

## Лабораторна робота №7

### Тема: HTML5 API. Cookie. LocalStorage. JSON. Класи

**Мета роботи:** ознайомитися з можливостями HTML5 API та навчитися використовувати функції на практиці, набути навичок у роботі з основами ООП в JS.

#### Завдання на лабораторну роботу, частина 1

0. Ознайомитись з теоретичними відомостями:

- Web Workers API:

[https://www.tutorialspoint.com/html5/html5\\_web\\_workers.htm](https://www.tutorialspoint.com/html5/html5_web_workers.htm)

- Geolocation API:

[https://www.tutorialspoint.com/html5/geo\\_demo.htm](https://www.tutorialspoint.com/html5/geo_demo.htm)

[https://www.w3schools.com/js/js\\_api\\_geolocation.asp](https://www.w3schools.com/js/js_api_geolocation.asp)

- History API:

[https://www.w3schools.com/js/js\\_api\\_history.asp](https://www.w3schools.com/js/js_api_history.asp)

- LocalStorage та SessionStorage API:

[https://www.tutorialspoint.com/html5/html5\\_web\\_storage.htm](https://www.tutorialspoint.com/html5/html5_web_storage.htm)

[https://www.w3schools.com/js/js\\_api\\_web\\_storage.asp](https://www.w3schools.com/js/js_api_web_storage.asp)

- Application Cache API:

<https://webplatform.github.io/docs/apis/appcache/ApplicationCache/>

1. Реалізувати пошук простих чисел у діапазоні від А од В за допомогою Web Workers API. Сторінка повинна містити поля для введення чисел «А», «В», кнопки «Старт», «Стоп» та поле *textarea* для виведення результату.

За допомогою High Resolution Time API визначити час пошуку кожного простого числа. Таким чином, у *textarea* повинно виводитись кожне знайдене просте число і біля нього час його пошуку.

За допомогою LocalStorage API реалізувати збереження поточного результату та забезпечення можливості початку пошуку з останнього знайденого числа у випадку, якщо браузер був закритий.

Використати Application Cache API для забезпечення роботи сторінки у випадку втрати з'єднання з Інтернетом.

За допомогою Notification API реалізуйте інформування користувача про завершення роботи воркера (виведіть повідомлення, що прості числа знайдені).

За допомогою Page Visibility API відслідкуйте, що користувач зробив вкладку неактивною і через 1 хвилину відправте йому через Notification API повідомлення.

За допомогою Geolocation API виведіть географічні координати користувача (за бажанням – відобразіть їх на карті Google Maps).

#### Завдання на лабораторну роботу, частина 2

0. Опрацювати теоретичні відомості за посиланнями:

<https://uk.javascript.info/json>

<https://uk.javascript.info/cookie>

<https://uk.javascript.info/localstorage>

<https://uk.javascript.info/class>

<https://uk.javascript.info/class-inheritance>

<https://uk.javascript.info/static-properties-methods>

## JSON

1. Створіть на сторінці текстовий input, який буде приймати імена файлів як масив в форматі JSON (наприклад, [" file1.jpg ", " file2.gif ", " file34.gif " ] )

- По натисканню на кнопку повинні з'явитися зображення в маленькому розмірі, по натисканню на які відкриватися ці зображення в реальному розмірі
- Якщо в текстовому полі введені дані не відповідають правильному формату JSON, необхідно повідомити про це користувача

Приклад веб-сторінки:

[" file1.jpg ", " file2.gif ", " file34.gif " ]

ПОКАЗАТИ

OK



vgdvgh

ПОКАЗАТИ

**Невірний формат JSON**

## COOKIE

2. Створити сторінку, розмір текстового шрифту на якій можна змінювати за допомогою кнопок A + / A-. Вибір користувача запам'ятовувати в cookie і при наступному вході на сторінку вибраний шрифт застосовувати автоматично.

3. Дано поле `textarea`. У нього можна ввести дані, потім відредагувати їх, потім ще відредагувати. Нехай `textarea` зберігає історію своїх змін (навіть після перезавантаження сторінки). Зверху над `textarea` повинні з'явитися стрілки, за допомогою яких можна переміщатися по історії. Запам'ятовувати зміни в cookie

## LOCALSTORAGE

4. Дана форма з `input`, `textarea`, `checkbox`, `radio`, `select` і тп. Користувач вводить якісь дані і закриває сторінку (не факт, що він заповнив всю форму). Зробіть так, щоб при наступному заході на сторінку введені ним раніше дані стояли на своїх місцях.

