

Лабораторна робота №8 Маніпуляція з DOM. Події інтерфейсу.

Мета: закріпити практичні навички по роботі з DOM (Document Object Model) та подіями інтерфейсу у веб-розробці, створення веб-елементів.

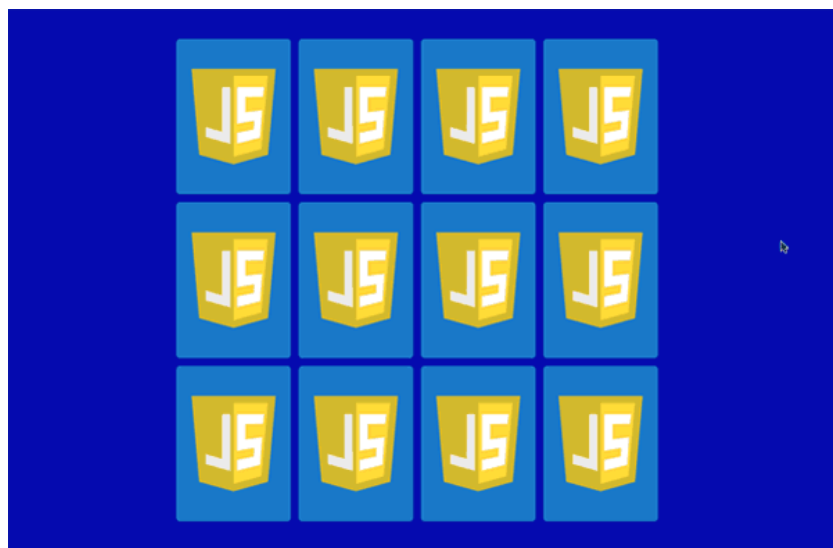
Вимоги та самостійна робота:

1. Shuffle Array <https://css-tricks.com/snippets/javascript/shuffle-array/>
2. CSS3 Transitions https://www.w3schools.com/css/css3_transitions.asp
3. Create a CSS Flipping Animation <https://davidwalsh.name/css-flip>
4. Вступ до подій <https://uk.javascript.info/events>
5. Події інтерфейсу <https://uk.javascript.info/event-details>
6. Карусель <https://uk.javascript.info/task/carousel>
7. Drag and Drop https://www.w3schools.com/html/html5_draganddrop.asp
8. HTML Drag and Drop API https://developer.mozilla.org/en-US/docs/Web/API/HTML_Drag_and_Drop_API

Практичні завдання:

Завдання 1.

Створити гру Memory Pair Game - карткову гру, в якій всі картки кладуться на поверхню лицьовою стороною вниз, і кожен раз перевертаються по дві картки. Мета гри - знайти пари відповідних карток. Гра закінчується, коли буде підібрана остання пара. Ви можете використовувати будь-який стиль та анімацію.



Основні функціональні можливості:

- Відображення ігрового поля з картками, які знаходяться лицьовою стороною вниз.
- Додати блок "Налаштування гри" для вибору рівня складності та налаштування параметрів гри:
 - Можливість обрати розмір ігрового поля (кількість рядків і стовпців). Мінімальний розмір ігрового поля 4 x 4. Врахуйте, що необхідно буде заготовити відповідну кількість картинок.
 - Вибір складності (easy, normal, hard) - пов'язаний з таймером зворотного відліку (наприклад. easy - 3 хв., normal - 2 хв., hard - 1 хв.). Рівень до дефолту - easy.
- Додати кнопку "Нова гра", щоб гравець міг почати нову гру, скинувши всі параметри налаштувань.
- Додати кнопку "Почати гру", щоб гравець міг почати гру з відповідними налаштуваннями гри.
- Відображення таймеру зворотного відліку, який починає відлік після початку гри.
- Функція перевертання картки, щоб побачити зображення.
- Перевірка відповідності карток: якщо вони збігаються, то вони залишаються оберненими; якщо ні, то вони повертаються лицьовою стороною вниз.
- Відображення поточного рахунку гравця (кількість підібраних пар).
- Завершення гри та виведення повідомлення про перемогу (поразку), коли всі пари карток знайдено або не знайдено за встановлений час. Показати час, який знадобився гравцеві на завершення гри.

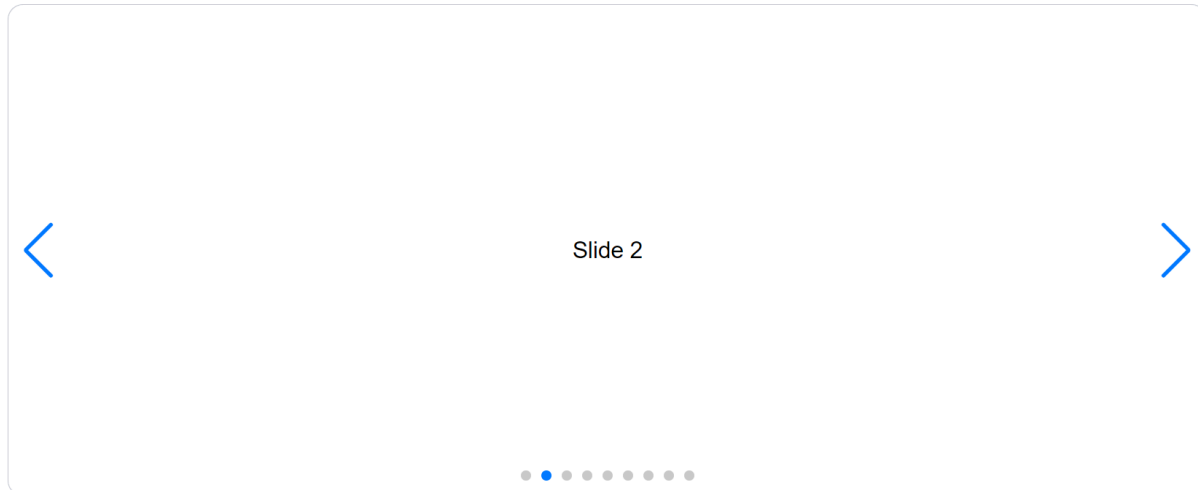
Додаткові функціональні можливості:

