**ЛАБОРАТОРНА РОБОТА №4**

*Об’єктна модель документа DOM*

***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.

**Порядок виконання роботи**

1. Ознайомитися із теоретичними відомостями.
2. Виконати завдання до лабораторної роботи.
3. Скласти та оформити звіт.

**Теоретичні відомості**

[DOM дерево](https://uk.javascript.info/dom-nodes)

[Навігація по DOM](https://uk.javascript.info/dom-navigation)

[Пошук: getElement\*, querySelector\*](https://uk.javascript.info/searching-elements-dom)

[Властивості вузлів: тип, тег та вміст](https://uk.javascript.info/basic-dom-node-properties)

[Атрибути та властивості](https://uk.javascript.info/dom-attributes-and-properties)

[Внесення змін в документ](https://uk.javascript.info/modifying-document)

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

**Завдання до роботи**

Завдання до лабораторної роботи складається із трьох частин:

Частина 1. Написання функцій для опрацювання дати та часу.

Частина 2. Написання функцій із регулярними виразами.

Частина 3. Робота із об’єктною моделлю документа.

## Частина 1: Написання функцій для опрацювання дати та часу

**Завдання 1.** Написати функцію, яка виводить поточну дату у вигляді:

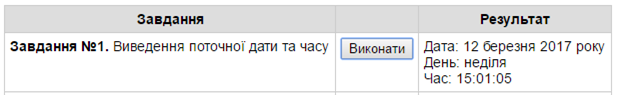
**Дата: 12 березня 2017 року**

**День тижня: неділя**

**Час: 15:08**

Реалізувати запуск цієї функції при натисканні на відповідну кнопку.

Результат работи має виглядати так:



Для того, щоб запустити функцію при натисканні на кнопку використовуйте встановлення обробника події click:

<**button onclick="***функція*()**"**>Виконати</**button**>

В функції можна звертатися до тегів HTML-документа і записувати в них контент. Наприклад, щоб записати рядок ***s*** у якості контента тегу, який має **id="block"**,потрібно:

**let** s = **'Текст. Тут можна використовувати <em>HTML</em>-теги'**;

**let** elem = **document**.getElementById(**'currentDate'**);

elem.**innerHTML** = s;

**Завдання 2.** Користувач вводить дату у форматі (дд.мм.рррр або дд/мм/рррр або дд-мм-рррр). Опублікувати цю дату у наступному вигляді:

* Якщо дата сьогоднішня - вивести **Сьогодні**
* Якщо вчорашня - **Вчора**,
* Якщо пройшло 2-6 днів, вивести: **2 дні тому, 3 дні тому,… 6 днів тому**
* Якщо пройшло 7 днів - **Тиждень тому**
* Якщо пройшов рік – **Рік тому**
* В інших випадках – вивести дату у форматі **РРРР.ММ.ДД**

## Частина 2: Написання функцій із регулярними виразами

**Завдання 1.** Напишіть функцію findDates (text), яка знаходить у тексті всі дати, записані у форматі РРРР-ММ-ДД. Дати мають бути валідними. Використати **регулярні вирази**.

**Завдання 2.** Напишіть функцію toCamelCase(), яка приймає значення імені змінної в стилі “snake\_case”, та повертає це ж ім’я в стилі «lowerCamelCase» Використати **регулярні вирази**. Для прикладу:

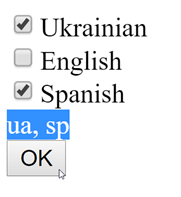
|  |  |
| --- | --- |
| Значення, що передається в функцію | Результат роботи функції |
| my\_variable | myVariable |
| new\_brand\_product | newBrandProduct |

**Завдання 3.** Напишіть функцію toSnakeCase(), яка працює як обернена до функції toCamelCase(). Використати **регулярні вирази**.

**Завдання 4.** Напишіть функцію findHexColor (text), яка знаходить у заданому тексті колір у форматі #ABC або # ABCDEF. Якщо не знайдено, повернути null. Використати **регулярні вирази**.

## Частина 3: Об’єктна модель документа DOM

**Завдання 1.** Дано чекбокси і кнопка. По натисканню на кнопку вивести в рядок значення вибраних елементів

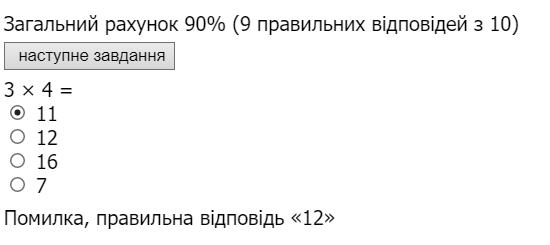


**Завдання 2.** На сторінці розміщені електронні адреси з прапорцями (чекбоксами). Реалізуйте можливість формування рядку з вибраними електронними адресами. При позначенні електронної адреси, вона додається в div-елемент. При знятті чекбокса, адреса видаляється з елементу



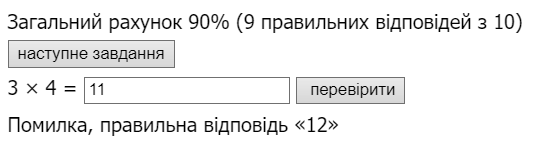
**Завдання 3.** Створити JavaScript, який перевіряє знання таблиці множення. Веб-сторінка текстовий напис для показу загального рахунку, кнопку «наступне завдання», текстовий напис для показу завдання, радіокнопки для вибору правильного варіанту відповіді, та текстовий напис для виводу результатів перевірки. Вибір варіанту відповіді є сигналом до початку перевірки. Для кожного завдання користувач має лише одна спробу вибору.

**Зовнішній вигляд веб-сторінки:**

****

**Завдання 4.** Створити JavaScript, який перевіряє знання таблиці множення. Веб-сторінка містить текстовий напис для показу загального рахунку, кнопку «наступне завдання», текстовий напис для показу завдання, текстове поле для вводу відповіді, кнопку «перевірити» та текстовий напис для виводу результатів перевірки.

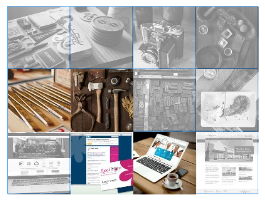
**Зовнішній вигляд веб-сторінки:**

****

**Завдання 5.** Створіть галерею зображень, кожне з яких неактивне. При клацанні по зображенню, воно стає активним. При наступному клацанні по ньому, воно знову стає неактивним. Для двох станів задайте наступні стилі:

o   *Активний стан:* фільтр *відтінки сірого(0%)*; непрозорість *1.*

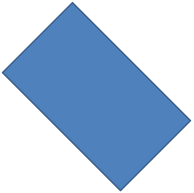
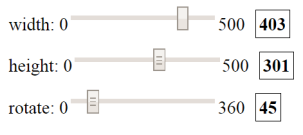
o   *Неактивний стан:* фільтр *відтінки сірого(100%)*; непрозорість *0.5.*

**

**Завдання 6.** Реалізуйте елементи форми. При фокусуванні на елемент, секція, в якому він знаходиться, стає активною (синій колір).

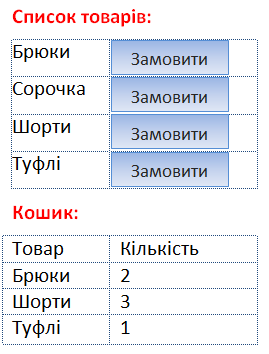


**Завдання 7.** Реалізувати можливість змінювати розмір блока, а також його кут повороту:



**Завдання 8.** Дана картинка в тегу img та кнопка. Зробіть кнопку, за натисканням на яку вперше ширина і висота картинки буде збільшуватися в 2 рази, вдруге – приходити у вихідний стан

**Завдання 9.** Є список товарів з кнопками *Замовити*. При натисканні на неї, одиниця даного товару заноситься в кошик.

****

**Завдання 10.** Реалізувати в контейнері можливість переглядати світлини одна за одною при клацанні курсора. Світлини змінюють одна одну з ефектом слайдингу.