

# ЗАВДАННЯ ДО ЛАБОРАТОРНИХ РОБІТ

## Лабораторна робота № 2. Частина \_1

**Тема:** Створення HTML – документу, що містить сценарій JavaScript засобами редактора Notepad++.

**Мета:** Повторення HTML. Навчитись використовувати редактор Notepad++ для створення сценаріїв JavaScript в HTML – документах.

**Порядок виконання.**

1. Провести ознайомлення з редактором – Notepad++ як засоба створення сценаріїв JavaScript в HTML – документах. В звіті описати засоби створення, редагування і перегляду редактора.
2. Створити HTML – документи, які містить наступний код:

Анн	<pre>&lt;html&gt; &lt;head&gt;   &lt;title&gt; Script_01 &lt;/title&gt; &lt;/head&gt; &lt;body&gt; &lt;script language="JavaScript"&gt;   var v_1 = 15, u_1 = "2", c;   c = v_1 + u_1;   c = c - v_1;   document.writeln("c = " + c); &lt;/script&gt; &lt;/body&gt; &lt;/html&gt;</pre>
Б	<pre>&lt;html&gt; &lt;script language "JavaScript"&gt; theTime = new Date(); theHour = theTime.getHours(); if (11 &gt; theHour)   document.writeln("&lt;body bgcolor='red' text='Black'&gt;"); else   document.writeln("&lt;body bgcolor='green' text='White'&gt;"); &lt;/script&gt; &lt;/body&gt; &lt;/html&gt;</pre>

3. Провести доповнення та редагування наведених документів з метою ознайомлення з структурою та перетворенням змінних. Отримані документи зберігти та переглянути браузером.
4. У звіті представити код створеної сторінки як додаток №1.
5. У звіті представити вигляд документу браузером як додаток №2.

## Лабораторна робота № 2. Частина \_2

**Тема:** Ознайомлення з організацією циклів в межах сценаріїв JavaScript.

**Мета:** Навчитись використовувати оператори `for`, `while`, `break`, `continue` при організації сценаріїв JavaScript в HTML – документах.

**Теоретичні відомості:** *Цикл* це послідовність операторів, виконання якої повторюється до тих пір, поки певна умова не стане помилковою. JavaScript містить оператори циклу :**for**, **while** , а також оператори **break** і **continue**, які використовуються усередині циклів.

**Оператор циклу for** має вигляд:

**for** (*ініціалізація; умова; зміна*) *оператор*

Тут *ініціалізація* і *зміна* — це будь-яке вирази, *умова* будь-який вираз, значення якого може бути перетворене до логічного типу, *оператор* — будь-яка група операторів JavaScript; якщо ці група містить більше ніж один оператор, то вона повинні бути поміщена у фігурні дужки { ...statements... }. *Ініціалізація* може містити декларацію змінної. Даний оператор зазвичай використовується в тих випадках, коли кількість повторень циклу відома заздалегідь.

Оператор **for** виконується таким чином:

1. Виконується вираз *ініціалізація* (цей вираз ініціалізує лічильник циклу).
2. Обчислюється значення виразу *умова*. Якщо воно помилкове, то управління передається операторові, наступному за даним оператором.
3. Виконується *оператор*.
4. Виконується вираз *зміна* (звичайний цей вираз збільшує або зменшує лічильник циклу) і управління передається етапу 2.

**Оператор циклу while** має вигляд:

**while** (*умова*) *оператор*

Тут *умова* будь-який вираз, значення якого може бути перетворене до логічного типу, *оператор* - будь-яка група операторів JavaScript; якщо ці група містить більше ніж один оператор, то вона повинні бути поміщена у фігурні дужки

{ ...statements... }.

Оператор **while** виконується таким чином:

1. Обчислюється значення виразу *умова*. Якщо воно помилкове, то управління передається операторові, наступному за даним оператором.
2. Виконується *оператор* і управління передається етапу 1.

