**Лабораторна робота №6**

Змн.

Арк.

№ докум.

Підпис

Дата

Арк.

1

*ДУ «Житомирська політехніка».23.121.03.00 – Лр6*

Розроб.

Білотіл Я. О.

Перевір.

Дмитренко І. А.

*Т. контр.*

Н. контр.

Затверд.

Звіт

з лабораторної роботи

Літ.

Аркушів

*19*

*ФІКТ Гр. IПЗк-23-1*

**Тема:** Події.

**Мета роботи:** вивчити засоби по роботі з об’єктною моделлю документа; навчитися додавати, змінювати та видаляти елементи веб-сторінки; засвоїти принципи роботи з CSS-стилями.

**Завдання на лабораторну роботу**

**Завдання 0: Ознайомитись з теоретичними відомостями за посиланням:**

[Вступ до подій](https://uk.javascript.info/events)

[Події інтерфейсу](https://uk.javascript.info/event-details)

[Форми та інтерактивні елементи](https://uk.javascript.info/forms-controls)

**Завдання 1.**

Дана таблиця з користувачами: номер, ім'я та прізвище. Під таблицею зробіть форму, з допомогою якої можна буде додати нового користувача в таблицю.

Код програми:

index.html

<!DOCTYPE html>

<html>

<head>

<title>Таблиця користувачів</title>

</head>

<body>

<h2>Таблиця користувачів</h2>

<table id="usersTable">

<tr>

<th>Номер</th>

<th>Ім'я</th>

<th>Прізвище</th>

</tr>

</table>

<h2>Додати нового користувача</h2>

<form id="addUserForm">

<label for="userName">Ім'я:</label>

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

<label for="userSurname">Прізвище:</label>

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

<input type="submit" value="Додати користувача">

</form>

</body>

</html>

script.js

var userNumber = 1;

document.getElementById("addUserForm").addEventListener("submit", function (event) {

event.preventDefault();

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

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

var table = document.getElementById("usersTable");

var row = table.insertRow(-1);

var cell1 = row.insertCell(0);

var cell2 = row.insertCell(1);

var cell3 = row.insertCell(2);

cell1.innerHTML = userNumber;

cell2.innerHTML = userName;

cell3.innerHTML = userSurname;

userNumber++;

// Очищаємо поля форми

document.getElementById("userName").value = "";

document.getElementById("userSurname").value = "";

});

Результат роботи програми:

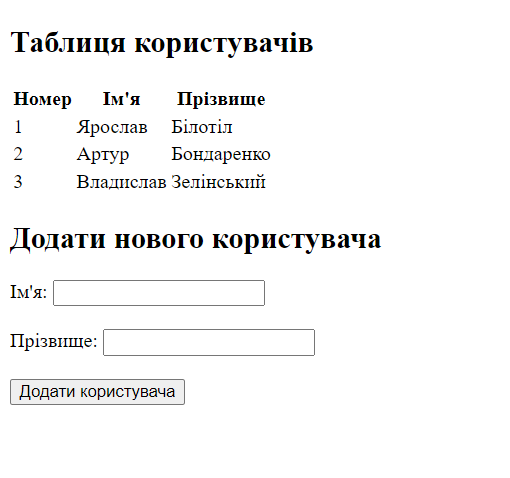


Рис. 1 – Завдання 1

**Завдання 2.**

В елемент textarea вводяться нулі та одиниці і з них потрібно сформувати квадратики чорного і білого кольору.

Код програми:

index.html

<!DOCTYPE html>

<html>

<head>

<title>Генерація квадратиків</title>

</head>

<body>

<h2>Введіть 0 та 1:</h2>

<textarea id="binaryInput" rows="4" cols="50"></textarea>

<button onclick="generateSquares()">Генерувати квадратики</button>

<div id="squaresContainer"></div>

</body>

</html>

style.css

.square {

width: 20px;

height: 20px;

display: inline-block;

border: 1px solid #000;

}

script.js

function generateSquares() {

const inputText = document.getElementById('binaryInput').value.trim(); // Отримуємо введені дані без пробілів

const lines = inputText.split('\n'); // Розділяємо введені рядки

const squaresContainer = document.getElementById('squaresContainer');

squaresContainer.innerHTML = ''; // Очищаємо контейнер перед новими квадратиками

lines.forEach(line => {

const row = document.createElement('div'); // Створюємо новий рядок

line.split('').forEach(char => {

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

square.classList.add('square');

square.style.backgroundColor = char === '0' ? 'black' : 'white';

row.appendChild(square);

});

squaresContainer.appendChild(row);

});

}

Результат роботи програми:

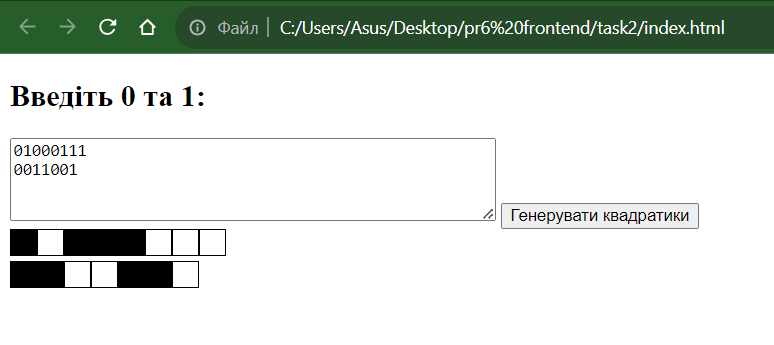


Рис. 2 – Завдання 2

**Завдання 3.**

**Вибрати потрібні кольори з палітри кольорів та залити ними даний прямокутник (блок).**

Кольори в палітрі повинні вибиратись клацанням. Повторне клацання повинно відміняти вибір. Заливка прямокутника відбувається одночасно з вибором кольорів. Якщо вибрано декілька кольорів, то заливка градієнтна.

Код програми:

index.html

<!DOCTYPE html>

<html>

<head>

<title>Палітра кольорів і заливка прямокутника</title>

</head>

<body>

<h2>Виберіть кольори:</h2>

<div id="colors">

<div class="color" style="background-color: #660000;"></div>

<div class="color" style="background-color: #ff0000;"></div>

<div class="color" style="background-color: #ff6600;"></div>

<div class="color" style="background-color: #ffff00;"></div>

<div class="color" style="background-color: #66cc00;"></div>

<div class="color" style="background-color: #009900;"></div>

<div class="color" style="background-color: #0099cc;"></div>

<div class="color" style="background-color: #003399;"></div>

<div class="color" style="background-color: #000066;"></div>

</div>

<div class="rectangle"></div>

</body>

</html>

style.css

.color {

width: 30px;

height: 30px;

display: inline-block;

border: 1px solid #000;

cursor: pointer;

}

.rectangle {

width: 300px;

height: 150px;

margin-top: 20px;

border: 1px solid #000;

background: linear-gradient(to right, #000, #fff);

}

script.js

const colors = document.querySelectorAll('.color');

const rectangle = document.querySelector('.rectangle');

let selectedColors = [];

colors.forEach(color => {

color.addEventListener('click', () => {

const clickedColor = color.style.backgroundColor;

const index = selectedColors.indexOf(clickedColor);

if (index === -1) {

selectedColors.push(clickedColor);

} else {

selectedColors.splice(index, 1);

}

if (selectedColors.length === 0) {

rectangle.style.background = 'linear-gradient(to right, #000, #fff)';

} else if (selectedColors.length === 1) {

rectangle.style.backgroundColor = selectedColors[0];

} else {

rectangle.style.background = `linear-gradient(to right, ${selectedColors.join()}, #fff)`;

}

});

});

Результат роботи програми:



Рис. 3 – Завдання 3

**Завдання 4.**

**Є блоки з текстом. В кожному з блоків є кнопки *Видалити* та *Редагувати***

При натисканні на кнопку *Видалити*, даний блок зникає. При натисканні на *Редагувати* блок переходить в стан редагування (наприклад, з допомогою *textarea*). При знятті фокуса з *textarea*, блок повертається в звичайний стан.