*Примітка:* Для виконання цього завдання потрібно створити об'єкт, в який записувати властивості зі значеннями, які ввів користувач. Потім цей об'єкт серіалізувати в JSON-формат і запам'ятати в локальне сховище. І потім потрібно назад зчитати зі сховища дані, десеріалізувати їх назад в об'єкт і заповнити форму відповідними значеннями

5. Записати момент, на якому користувач зупинився при перегляді відео в `localStorage`, а потім відновлювати його при відкритті сторінки.

## КЛАСИ

6. Створити батьківський клас **Shape**, який буде описувати геометричну фігуру. Клас має містити 2 методи:

- 1) створення фігури(квадрату) без кольору, з контуром, щоб його було видно
- 2) заправку кольору (заповнює фарбу до 100%)

Створити 2 класи-нащадки (коло, прямокутник, трикутник, овал тощо). Класи мають перевизначити:

- 1) батьківський конструктор, в який додати ще змінну - колір фігури
- 2) метод створення фігури, який буде зафарбовувати фігуру та застосовувати деякі нюанси створюваної фігури (заокруглення тощо)

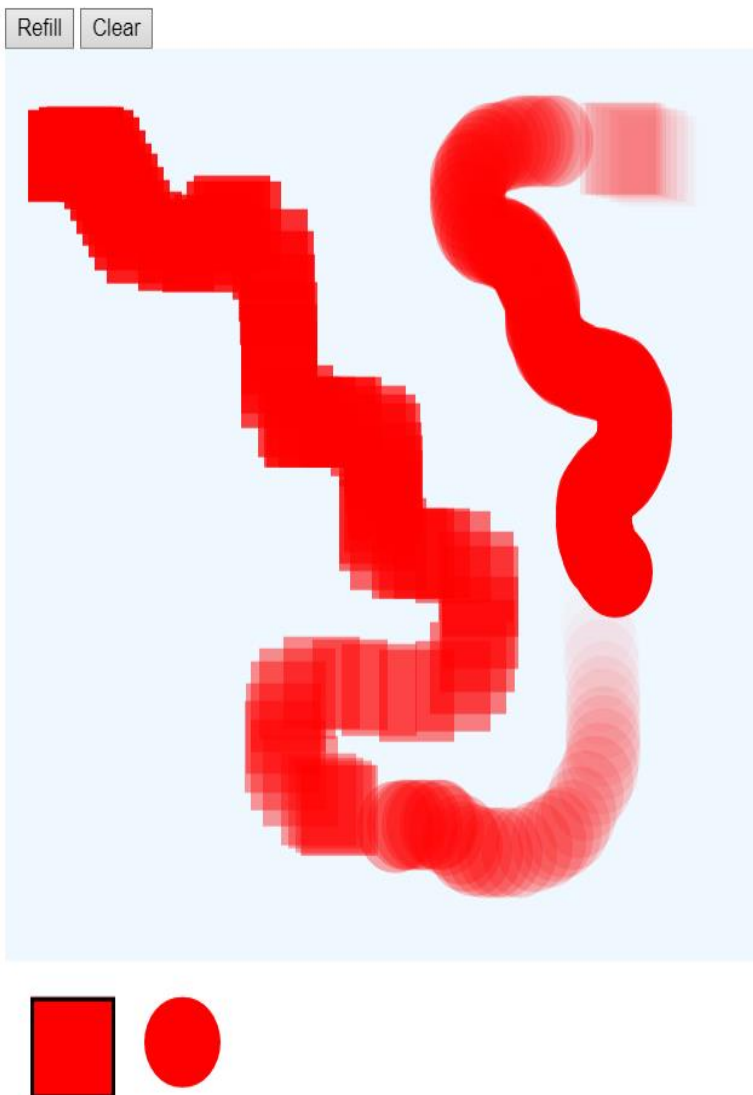
Написати код, щоб при переміщенні вибраної фігури відбувалося малювання лінії за траєктор

ією миші, причому при малюванні фігури повинна витрачатися фарба, і врешті решт лінії вже не буде видно при малюванні.

Після заправки кольору повинна знову малюватися лінія насиченим кольором, який також починає витрачатися.

При малюванні лінії вибраною фігурою колір витрачається, і якщо колір не заправлений, то при виборі іншої фігури лінія також буде продовжуватися малюватися ненасиченим кольором (застосувати статичні методи та властивості)

Приклад веб-сторінки:



2. Створіть репозиторій IPLab7 Закомітьте у нього виконані завдання.  
Надати доступ до репозиторію викладачам: