**Практична робота 2**

**Тема:** Підключення зовнішніх скриптів JavaScript та послідовність їх виконання в HTML-документі.

**Мета:** Навчитися підключати зовнішні скрипти JavaScript. Ознайомитися з поняттям асинхронних скриптів JavaScript. Навчитися використовувати атрибути defer/async. Вміти використовувати спливаючі діалогові вікна введення/виведення та виведення даних у консоль браузера.

**Виконав:** Братчиков Роман Русланович

**Група:** 244а

**Хід роботи**

* 1. **Створення html файлу**

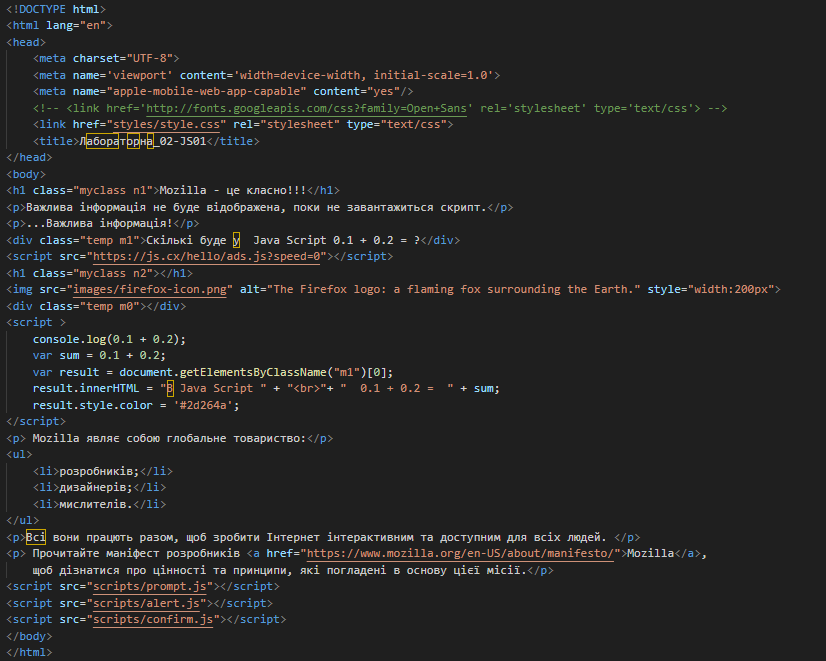


Рис 1 Розмітка вебсторінки

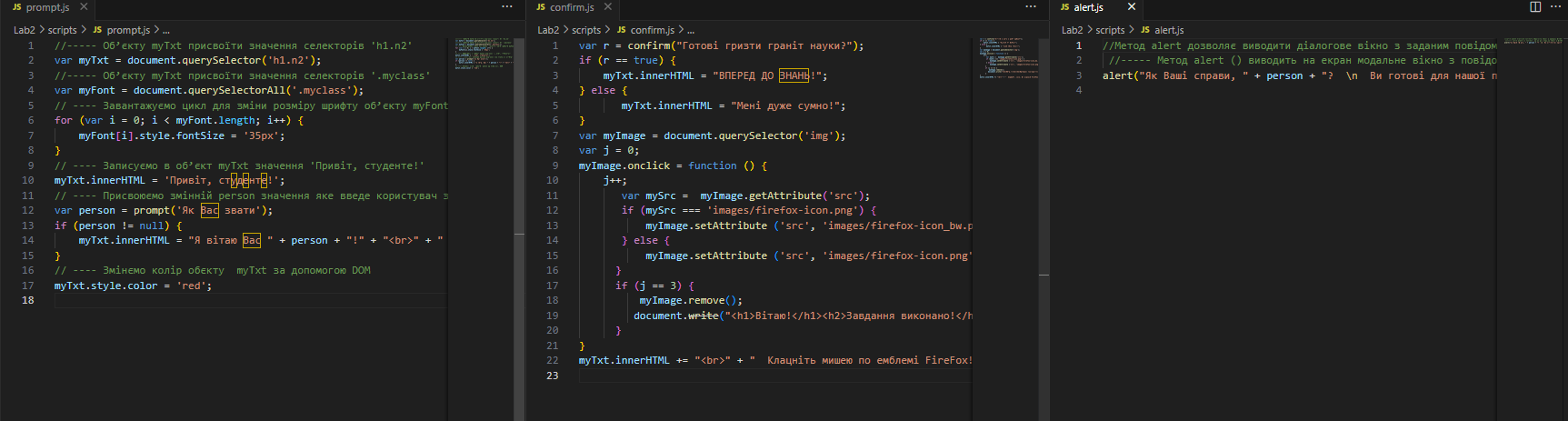
**1.2 Додаємо скрипти в папку Scripts  
**

Рис 2 Скрипти для сторінки, які викликаються через тег <script>

Ці скрипти виконують наступні виклики діалогових вікон:

**alert.js:**

Метод alert дозволяє виводити діалогове вікно з заданим повідомленням і кнопкою Оk.

