**Лабораторна робота №10**

**Тема: LocalStorage. JSON. Наслідування класу. Статичні властивості та методи**

***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:**

[Методи JSON, toJSON](https://uk.javascript.info/json)

[LocalStorage, sessionStorage](https://uk.javascript.info/localstorage)

[Базовий синтаксис класу](https://uk.javascript.info/class)

[Наслідування класу](https://uk.javascript.info/class-inheritance)

[Статичні властивості та методи](https://uk.javascript.info/static-properties-methods)

**Завдання 1**

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.

Лістининг коду

HTML

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Task 1</title>

    <link rel="stylesheet" href="style.css">

</head>

    <body>

        <form id="myForm">

            <label for="name">Ім'я:</label>

            <input type="text" id="name" name="name">

            <label for="email">Email:</label>

            <input type="text" id="email" name="email">

            <label for="message">Повідомлення:</label>

            <textarea id="message" name="message"></textarea>

            <label>Стать:</label>

            <div class="gender-wrapper">

                <div>

                    <input type="radio" id="male" name="gender" value="male">

                    <label for="male">Чоловіча</label>

                </div>

                <div>

                    <input type="radio" id="female" name="gender" value="female">

                    <label for="female">Жіноча</label>

                </div>

            </div>

            <label for="subscribe">Підписатися:</label>

            <input type="checkbox" id="subscribe" name="subscribe">

            <label for="country">Країна:</label>

            <select id="country" name="country">

                <option value="ukraine">Україна</option>

                <option value="germany">Германія</option>

                <option value="usa">США</option>

            </select>

            <button type="button" onclick="saveData()">Зберегти дані</button>

        </form>

        <script src="https://code.jquery.com/jquery-3.6.4.min.js"></script><!-- Підключення бібліотеки jquery-->

        <script src="main.js"></script>

    </body>

</html>

CSS

body {

    font-family: Arial, sans-serif;

    background-color: #f4f4f4;

    margin: 0;

    padding: 0;

    display: flex;

    justify-content: center;

    align-items: center;

    height: 100vh;

}

form {

    background-color: #fff;

    padding: 20px;

    border-radius: 8px;

    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

    width: 300px;

    box-sizing: border-box;

}

label {

    display: block;

    margin-bottom: 8px;

}

input,

textarea,

select,

button {

    width: 100%;

    padding: 8px;

    margin-bottom: 12px;

    box-sizing: border-box;

}

.gender-wrapper {

    display: flex;

    justify-content: space-between;

    margin-bottom: 12px;

}

button {

    background-color: #4caf50;

    color: #fff;

    border: none;

    border-radius: 4px;

    cursor: pointer;

}

button:hover {

    background-color: #45a049;

}

JS

$(document).ready(function () {

    loadData();

});

// Збереження даних в localStorage при зміні полів форми

$("#myForm").on("input change", "input, textarea, select", function () {

    saveData();

});

// Збереження даних в localStorage при закритті сторінки

$(window).on("beforeunload", function () {

    saveData();

});

function saveData() {

    var form = $("#myForm");

    form.find("input, textarea, select").each(function () {

        var id = $(this).attr("id");

        var value;

        if ($(this).is(":checkbox"))

            value = $(this).prop("checked");

        else

            if ($(this).is(":radio"))

                value = $(this).prop("checked");

            else

                value = $(this).val();

        localStorage.setItem(id, value);

    });

}

function loadData() {

    var form = $("#myForm");

    form.find("input, textarea, select").each(function () {

        var id = $(this).attr("id");

        var savedValue = localStorage.getItem(id);

        if (savedValue !== null) {

            if ($(this).is(":checkbox"))

                $(this).prop("checked", savedValue === "true");

            else

                if ($(this).is(":radio"))

                    $(this).prop("checked", savedValue === "true");

                else

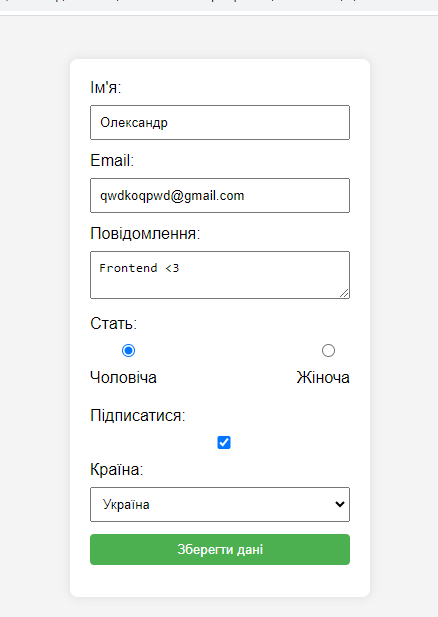
                    $(this).val(savedValue);

        }

    });

