

Практична робота №1

Тема: *Введення в JavaScript. Змінні, операції, умовні оператори, цикли*

Завдання0.

Опрацювати теоретичний матеріал:

Пункти 1.1-1.4, 2.1-2.13, 3.1-3.3, 5.2-5.3

за посиланням <https://learn.javascript.ru>

Завдання1. Вивід тексту в браузер

- 1) Виведіть на сторінку фразу «Я вчуся програмувати на мові JavaScript».
- 2) Виділіть слово JavaScript тегом .
- 3) Виділіть всю фразу цілком тегом <i>.

Завдання2. Змінні

- 1) Створіть змінну, в якій вкажіть **Ваш вік**.
- 2) Виведіть на сторінку фразу «Мій вік *тут_змінна_з_віком* років».

Завдання3. Типи змінних

- 1) Створіть **строкову** змінну, в якій напишіть **Ваше ім'я**.
- 2) Створіть **булевську** змінну, в якій запишіть відповідь на твердження: «Ви чоловік?». Якщо так, то поставте **true**, якщо ні, то поставте **false**.
- 3) Виведіть на сторінку фразу «Мене звуть *тут_змінна_з_іменем*».
- 4) Виведіть на наступному рядку фразу «Я чоловік? - *Тут_булевська_змінна* ».

Завдання4. Операції

- 1) Дано такий вираз: $(5 + 7) / ((12 - 5.5) * (5 - 8 * 12))$. Розрахуйте на JavaScript і виведіть результат на сторінку.
- 2) Кожне число в виразі зробіть змінною, наприклад, let x = 5; let y = 7 і так далі. При цьому якщо число «5» зустрічається двічі, то не треба створювати другу змінну.
- 3) Підставте в вираз замість чисел змінні, подивіться на результат і переконайтеся, що він збігається з розрахунком в 1-му пункті.
- 4) Створіть 2 рядки «17» і «-5».
- 5) Складіть ці рядки знаком «+».
- 6) Складіть ці рядки знаком «+», пропустивши їх попередньо через функцію Number.

Завдання5. Умовні оператори

- 1) Створіть булевську змінну, в якій запишіть відповідь на твердження: «Ви чоловік?». Якщо так, то поставте true, якщо немає, то поставте false.
- 2) Запишіть умову, в якому перевірте значення булевої змінної, і якщо воно істинне, то виведіть на сторінку «Я чоловік», інакше виведіть на сторінку «Я жінка».
- 3) Створіть змінну, в якій вкажіть Ваш вік.
- 4) Створіть скрипт, який буде виводити словесну інтерпретацію віку. Якщо менше 18, то виведіть «У мене юний вік», якщо 18 і більше до 30, то «Я молодий», якщо від 31 до 50 - «Я дорослий», якщо від 51 до 80, то «Я зрілий», якщо більше 80, то «Я старий».

Завдання6. Оператори циклу

- 1) Використовуючи тег <table> і цикл while виведіть таблицю множення.

Примітка: Вам потрібно використати подвійний цикл, тобто цикл в циклі:

```
for (let i = 1; i < 10; i++) {  
  for (let j = 1; j < 10; j++) {
```

```
    let result = i * j;  
  }  
}
```

Методичні рекомендації

Що таке JavaScript?

JavaScript була створена для додавання інтерактивності в HTML сторінки

JavaScript - це мова сценаріїв або мова скриптів

Мова сценаріїв - це полегшена мова програмування

JavaScript зазвичай вставляється прямо в HTML сторінки

JavaScript є мовою, що інтерпретується (це означає, що скрипти виконуються без попередньої компіляції)

Кожен може використовувати JavaScript без покупки ліцензії

Виведення тексту в JavaScript:

Звичайний текст в JavaScript виводиться в такий спосіб:

```
<html>  
<body>  
<script type = "text / javascript">  
document.write ("Hello World!");  
</ script>  
</ body>  
</ html>
```

Спочатку йде відкриваючий тег `<script>` з атрибутом `type` в значенні `"text/javascript"`. Цей рядок означає, що всередині тега `<script>` йде код, написаний на мові `javascript`. У наступному рядку написано: `"document.write (" Hello World ");"`. `Document` - це об'єкт мови JavaScript, а `write ()` - це метод цього об'єкта, який приймає як параметр рядок (в нашому прикладі - це `"Hello World"`) і виводить його в браузер.