При використанні даного оператора необхідно переконатися, що рано чи пізно *умова* стане помилковою, інакше сценарій увійде до нескінченного циклу

### **Оператор continue**

Передає управління операторові перевірки істинності умови в циклі *while* і операторові оновлення значення лічильника в циклі *for*. Оператор *continue* не перериває цикл, а робить наступне: У циклі *while* переходить назад до умови; У циклі *for* переходить до модернізованого виразу.

### **Оператор break**

Оператор перериває поточний цикл *for* і *while*, передаючи управління першому операторові після циклу.

#### **Порядок виконання.**

1. Базуючись на раніше створених HTML – документах створити JavaScript сценарій, що містить оператори циклу.
2. У звіті представити HTML код сторінки з формою як додаток №1.
3. У звіті представити вигляд документу браузером як додаток №2.
4. Підготувати звіт з описом проведеної роботи.

## Лабораторна робота № 2. Частина \_3

**Тема:** Ознайомлення з використанням умовного оператора і оператора вибору в межах сценаріїв JavaScript.

**Мета:** Навчитись використовувати оператори **if.else** та **switch** при організації сценаріїв JavaScript.

### Теоретичні відомості: **Умовний оператор**

Умовний оператор **if.else** дозволяє перевірити певну умову і, залежно від його істинності, виконати ту або іншу послідовність операторів. Він має дві форми:

```
if (умова) оператор1  
if (умова) оператор1 else оператор2
```

Тут *умова* це будь-який вираз, значення якого може бути перетворене до логічного типу, *оператор1* і *оператор2* будь-які групи операторів JavaScript; якщо ці групи містять більш за одного оператора, то вони повинні бути поміщені у фігурні дужки { ...statements... }.

Перша форма оператора означає, що якщо значення *умови* істинне, то виконуються *оператор1*; якщо воно помилкове, то управління передається операторові, наступному за **if**.

Друга форма оператора означає, що якщо значення *умови* істинне, то виконуються *оператор1*; якщо воно помилкове, то виконуються *оператор2*.

### **Оператор вибору**

Оператор вибору **switch** виконує ту або іншу послідовність операторів залежно від значення певного виразу. Він має вигляд:

```
switch (вираз) {  
    case значення:  
        оператори  
        break;  
    case значення:  
        оператори  
        break;  
    ...  
    default:  
        оператори  
}
```

Тут *вираз* це будь-який вираз, *значення* це можливе значення виразу, а *оператори* будь-які групи операторів JavaScript.

Оператор вибору спочатку обчислює значення *виразу*, а потім перевіряє, чи немає цього значення в одній з міток **case** *значення*. Якщо така мітка є, то виконуються *оператори*, наступні за нею; якщо немає, то виконуються *оператори*, наступні за міткою **default** (якщо вона відсутня, то управління передається операторові, наступному за **switch**).

Необов'язковий оператор **break** указує, що після виконання *операторів* управління передається операторові, наступному за **switch**. Якщо **break** відсутній, то після виконання *операторів* починають виконуватися *оператори*, що стоять після наступної мітки **case**.

#### Порядок виконання.

1. Базуючись на раніше створених HTML – документах створити JavaScript сценарій, що містить оператори **if.else** та **switch** .
2. У звіті представити HTML код сторінки з формою як додаток №1.
3. У звіті представити вигляд документу браузером як додаток №2.
4. Підготувати звіт з описом проведеної роботи.

## Лабораторна робота № 2. Частина \_4

**Тема:** Ознайомлення з об'єктною структурою в межах сценаріїв JavaScript.

**Мета:** Навчитись використовувати об'єкти браузера: **window**, **location**, **history**, **document**, **navigator** а також вбудованого об'єкта **Date** при організації сценаріїв JavaScript.

### Порядок виконання.

1. Базуючись на раніше створених HTML – документах створити JavaScript сценарій, що демонструє використання об'єкту **window**.