- Можливість обирати або грає 1 гравець або 2 гравця.
- Для кожного гравця можна задати ім'я.
- У налаштуваннях додати кількість раундів (по дефолту 1).
- Після n-раундів виводити ім'я переможця зі статистикою по кожному раунду.

Завдання 2.

Створити веб-елемент слайдер (карусель), який дозволяє юзеру прокручувати контент (зображення, тексти, відео тощо) горизонтально за допомогою кнопок, стрілок.

Приклад: <https://swiperjs.com/demos>

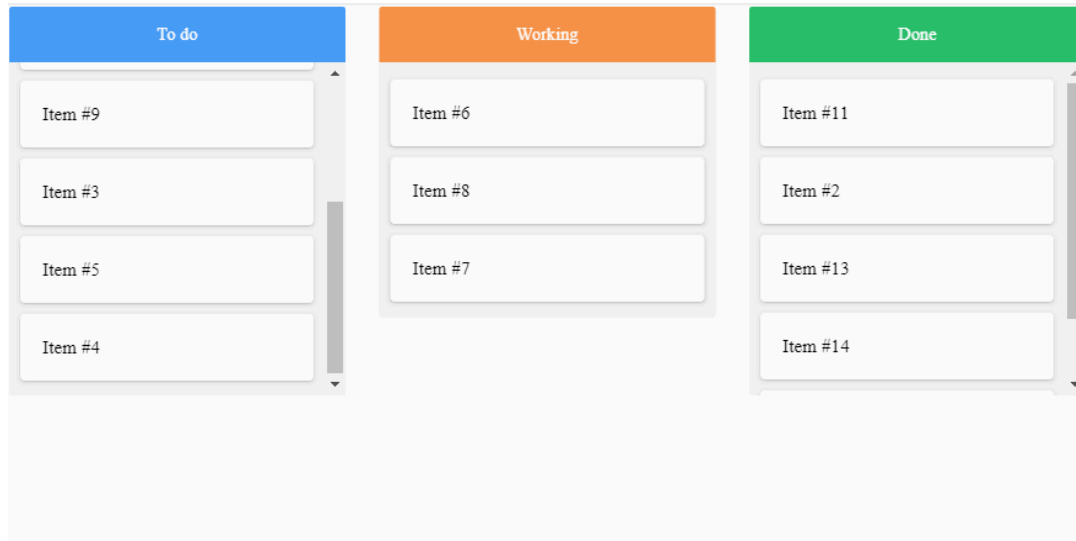


Основні функціональні можливості:

- Прокрутка контенту з анімацією (переміщення контенту вперед або назад). Якщо юзер дійшов до останнього слайду, перемістити знову на перший (continuous loop mode).
- Навігаційні елементи: кнопки у вигляді стрілок (slider-arrows), елементи керування (pagination, slider-dots), які дозволяють користувачам переміщатися між слайдами. Зміна поточного елемента за допомогою клавіш ← та → (keyboard events).
- Анімація: анімаційні ефекти для плавного зміщення контенту.
- Параметри конфігурації: швидкість анімації (duration of transition between slides (in ms)), автоматичне прокручування (autoplay), відображення slider-arrows, відображення pagination, slider-dots.
- Якщо ввімкнено автоматичне прокручування (autoplay) і курсор миші знаходиться на слайді, то зупиняти автоматичне прокручування, доти доки
- юзер не забере курсор миші.
- Адаптивність.

Завдання 3.

Створити веб-сторінку для управління завданнями у форматі Kanban Board (див рисунок). Завдання можна зробити статичними. Головне реалізувати переміщення завдань з використанням HTML Drag and Drop API. Дизайн на ваш розсуд.



Контрольні запитання:

1. Що таке події браузера і яке їх призначення?
2. Що таке "бульбашковий механізм" у контексті подій браузера? Як він працює?
3. Як використовується делегування подій для ефективної обробки подій в браузері?
4. Які типові дії браузера можна спостерігати при взаємодії з веб-сторінкою?
5. Як можна запустити користувацьку подію в браузері?
6. Що таке події миші та які є типові приклади використання?
7. В чому полягає різниця між подіями mouseover/out та mouseenter/leave?
8. Як відбувається взаємодія з подіями миші при використанні Drag'n'Drop?
9. Які події вказівника доступні в браузері? Для чого вони використовуються?
10. Що таке події клавіатури keydown та keyup і як вони відрізняються?
11. Як відбувається обробка подій прокрутки сторінки у браузері?