Виведення тексту в JavaScript за допомогою HTML тегів:

```
<Html>  
<Body>  
<Script type = "text / javascript">  
document.write ("<h1> Привіт світ! </ h1>");  
</ Script>  
</ Body>  
</ Html>
```

Оголошення (Створення) змінних JavaScript

Змінні створюються за допомогою ключового слова `"let"`. Після цього ключового слова йде ім'я змінної. В принципі, серйозних обмежень на ім'я змінної немає, проте є певні правила:

1) Змінна не може складатися тільки з одних цифр.

- 2) Змінна не повинна містити символів пробілів.
- 3) Змінна не може бути ключовим словом (наприклад, не можна назвати змінну "let").
- 4) Змінна не може містити різні спецсимволи (лапки, апострофи, окличний і знаки, крапка, кома та інше), однак, дефіс і знак підкреслення використовувати можна.

Після визначення імені змінної можна її форматувати, і через знак "=" написати її значення.

```
let x = 5;  
let y = x + 1;  
x = 6;  
x += 3;  
x ++;
```

Виведення змінних в вікно браузера: виводити їх можна, наприклад, використовуючи функцію `document.write ()`, передавши як параметр ім'я змінної:

```
document.write (x);
```

Щоб з'єднати рядки, використовується знак `+`:

```
<script type="text/javascript">  
    var x = 5;  
    var y = 7;  
    var abc = 7.5;  
    document.write ("x = " + x);  
</script>
```

В браузері виведеться : `x = 5`

Типи змінних

цілий (integer), **дійсний** (double), **строковий** (string), **булевський** (boolean), **масив** (array).
Давайте розберемо кожен тип змінної в JavaScript докладніше.

Змінна будь-якого типу починається з ключового слова "let", і лише значення змінної визначає тип змінної. Більш того, тип змінної в JavaScript може бути змінений в будь-якій точці програми.

Цілий тип - це звичайні ціле число, наприклад, -100, -34, 0, 15, 259 і так далі. Відповідно, цілий тип створюється за допомогою присвоєння змінної цілого значення, наприклад, так:

```
let number = -323;
```

Дійсний тип, або, як прийнято його називати, double. Тип double - це дійсні числа, тобто абсолютно будь-які числа, крім уявних (якщо не знаєте, що таке уявні числа, то не зважайте на це увагу). Наприклад, -3.4, -1.032, 35.599212, 0.0001. Так само як і з іншими типами, для створення дійсної змінної необхідно ініціалізувати змінну одним з подібних значень. Ось приклад:

```
let number = 32.3291;
```

Строковий. Строковий тип в JavaScript використовується дуже часто, не рідше, ніж інші, якщо не більше. Строковий тип - це будь-який рядок. Риклад: "string", "732", "My name's Ivan". Зверніть увагу, що рядок задається в лапках. Зокрема, рядок "732" відрізняється від числа 732. Створюється рядок аналогічно іншим типам в JavaScript:

```
let str = "some string";
```

Булевський. Тип цієї змінної може містити одне з двох значень: true (істина) або false (неправда). Змінна використовується в умовних операторах, про які ми поговоримо пізніше. А поки приклад створення булевою змінної:

```
let bool = true;
```

Масив. Найскладніший тип з усіх, проте, дуже важливий і присутній майже у всіх мовах програмування. Це тип, який містить в собі кілька різних змінних, причому, можливо, що навіть різних типів. Створюється масив в JavaScript таким чином:

```
let array = new Array (3, 7, 12, true, 4.5, "some string", true);
```

У прикладі створюється масив з 7-ми значень, які можуть бути лічені або перезаписані. Постає питання: як звернутися до елементу масиву. Дуже просто! Синтаксис звернення до елементу масиву наступний: назва_масиву [номер_елемента]. Зверніть увагу, що нумерація в масиві починається з "0". Тому елемент зі значенням "3" - це елемент під номером "0". Вивести елемент масиву в вікно браузера і перезаписати його:

```
document.write (array [0]);  
array [2] = array [1] + 1;  
document.write (array [2]);
```

Операції

Розглянемо операції +, —, *, /, %:

```
let x=5;  
let y=7;  
let sum= x+y;  
document.write(x + " + " + y + "=" + sum); // результат: 5 + 7 = 12  
document.write("<br>"); //розрив рядка  
document.write(x + " + " + y + "=" + x + y); //результат: 5 + 7 = 57  
щоб вивести суму як x+y, необхідно записати цей вираз в дужках:  
document.write(x + " + " + y + "=" + (x + y)); // результат: 5 + 7 = 12  
document.write(x + " * " + y + "=" + (x * y)); // результат: 5 * 7 =35  
document.write(x + " / " + y + "=" + (x / y)); // результат: 5 / 7 = 0.714285714285714  
document.write(7 + " % " + 5+ "=" + (7 % 5)); // результат: 7 % 5 =2
```