Наприклад:	<pre>&lt;html&gt; &lt;head&gt; &lt;title&gt;Test_04&lt;/title&gt; &lt;/head&gt; &lt;body&gt; &lt;script language="JavaScript"&gt; contentsWindow = window.open("http://www.lp.edu.ua/", "contents", "top=100", "left=200", "width=400", height="200", "toolbar=no", "status=no"); &lt;/script&gt; &lt;/body&gt; &lt;/html&gt;</pre>
------------	---

2. Додатково ознайомитись з методами і властивостями об'єктів: **location**, **history**, **document**, **navigator** і використати деякі з них при організації сценаріїв JavaScript.
3. Створити JavaScript сценарій, що демонструє використання об'єкту **Date**.

Наприклад:	<pre>&lt;html&gt; &lt;head&gt; &lt;title&gt;Test_05&lt;/title&gt; &lt;/head&gt; &lt;body&gt; &lt;script language="JavaScript"&gt; now= new Date(); document.write("Time: " + now.getHours() + ":" + now.getMinutes() + "&lt;br&gt;"); document.write("Date: " + (now.getMonth() + 1) + "/" + now.getDate() + "/" + (1900 + now.getFullYear())); &lt;/script&gt; &lt;/body&gt; &lt;/html&gt;</pre>
------------	---

4. У звіті представити HTML код сторінки з формою як додаток №1.
5. У звіті представити вигляд документу браузером як додаток №2.
6. Підготувати звіт з описом проведеної роботи.

## Лабораторна робота № 2. Частина \_5

**Тема:** Ознайомлення з організацією сценаріїв JavaScript. Табулювання функцій.

**Мета:** Складання найпростіших алгоритмів при організації сценаріїв JavaScript.

### Порядок виконання.

1. Створити JavaScript сценарій, що демонструє використання таблиці.

Приклад сценарію, що буде інформаційну таблицю.	<pre>&lt;html&gt;&lt;head&gt;&lt;title&gt;Броузер&lt;/title&gt;&lt;/head&gt; &lt;body bgcolor="#fff0ff"&gt; &lt;center&gt;&lt;table border=1&gt; &lt;tr&gt;&lt;td&gt;Ім'я браузера&lt;/td&gt;&lt;td&gt; &lt;script language="JavaScript"&gt; document.write(navigator.appName);&lt;/script&gt;&lt;/td&gt; &lt;tr&gt;&lt;td&gt;Версія браузера&lt;/td&gt;&lt;td&gt; &lt;script language="JavaScript"&gt; document.write(navigator.appVersion);&lt;/script&gt;&lt;/td&gt; &lt;tr&gt;&lt;td&gt;Кодова назва браузера&lt;/td&gt;&lt;td&gt; &lt;script language="JavaScript"&gt; document.write(navigator.appCodeName);&lt;/script&gt;&lt;/td&gt; &lt;tr&gt;&lt;td&gt;Заголовок користувачького &lt;br&gt;агента&lt;/td&gt;&lt;td valign=top&gt; &lt;script language="JavaScript"&gt; document.write(navigator.userAgent);&lt;/script&gt;&lt;/td&gt; &lt;/table&gt;&lt;center&gt;&lt;/body&gt;&lt;/html&gt;</pre>
---	--

2. Ознайомитись з методами і властивостями об'єкта **Math** і використати деякі з них при організації сценаріїв JavaScript.
3. Створити JavaScript сценарій, що демонструє використання об'єкта **Math**.

**Постановка задачі табулювання функції.** За даними дійсними значеннями  $a, b$ , цілим  $n$  знайти значення функції  $f(x_i)$ , де

$$x_i = a + ih, i = \overline{0, n}, h = \frac{b-a}{n}. \text{ Одержані значення } x_i, f(x_i)$$

надрукувати у вигляді таблиці. Визначити найбільше та найменше значення функції на проміжку табуляції, а також відповідні значення аргументів.