}

Результат

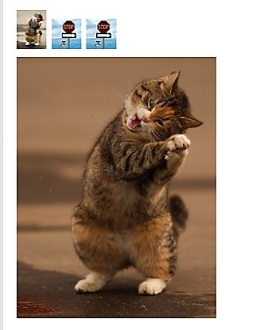


**Завдання 2**

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:*

**

Лістининг коду

HTML

<!DOCTYPE html>

<html lang="en">

    <head>

        <meta charset="UTF-8">

        <meta name="viewport" content="width=device-width, initial-scale=1.0">

        <title>Task 2</title>

        <link rel="stylesheet" href="style.css">

    </head>

    <body>

        <input type="text" id="jsonInput" placeholder="Введіть JSON масив файлів" value='["image1.jpg", "image2.jpg", "image4.jpg", "image5.jpg"]'>

        <button onclick="showImages()">Показати зображення</button>

        <div id="errorMessage"></div>

        <div id="imageContainer"></div>

        <div id="fullImageContainer">

            <img id="fullImage" onclick="hideFullImage()" alt="Full Image">

        </div>

        <script src="main.js"></script>

    </body>

</html>

JS

function showImages() {

    var jsonInput = document.getElementById("jsonInput").value;

    try {

        var filesArray = JSON.parse(jsonInput);

        document.getElementById("errorMessage").innerHTML = "";

        document.getElementById("imageContainer").innerHTML = "";

        for (var i = 0; i < filesArray.length; i++) {

            var thumbnail = document.createElement("img");

            thumbnail.src = filesArray[i].trim();

            thumbnail.className = "thumbnail";

            thumbnail.title = "Натисніть, щоб відкрити в реальному розмірі";

            thumbnail.addEventListener("click", function () {

                showFullImage(this.src);

            });

            document.getElementById("imageContainer").appendChild(thumbnail);

        }

    } catch (error) {

        document.getElementById("errorMessage").innerHTML = "Помилка у форматі JSON";

    }

}

function showFullImage(src) {

    var fullImage = document.getElementById("fullImage");

    fullImage.src = src;

    document.getElementById("fullImageContainer").style.display = "flex";

}

function hideFullImage() {

    document.getElementById("fullImageContainer").style.display = "none";

}

CSS

body {

    font-family: Arial, sans-serif;

    margin: 0;

    padding: 20px;

    display: flex;

    flex-direction: column;

    align-items: center;

}

#imageContainer {

    display: flex;

    flex-wrap: wrap;

    justify-content: center;

}

.thumbnail {

    margin: 10px;

    cursor: pointer;

    width: 40px;

    height: 40px;

    object-fit: cover;

    transition: transform 0.3s ease-in-out;

}

.thumbnail:hover {

    transform: scale(1.2);

}

#jsonInput {

    margin-bottom: 10px;

}

#errorMessage {

    color: red;

    margin-bottom: 10px;

}

#fullImageContainer {

    display: none;

}

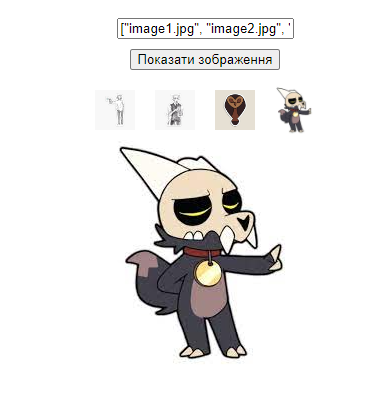
#fullImage {

    max-width: 100%;

    max-height: 100%;

}

Результат



**Завдання 3**

1. Реалізувати клас **Student**, що описує студента, та створити масив студентів (прізвище, ім'я, оцінки з предметів)

1. Реалізувати клас **ListOfStudents** для генерації html-коду таблиці зі списком студентів:

* масив студентів необхідно передавати через конструктор
* таблицю отримувати з допомогою методу **getTableList()**

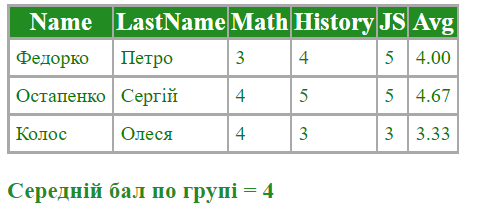
*Створити об'єкт класу* ***ListOfStudents*** *і вивести результат роботи методу* ***getTableList()****.*

1. 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:***

******

Лістининг коду

HTML

<!DOCTYPE html>

<html lang="en">

    <head>

        <meta charset="UTF-8">

        <meta name="viewport" content="width=device-width, initial-scale=1.0">

        <title>Task 3</title>

    </head>

    <body>

        <script src="main.js"></script>

    </body>

</html>

JS

class Student {

    constructor(lastName, firstName, grades) {

        this.lastName = lastName;

        this.firstName = firstName;

        this.grades = grades;

    }

    getAverageGrade() {

        if (this.grades.length === 0) return 0;

        const sum = this.grades.reduce((acc, grade) => acc + grade, 0);

        return sum / this.grades.length;

    }

}

class ListOfStudents {

    constructor(students) {

        this.students = students;