**confirm.js:**

Метод confirm дозволяє вивести діалогове вікно з повідомленням і двома кнопками – Оk і Відміна (Cancel). На відміну від методу alert цей метод повертає логічну величину, значення якої залежить від того, на якій з двох кнопок клацнув користувач. Якщо він клацнув на кнопці Оk, то повертається значення true (істина, так); якщо він клацнув на кнопці Скасування, то повертається значення false:

**prompt.js**

Метод prompt дозволяэ вивести діалогове вікно з полем вводу, де користувач може ввести інформацію, яка приймається як рядок та оброблюється відповідно заданого сценарію.

Послідовність виконання скріптів аналогічна до порядку введеного в розмітці, тобто першим

іде prompt, другим alert і третім confirm.(Рис.3)

**2. Асинхронні скрипти defer/async**

При додаванні нового скрипту змінився формат подачі інформації, наприклад до додавання скрипту сторінка відображала інформацію до скрипту prompt, а при новому скрипті завантження показується тільки після запуску скрипту alert.(рис.4)

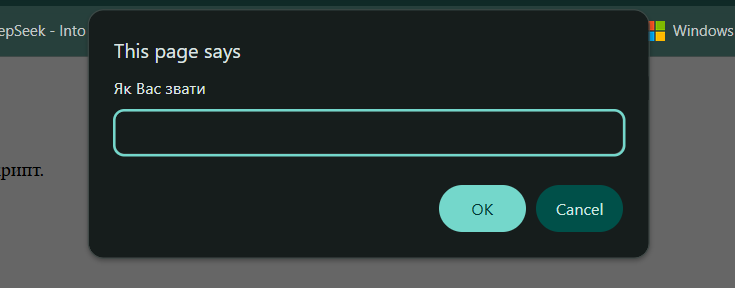
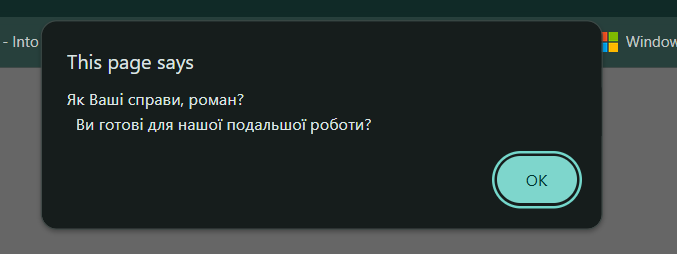
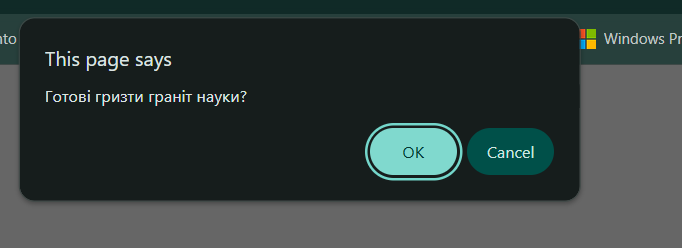
  