4. У звіті представити HTML код сторінки з формою як додаток №1.
5. У звіті представити вигляд документу браузером як додаток №2.
6. Підготувати звіт з описом проведеної роботи.

### Варіанти завдань

№	Функція $f(x_i)$	a	b	n
1.	$e^{2x-\sin^2 x}$	$-\pi$	$\pi$	9
2.	$x^2 + \operatorname{arctg} x^2$	1	3	10
3.	$(x - \operatorname{tg} x)\sqrt{x}$	0	1	9
4.	$\frac{x^2}{(1 +  \sin x )}$	$-\pi$	$\pi$	10
5.	$\sqrt{2.5 +  \cos x } x^2$	1	3	8
6.	$\sqrt{x}  \cos x  + \sin x$	1	$3\pi$	9
7.	$\sqrt{x^2 - 1} + \ln(2x + 3)$	1	3	10
8.	$\arcsin(2x - 1) - \ln 1 - x^2 $	0.2	0.6	10
9.	$\arcsin(x + 2) + x^2$	-2.5	-1	10
10.	$\cos \frac{\pi x}{3} + e^{1-\sin x}$	$-\pi$	0	8
11.	$\operatorname{tg} x + \operatorname{ctg}(1 - x^2)$	0	0.4	10
12.	$(x^3 - 3x + 2)\sqrt{2x^3 - 1}$	1	7	9
13.	$\ln(\pi +  \sin x^2 ) + x^4$	1.7	3.3	8
14.	$\arccos \frac{x-1}{\sqrt{15}} + \operatorname{arctg} x$	-1	1	11
15.	$2x^2 + \ln x$	1	2.5	12



## Лабораторна робота № 2. Частина . Частина \_6

**Тема:** Ознайомлення з організацією сценаріїв JavaScript для роботи з даними стрічкового типу .

**Мета:** Набуття практичних навичок у обробці текстової інформації на мові сценаріїв JavaScript та відлагодження відповідних програм.

### Порядок виконання.

1. Ознайомитись з методами і властивостями об'єкта **String** і використати деякі з них при організації сценаріїв JavaScript.
2. Створити JavaScript сценарій, що демонструє використання об'єкта **String**.
3. У звіті представити HTML код сторінки з формою як додаток №1.
4. У звіті представити вигляд документу браузером як додаток №2.
5. Підготувати звіт з описом проведеної роботи

### Варіанти завдань

1. Заданий текст з 80 символів. Роздрукувати спочатку всі цифри, які входять в нього, а потім всі решта символів, зберігаючи при цьому взаємне розміщення символів в кожній з цих двох груп.
2. Заданий текст з 80 символів. Визначити, чи заданий текст є симетричним, тобто чи читається він однаково зліва направо так і справа наліво.
3. Скільки різних букв входить у текст із числом символів не більше 100 й обмеженому крапкою (.).
4. Задано натуральне число **k**. Надрукувати **k**-ту цифру послідовності чисел:
  - a. а) 1234567891011121314..., в якій виписані підряд всі натуральні числа;
  - b. б) 11235813..., в якій виписані підряд всі числа Фібоначчі:  $a_n = a_{n-1} + a_{n-2}$ ,  $a_0 = 1$ ,  $a_1 = 1$ .
5. Ввести стрічку довільної довжини, яку трактуватимемо як текст, у якому слова відокремлені одним або кількома пробілами (довжина слів, загалом, довільна). Надрукувати в стовпчик всі слова з непарними номерами.
6. Ввести стрічку довільної довжини, яку трактуватимемо як текст, у якому слова відокремлені одним або кількома пробілами (довжина слів, загалом, довільна). Вивести на друк у вигляді стрічки всі слова з парними номерами, відокремлюючи слова двома пробілами.
7. Ввести стрічку довільної довжини, яку трактуватимемо як текст, у якому слова відокремлені одним або кількома пробілами (довжина слів, загалом, довільна). Вивести всі слова поданого тексту наступним чином: всі слова з непарними номерами надрукувати в першому стовпці, з парними – у другому.
8. Ввести стрічку довільної довжини, яку трактуватимемо як текст, у якому слова відокремлені одним або кількома пробілами (довжина слів, загалом, довільна, але не більша дев'яти символів). Вивести стрічку цифр, де кожна цифра визначає кількість символів у відповідному слові.