    }

    getTableList() {

        let tableHtml = '<table border="1"><tr><th>Last Name</th><th>First Name</th><th>Subject 1</th><th>Subject 2</th><th>Subject 3</th><th>Average Grade</th></tr>';

        this.students.forEach(student => {

            tableHtml += `<tr><td>${student.lastName}</td><td>${student.firstName}</td><td>${student.grades[0]}</td><td>${student.grades[1]}</td><td>${student.grades[2]}</td><td>${student.getAverageGrade().toFixed(2)}</td></tr>`;

        });

        tableHtml += '</table>';

        return tableHtml;

    }

}

class StylesTable extends ListOfStudents {

    getStyles() {

        return '<style>table { border-collapse: collapse; width: 50%; } th, td { padding: 10px; text-align: left; }</style>';

    }

    getTotalAvg() {

        const totalAvg = (this.students.reduce((acc, student) => acc + student.getAverageGrade(), 0) / this.students.length).toFixed(2);

        return `<div style="background-color: lightgreen; padding: 10px; margin-top: 10px;"><strong>Total Average Grade:</strong> ${totalAvg}</div>`;

    }

}

const studentsArray = [

    new Student('Doe', 'John', [3, 4, 5]),

    new Student('Smith', 'Jane', [4, 5, 5]),

    new Student('Johnson', 'Bob', [4, 3, 3]),

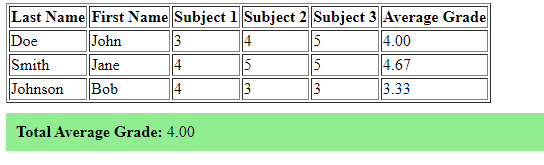
];

const styledTable = new StylesTable(studentsArray);

document.body.innerHTML += styledTable.getTableList();

document.body.innerHTML += styledTable.getTotalAvg();

Результат

****

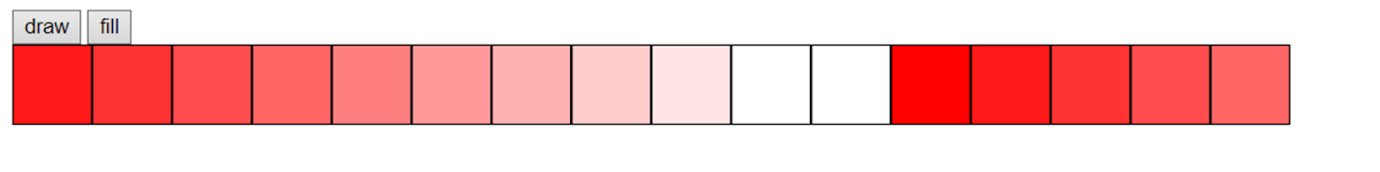
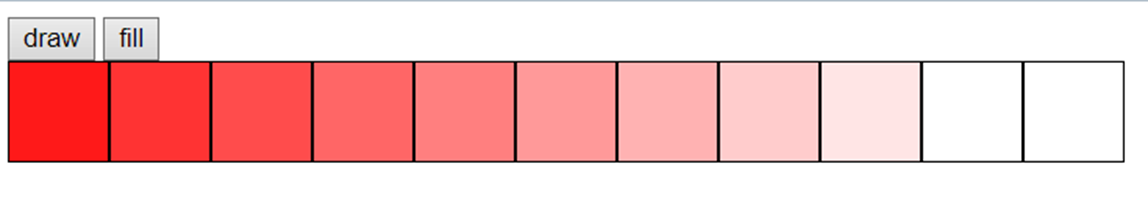
**Завдання 4**

Створіть клас **Shape**, який 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:***

****

Лістининг коду

HTML

<!DOCTYPE html>

<html lang="en">

    <head>

        <meta charset="UTF-8">

        <meta name="viewport" content="width=device-width, initial-scale=1.0">

        <title>Task 4</title>

        <link rel="stylesheet" href="style.css">

    </head>

    <body>

        <button onclick="drawShape()">Draw</button>

        <button onclick="fillPaint()">Fill</button>

        <div class="shape-container"></div>

        <script src="main.js"></script>

    </body>

</html>

CSS

.shape-container {

    display: flex;

}

.shape {

    width: 50px;

    height: 50px;

    border: 1px solid black;

    margin-block: 10px;

    transition: background-color 0.5s;

}

JS

class Shape {

    static total = 100;

    static fill() {

        this.total = 100;

        console.log('Paint filled to 100%');

    }

    static draw() {

        this.total -= 10;

    }

}

function drawShape() {

    const shapeElement = document.createElement('div');

    shapeElement.classList.add('shape');

    const transparency = Shape.total / 100;

    const color = `rgba(255, 0, 0, ${transparency})`;

    shapeElement.style.backgroundColor = color;

    Shape.draw()

    document.querySelector('.shape-container').appendChild(shapeElement);

}

function fillPaint() {

    Shape.fill();

}

Результат

