

## Лабораторна робота №6

### Тема: **Cookie. LocalStorage. JSON. Класи**

**Мета роботи:** вивчити засоби по роботі з cookie, localStorage, JSON, освоїти роботу з класами в JavaScript

#### Завдання на лабораторну роботу

**Завдання 0:** Ретельно вивчити теорію за посиланнями:

<https://learn.javascript.ru/json>  
<https://learn.javascript.ru/cookie>  
<https://learn.javascript.ru/localstorage>  
<https://learn.javascript.ru/class>  
<https://learn.javascript.ru/class-inheritance>  
<https://learn.javascript.ru/static-properties-methods>  
<https://learn.javascript.ru/network>

#### Завдання 1. **COOKIE**

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 потрібно в cookie

#### Завдання 2. **LOCALSTORAGE**

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.

#### Завдання 3. **JSON**

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

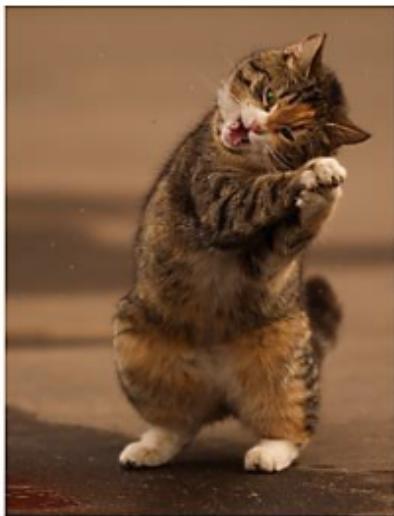
- По натисканню на кнопку повинні з'явитися зображення в маленькому розмірі, по натисканню на які відкриватися ці зображення в реальному розмірі
- Якщо в текстовому полі введені дані не відповідають правильному формату JSON, необхідно повідомити про це користувача

### Приклад веб-сторінки:

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

ПОКАЗАТИ

ОК



vgdvgh

ПОКАЗАТИ

Невірний формат JSON

### Завдання 4. КЛАСИ

- 1) Реалізувати клас **Student**, що описує студента, та створити масив студентів (прізвище, ім'я, оцінки з предметів)
- 2) Реалізувати клас **ListOfStudents** для генерації html-коду таблиці зі списком студентів:
  - масив студентів необхідно передавати через конструктор
  - таблицю отримувати з допомогою методу **getTableList()**

Створити об'єкт класу **ListOfStudents** і вивести результат роботи методу **getTableList()**.

- 3) Реалізувати клас **StylesTable**, який успадковується від класу **ListOfStudents**
  - додати метод **getStyles()**, який повертає рядок зі стилями таблиці в тегах `style`.
  - перевизначити метод **getTableList()**, який додає стилі до того що повертає метод **getTableList()** з батьківського класу.
  - реалізувати метод **getAvg()**, що підраховує середній бал кожного студента і додає його до кінця рядка кожному студенту.

- реалізувати метод **getTotalAvg()**, що підраховує загальний середній бал групи

Створити об'єкт класу **StylesTable** і вивести результат роботи методу **getTableList()**, **getAvg()**, **getTotalAvg()**.

Приклад результату:

Name	LastName	Math	History	JS	Avg
Федорко	Петро	3	4	5	4.00
Остапенко	Сергій	4	5	5	4.67
Колос	Олеся	4	3	3	3.33

**Середній бал по групі = 4**

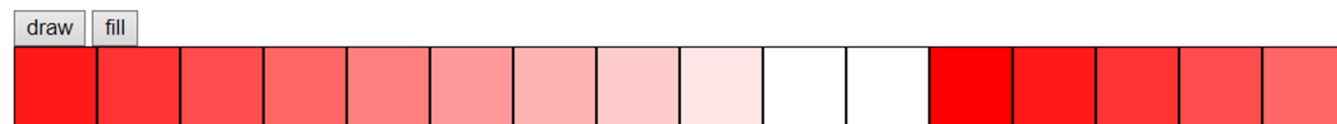
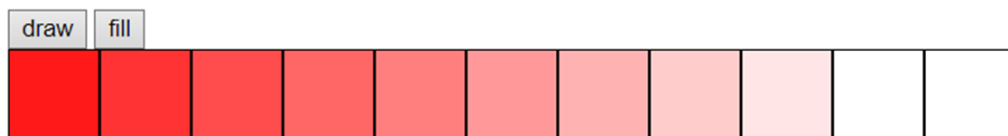
## Завдання 5. СТАТИЧНІ МЕТОДИ ТА ВЛАСТИВОСТІ

Створіть клас **Shape**, яке має 2 властивості та 2 методи:

- статична властивість **total** – кількість фарби
- а - радіус кола або довжина сторони квадрата)
- статичний метод **fill()**, який заправлятиме фарбу до 100%
- метод **draw()**, який малюватиме фігуру

При створенні об'єкта (кнопка **draw**) малюється фігура певного розміру з кількістю кольору, що дорівнює змінній **total**. Кожна нова фігура витрачає певну кількість фарби і настане момент, коли фарби не залишиться. При натисканні кнопки **fill** кількість фарби поповнюється на 100%. Продемонструвати роботу методів.

Приклад результату:



## Завдання 6. FETCH API

Користуючись запитом до API ресурсу <https://dog.ceo/api/breeds/image/random> відобразити на сторінці галерею фотографій собак по натисканню на кнопку, наприклад:



*Закомітити виконані завдання та надати доступ своїм викладачам*