та часом?
2. Яка різниця між методами setTimeout та setInterval у JavaScript? Для чого вони використовуються?
3. Що таке Map та Set у JavaScript? Яка різниця між ними?
4. Які основні операції можна виконувати з даними, що зберігаються в Map та Set?
5. Що таке WeakMap та WeakSet у JavaScript? Які особливості їхнього використання порівняно з Map та Set?
6. В чому основні відмінності між Map та WeakMap?
7. Як можна використовувати setTimeout та setInterval для виконання певних завдань з певною затримкою чи інтервалом?
8. Які сценарії використання WeakMap та WeakSet можна запропонувати для оптимізації роботи вашого додатку?