Рис 3 Послідовність виводу діалогових вікон

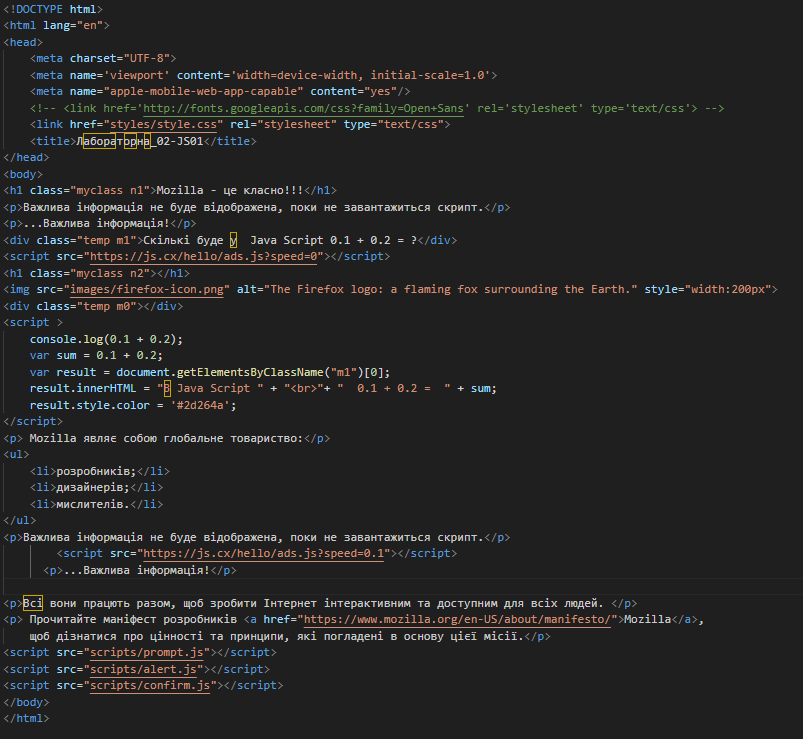


Рис 4 Додавання нового скрипту

**2.1 Зміна скриптів на async**

Після зміни потоку скріптів, всі вони активізувались до prompt, що тільки підтверджує факт того, що асинхронні скрипти запускаються і відбуваються одночасно, і ідуть не по порядку, а одночасно, але скрипти, які в своїй логіці очікують інпуту, то вони чекають свого виконання, а не запускаються одночасно(Рис.5)

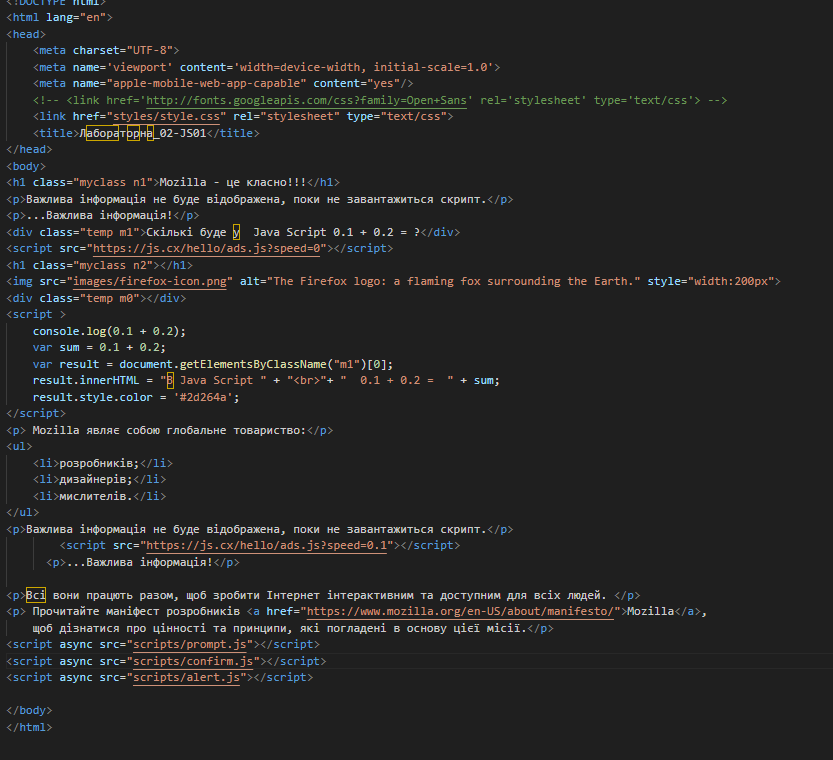


Рис 5 зміна місць скриптів та додавання асинхронності

**3. Виведення інформації засобами JavaScript**

Існують чотири способи виведення інформації засобами JavaScript:

1. За допомогою методу innerHTML :

<**div class="temp m1"**>…/**div**>

<**script**>  
 **var *sum*** = 0.1+0.2;  
 **var *result*** = **document**.getElementsByClassName(**"m1"**)[0];  
 ***result***.**innerHTML** = **"В Java Script "** + **"<br>"**+ **" 0.1 + 0.2 = "** + ***sum***;  
 ***result***.style.**color** = **'#2d264a'**;  
</**script**>

1. Запис (перезапис) сторінки за допомогою методу document.write( ) :

**document.write(9 + 10);**

1. Виведення в модальному вікні за допомогою window.alert( ) :

**window**.alert(“Важлива інформація!”);

1. Виведення в консоль браузера за допомогою console.log( ):

**console**.log(20 + 17);

Всі 4 методи були використані в сторінці на наступному скріні:

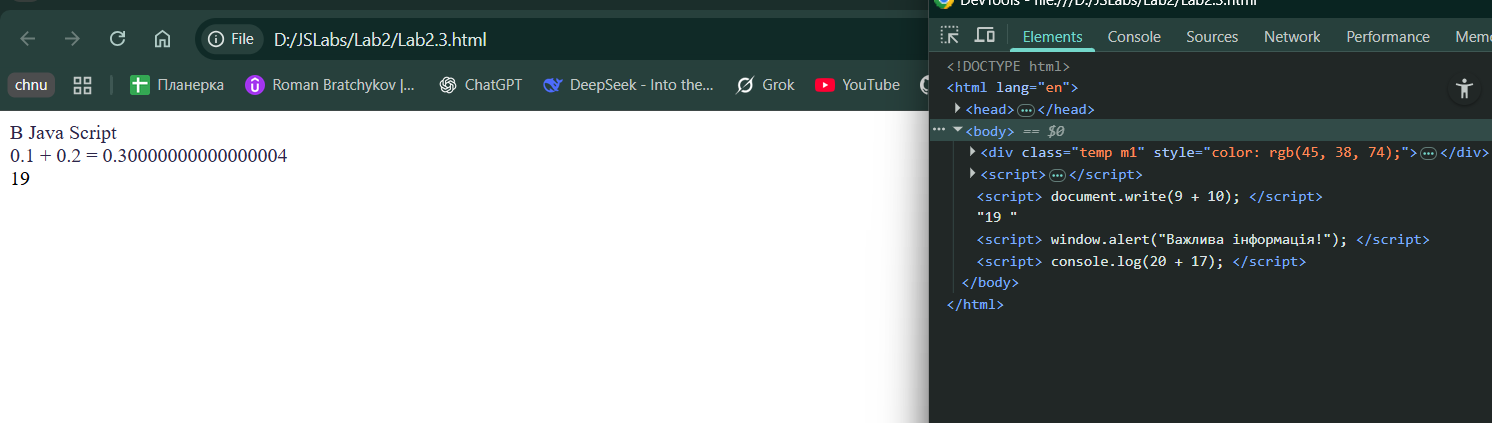


Рис 6 Приклад розмітки та додавання скриптів, теги html не були використані прямо для виводу

**4.Спливаючі вікна в JavaScript**

У JavaScript є три типи спливаючих вікон: вікно сповіщення, поле підтвердження та вікно запиту:

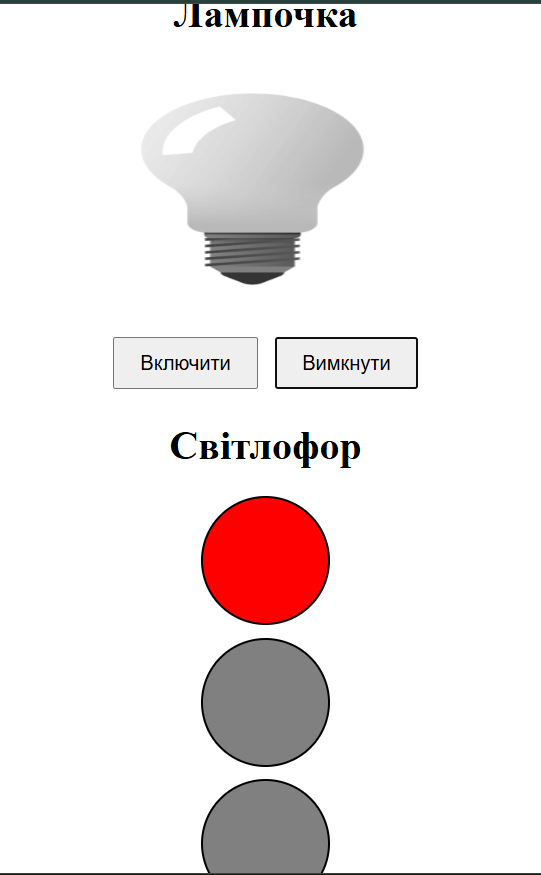
* вікно alert, для продовження роботи скрипта необхідно обов’язково натиснути кнопку Ok (alert("УВАГА!");)
* вікно підтвердження window.confirm( ), яке повертає два значення true або false;
* вікно запрошення prompt, яке повертає введене значення або null.

Всі три вікна були використані в першому завданні для введення інформації або булевих значень

1. **Створення html документу**

В файлі [testFile.html](https://github.com/RomanBratchykov/JSLabs/blob/main/Lab2/testPage.html) було створено два об’єкти які відповідають наступним завданням:

* 1. Створити HTML-документ на основі прикладу першого HTML-документу лабораторної роботи який буде містити дві кнопки та зображення лампочки. При натисненні кнопки включити – лампочка буде міняти колір з білого на жовтий, а при вимкнути – навпаки.
  2. Створити HTML-документ який буде симулятором роботи світлофору. Для зміни кольору застосуйте CSS та JavaScript. В результаті маємо таку вебсторінку:

**Висновок.**

В ході виконання лабораторної роботи було вивчено про роботу скриптів в вебсторінках, про їх підключення, про різні методи виводу та вводу. Також було пройдено тему асинхронних скриптів, та створено власну сторінку з використанням кнопок та скриптів