9. Ввести стрічку довільної довжини, яку трактуватимемо як текст, у якому слова відокремлені одним або кількома пробілами (довжина слів, загалом, довільна). Вивести слово найбільшої довжини. Якщо таких слів декілька, вивести найперше з них.
10. Ввести стрічку довільної довжини, яку трактуватимемо як текст, у якому слова відокремлені одним або кількома пробілами (довжина слів, загалом, довільна). Вивести слово найменшої довжини. Якщо таких слів декілька, вивести всі ці слова.
11. Ввести стрічку довільної довжини, яку трактуватимемо як текст, у якому слова відокремлені одним або кількома пробілами (довжина слів, загалом, довільна). Вважаючи, що в заданому тексті можливе повторення слів, вивести всі слова в стовпчик без повторень. У другому стовпці надрукувати частоту використання відповідного слова.
12. Ввести стрічку довільної довжини, яку трактуватимемо як текст, у якому слова відокремлені одним або кількома пробілами (довжина слів, загалом, довільна). Враховуючи, що в тексті зустрічаються слова, що містять як букви, так і цифри, а також просто числа, що містять цифри, вивести на екран монітора всі числа.
13. Ввести стрічку довільної довжини, яку трактуватимемо як текст, у якому слова відокремлені одним або кількома пробілами (довжина слів, загалом, довільна). Вважаючи, що текст може містити слова, які складаються як з букв, так із цифр, а також слова, у які цифри не входять, вивести у вигляді стрічки всі слова, що не містять цифри. Слова у стрічці відокремити трьома пробілами.
14. Ввести стрічку довільної довжини, яку трактуватимемо як текст, у якому слова відокремлені одним або кількома пробілами (довжина слів, загалом, довільна). Вивести слова, які починаються лише на певну задану з клавіатури букву, наприклад, букву А.
15. Ввести стрічку довільної довжини, яку трактуватимемо як текст, у якому слова відокремлені одним або кількома пробілами (довжина слів, загалом, довільна). Вивести всі слова, які закінчуються заданою буквою, наприклад літерою Z. Літеру задавати з клавіатури.
16. Ввести стрічку довільної довжини, яку трактуватимемо як текст, у якому слова відокремлені одним або кількома пробілами (довжина слів, загалом, довільна). Вивести лише слова, в які входить задане з клавіатури буквосполучення.
17. Ввести стрічку довільної довжини, яку трактуватимемо як арифметичний вираз, який містить ідентифікатори змінних, константи, знаки арифметичних операцій та дужки. Окремі конструкції можуть бути відокремлені одним або кількома пробілами. Підрахувати кількість відкриваючих та закриваючих дужок і вивести одержану інформацію на екран. Якщо число відкриваючих та закриваючих дужок не рівне між собою, тоді крім числової інформації вивести повідомлення про помилку.
18. Ввести стрічку довільної довжини, яку трактуватимемо як арифметичний вираз, який містить ідентифікатори змінних, константи, знаки арифметичних операцій та дужки. Окремі конструкції можуть бути відокремлені одним або кількома пробілами. Вивести арифметичний вираз, що міститься у дужках із найвищим пріоритетом.
19. Ввести стрічку довільної довжини, яку трактуватимемо як арифметичний вираз, який містить ідентифікатори змінних, константи, знаки арифметичних операцій та дужки. Окремі конструкції можуть бути відокремлені одним або кількома пробілами. Вивести інформацію про частоту використання арифметичних операцій у даному виразі (у першому стовпчику символ операції, у другому – частоту використання).