Код програми:

index.html

<!DOCTYPE html>

<html>

<head>

<title>Блоки з можливістю редагування та видалення</title>

</head>

<body>

<div class="blocks">

<div class="block">

<p>Lorem ipsum dolor sit amet, consectetur adipisicing elit.</p>

<button class="editButton">Редагувати</button>

<button class="deleteButton">Видалити</button>

</div>

<div class="block">

<p>Lorem ipsum dolor sit amet, consectetur adipisicing elit.</p>

<button class="editButton">Редагувати</button>

<button class="deleteButton">Видалити</button>

</div>

<div class="block">

<p>Lorem ipsum dolor sit amet, consectetur adipisicing elit.</p>

<button class="editButton">Редагувати</button>

<button class="deleteButton">Видалити</button>

</div>

</div>

</body>

</html>

style.css

.block {

border: 1px solid #000;

margin: 10px;

padding: 10px;

width: 200px;

}

.editMode {

border: 1px solid blue;

}

script.js

const blocks = document.querySelectorAll('.block');

blocks.forEach(block => {

const editButton = block.querySelector('.editButton');

const deleteButton = block.querySelector('.deleteButton');

const paragraph = block.querySelector('p');

let isEditMode = false;

editButton.addEventListener('click', () => {

if (isEditMode) return;

isEditMode = true;

block.classList.add('editMode');

const textarea = document.createElement('textarea');

textarea.value = paragraph.textContent;

block.replaceChild(textarea, paragraph);

textarea.focus();

textarea.addEventListener('blur', () => {

isEditMode = false;

block.classList.remove('editMode');

const newParagraph = document.createElement('p');

newParagraph.textContent = textarea.value;

block.replaceChild(newParagraph, textarea);

});

});

deleteButton.addEventListener('click', () => {

block.remove();

});

});

Результат роботи програми:

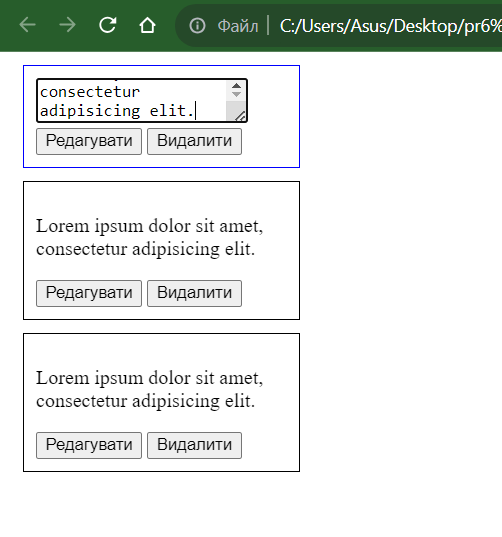


Рис. 4 – Завдання 4

**Завдання 5.**

**Задати можливість заливати квадратики вибраним кольором.**

Є палітра кольорів, в якій клацанням можна вибрати лише один колір. І є прямокутне полотно (контейнер, блок), яке заповнене маленькими блоками-квадратиками. При клацанні курсором на ці квадратики, вони повинні заливатись вибраним кольором.

Код програми:

index.html

<!DOCTYPE html>

<html>

<head>

<title>Заливання квадратиків кольором</title>

</head>

<body>

<div class="palette">

<div class="color" style="background-color: #660000;"></div>

<div class="color" style="background-color: #ff0000;"></div>

<div class="color" style="background-color: #ff6600;"></div>

<div class="color" style="background-color: #ffff00;"></div>

<div class="color" style="background-color: #66cc00;"></div>

<div class="color" style="background-color: #009900;"></div>

<div class="color" style="background-color: #0099cc;"></div>

<div class="color" style="background-color: #003399;"></div>

<div class="color" style="background-color: #000066;"></div>

</div>

<div class="canvas">

</div>

</body>

</html>

style.css

.palette {

display: grid;

grid-template-columns: repeat(3, 30px);

gap: 5px;

}

.color {

width: 30px;

height: 30px;

border: 1px solid #000;

cursor: pointer;

}

.canvas {

display: grid;

grid-template-columns: repeat(10, 30px);

gap: 2px;

margin-top: 20px;

}

.square {

width: 30px;

height: 30px;

border: 1px solid #ccc;

cursor: pointer;

}

script.js

const canvas = document.querySelector('.canvas');

for (let i = 0; i < 100; i++) {

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

square.classList.add('square');

canvas.appendChild(square);

}

const squares = document.querySelectorAll('.square');

let selectedColor = null;

squares.forEach(square => {

square.addEventListener('click', () => {

if (selectedColor) {

square.style.backgroundColor = selectedColor;

}

});