Інкремент та декремент:

```
x++;  
document.write("Інкремент x = " + x); // результат: Інкремент x = 6  
y--;
```

document.write("Декремент y = " + y); // результат: Декремент y = 6

Строкові операції:

```
let str1="a";
```

```
let str2="b";
```

```
document.write(str1 + str2); // результат: ab
```

```
let str1="12";
```

```
let str2="34";
```

```
document.write(str1 + str2); // результат: 1234
```

Перетворити рядок в число за допомогою функції **Number**:

```
str1 = Number(str1);
```

```
str2 = Number(str2);
```

```
document.write(str1 + str2); // результат: 46
```

або так:

```
document.write(Number(str1) + Number(str2)); // результат: 1234
```

Логічні операції: аналогічні логічним операціям в C++

оператор	опис	приклад
==	дорівнює	x == 8 це брехня x == 5 це істина
===	точно так само (значення і тип збігаються)	x === 5 це істина x === "5" це брехня
!=	не дорівнює	x != 8 це істина
>	більше ніж	x > 8 це брехня
<	менше ніж	x < 8 це істина
>=	більше або дорівнює	x >= 8 це брехня
<=	менше або дорівнює	x <= 8 це істина

```
let x = 2.5;
```

```
let y = 3.7;
```

```
let b = x > y;
```

```
document.write(x + ">" + y + "=" + b); // результат: 2.5 > 3.7 = false
```

```
let b = x < y;
```

```
document.write(x + "<" + y + "=" + b); // результат: 2.5 < 3.7 = true
```

оператор	опис	приклад
&&	логічне І	(X < 10 && y > 1) це істина
	логічне АБО	(X == 5 y == 5) це брехня
!	логічне НЕ	!(X == y) це істина

Умовні оператори

У JavaScript є три умовних оператори:

- if

- switch
- тернарний ?:

Умовний перевіряє чи є умова істиною (true, тип **Boolean**), якщо так то виконується відповідний код. Також можна виконати інший код якщо умова не виконується (дорівнює false).

Приклад умовного оператора if:

```
if(3>1) document.write("три більше один");
else document.write ("три не більше один");
```



== - порівняння. Використовується якщо потрібно порівняти обидва значення:

```
let x=3, y=3;
if(x == y) document.write ("x дорівнює y");
```



=== - подвійне порівняння. Подвійне порівняння використовується для більш точного порівняння.

Наприклад якщо порівнювати число 5 і рядок "5" то == дорівнює true, а при === дорівнює false:

```
let x=5, y="5";
if(x == y) document.write ("5==\"5\" – true");
if(x === y) document.write ("5===\"5\" – true");
else document.write ("5===\"5\" – false");
```

Оператори циклу

Робота з циклами в JavaScript аналогічна роботі з циклами в C++.

Існує кілька видів циклів: **for**, **while** і **do-while**.

Цикл for

for (змінна_ітерації = початкове_значення; умова; дія_після_кожної_ітерації)
 {
 // Код програми
 }

Приклад:

```
for (i = 0; i < 100; i ++)  
  document.write (i + " ");
```

Цикл while

```
while (умова) {  
  // Код програми  
}
```

Приклад:

```
let i = 0;
while (i < 100) {
  i++;
  document.write (i + "");
}
```

Цикл do-while

```
do {
  // Код програми
} While (умова)
```

Приклад:

```
let i = 0;
do {
  i++;
  document.write (i + "");
} while (i < 100)
```

Оператор break

Даний оператор дозволяє достроково вискочити з циклу:

```
for (i = 0; i < 100; i++) {
  if (i == 50) break;
  document.write (i + "");
}
```

Ви можете запустити цей скрипт і виявите, що вивелися тільки числа до **49**, так як при **i = 50** цикл перервався, завдяки оператору **break**.

Оператор continue.

Даний оператор дозволяє перейти до наступної ітерації циклу:

```
for (i = 0; i < 100; i++) {
  if (i == 50) continue;
  document.write (i + "");
}
```

Якщо Ви запустите цей скрипт, то побачите, що не вистачає числа **50**. Це сталося через те, що при **i = 50**, ми переходимо до наступної ітерації циклу, перед якою **i** збільшується на **1** і стає рівним **51**-му.