});

const colors = document.querySelectorAll('.color');

colors.forEach(color => {

color.addEventListener('click', () => {

selectedColor = color.style.backgroundColor;

});

});

Результат роботи програми:

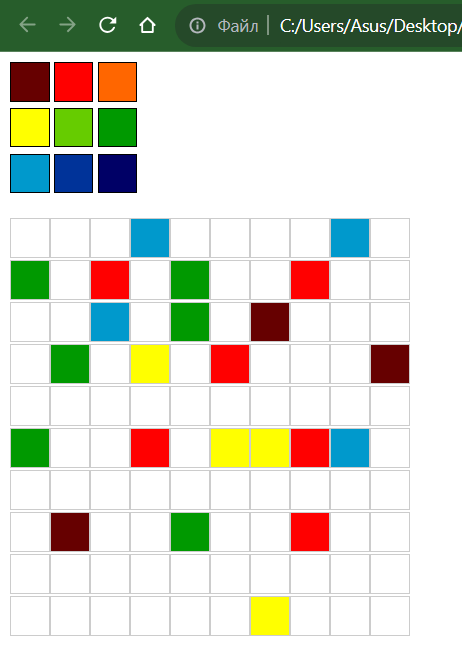


Рис. 5 – Завдання 5

**Завдання 6.**

Є ряд гральних карт, повернутих до нас «сорочкою». При клацанні на карту, вона повертається до нас  лицьовим боком протягом 1-2 секунд (повинен бути саме поворот). При повторному клацанні, карта знову повертається «сорочкою».

Код програми:

index.html

<!DOCTYPE html>

<html>

<head>

<title>Гральні карти</title>

<style>

.card {

width: 200px;

height: 300px;

background: url('back\_of\_card.jpg') center/cover;

cursor: pointer;

transition: transform 1s;

}

</style>

</head>

<body>

<div class="card" onclick="flipCard(this)"></div>

</body>

</html>

style.css

.card {

width: 200px;

height: 300px;

background: url('back\_of\_card.jpg') center/cover;

cursor: pointer;

transition: transform 1s;

}

script.js

let isFlipped = false;

let isFlipping = false;

function flipCard(card) {

if (!isFlipping) {

isFlipping = true;

if (!isFlipped) {

card.style.transform = "rotateY(90deg)";

setTimeout(() => {

card.style.backgroundImage = "url('front\_of\_card.jpg')";

card.style.transform = "rotateY(180deg)";

isFlipped = true;

isFlipping = false;

}, 1000);

} else {

card.style.transform = "rotateY(90deg)";

setTimeout(() => {

card.style.backgroundImage = "url('back\_of\_card.jpg')";

card.style.transform = "rotateY(0)";

isFlipped = false;

isFlipping = false;

}, 1000);

}

}

}

Результат роботи програми:

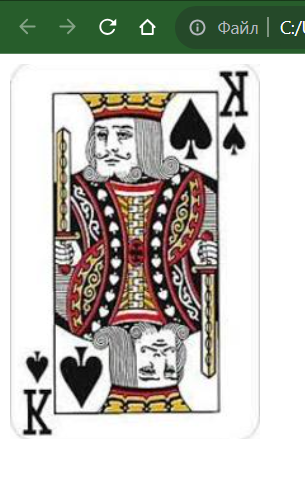


Рис. 6 – Завдання 6

**Завдання 7.**

Створити html-сторінку зі списком повідомлень на форумі та формою для додавання нового повідомлення. Після заповнення форми додати повідомлення до списку на екрані.

Код програми:

index.html

<!DOCTYPE html>

<html>

<head>

<title>Форум</title>

</head>

<body>

<h1>Форум</h1>

<table id="messageList">

<thead>

<tr>

<th>Ім'я</th>

<th>Повідомлення</th>

<th>Час надсилання</th>

</tr>

</thead>

<tbody>

</tbody>

</table>

<h2>Додати нове повідомлення</h2>

<form id="messageForm">

<label for="username">Ім'я:</label>

<input type="text" id="username" name="username" required><br><br>

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

<textarea id="message" name="message" rows="4" cols="50" required></textarea><br><br>

<input type="submit" value="Додати повідомлення">

</form>

</body>

</html>

script.js

const messageForm = document.getElementById('messageForm');

const messageList = document.querySelector('#messageList tbody');

messageForm.addEventListener('submit', function (event) {

event.preventDefault();

const username = messageForm.elements['username'].value;

const message = messageForm.elements['message'].value;

if (username && message) {

const now = new Date();

const timestamp = now.toLocaleString();

const newRow = document.createElement('tr');

newRow.innerHTML = `

<td>${username}</td>

<td>${message}</td>

<td>${timestamp}</td>

`;

messageList.appendChild(newRow);

messageForm.reset();

} else {

alert("Будь ласка, заповніть усі поля");

}

});

Результат роботи програми:

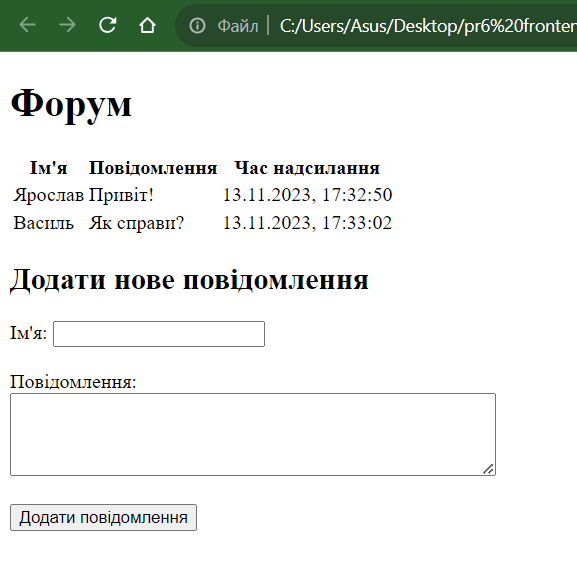


Рис. 7 – Завдання 7

**Завдання 8.**

Створити html-сторінку зі списком книг. При натисканні на книгу, колір фону повинен змінюватися на зелений. Врахуйте, що при повторному натисканні на іншу книгу, попередньої – необхідно повертати колишній колір.

Код програми:

index.html

<!DOCTYPE html>

<html>

<head>

<title>Список книг</title>

</head>

<body>

<h1>Список книг</h1>

<ul>

<li id="book1">«Володар кілець» Джон Р. Р. Толкін</li>

<li id="book2">«Гордість і упередження» Джейн Остін</li>

<li id="book3">«Темні початки» Філіп Пулман</li>

<li id="book4">«Автостопом по галактиці» Дуглас Адамс</li>

<li id="book5">«Гаррі Поттер і Кубок вогню» Джоан Роулінг</li>

<li id="book6">«Вбити пересмішника» Харпер Лі</li>

</ul>

</body>

</html>

style.css

ul {

list-style: none;

padding: 0;

}

li {

padding: 10px;

margin: 5px;

cursor: pointer;

}

li:hover {

background-color: #f0f0f0;

}

.green {

background-color: green;

}

script.js

const books = document.querySelectorAll('li');

books.forEach(book => {

book.addEventListener('click', () => {

if (book.classList.contains('green')) {

book.classList.remove('green');

} else {

books.forEach(otherBook => {

otherBook.classList.remove('green');

});

book.classList.add('green');

}

});

});

Результат роботи програми:

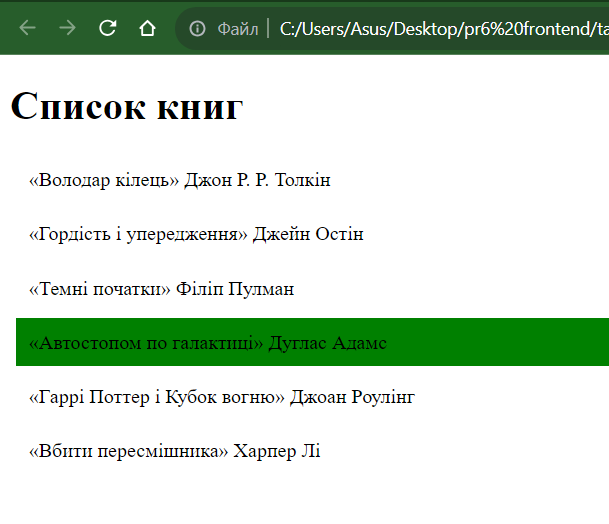


Рис. 8 – Завдання 8