

# Лабораторна робота № 3

## Використання функцій JavaScript

*Мета роботи: навчитися використовувати стандартні і створювати функції користувача в JavaScript.*

### Теоретична частина

Функції є набором інструкцій, що виконують певну дію або обчислюють певне значення. Синтаксис визначення функції:

```
function ім'я_функції ( [параметр [, ...]]) {  
    // Інструкції  
}
```

Визначення функції починається з ключового слова `function`, після якого слідує ім'я функції. Найменування функції підпорядковується тим самим правилам, як і найменування змінної: воно може містити лише цифри, літери, символи підкреслення і долара (\$) і має починатися з літери, символу підкреслення чи символу долара.

Після імені функції у дужках йде перелік параметрів. Навіть якщо параметрів функції немає, то просто йдуть порожні дужки. Далі у фігурних дужках йде тіло функції, що містить набір інструкцій.

Щоб виконати тіло функції, необхідно викликати функцію з передачею параметрів або без параметрів:

```
function sum (a,b,c) {  
    let d = a + b + c;  
    console.log(d);  
}  
sum(1, 2, 3);  
sum();
```

На відміну від компільованих мов програмування, де кількість фактичних параметрів, що передаються при виклику функції, і кількість формальних параметрів, описаних при оголошенні, повинні збігатися, JavaScript дозволяє не дотримуватися цього правила. Якщо для параметрів не передається значення, то за замовченням вони мають значення "undefined".

Для визначення кількості переданих параметрів та доступу до їх значень при виклику функції створюється масив `arguments`, доступний усередині тіла функції.

Є спосіб визначення значення для параметрів за замовчуванням:

```
function sum (a=1, b=2, c=3) {  
    let d = a + b + c;  
    console.log(d);  
}
```

За допомогою spread-оператора можна вказати, що за допомогою параметра можна передати змінну кількість значень:

```
function display(season, ... temps) {
    console.log(season);
    for (index in temps) {
        console.log(temps[index]);
    }
}
display("Весна", -2, -3, 4, 2, 5);
display("Літо", 20, 23, 31);
```

Функція може повертати результат. Для цього використовується оператор return :

```
function sum (a, b) {
    return a + b;
}
console.log(sum (1, 2));
```

Функції можуть виступати як інші параметри. JavaScript дозволяє визначати функції всередині функцій та повертати результат у вигляді функції.

Серед функцій можна окремо виділити рекурсивні функції. Їхня суть полягає в тому, що функція викликає саму себе.

Розглянемо, наприклад, функцію, що визначає факторіал числа:

```
function getFactorial(n) {
    if (n===1) {
        return 1;
    }
    else {
        return n*getFactorial(n - 1);
    }
}
let result=getFactorial(4);
console.log(result); // 24
```

## ***Практична частина***

### **Використання вбудованих функцій**

**Завдання 1-2.** Створіть файли lab3\_1.html та lab3\_2.html, які демонструють використання вбудованих функцій об'єкта Date.

```
<!DOCTYPE HTML>
<html>
    <head>
        <meta charset="UTF-8">
```

```

        <title>Прізвище, група</title>
    </head>
    <body>
        <script>
            "use strict";
            let today=new Date();
            let hours=today.getHours();
            let minute=today.getMinutes();
            if (minute<10)minute="0" + minute;
            if (hours < 12) {
                var time12 = hours + ":" + minute + " am ";
            }
            else {
                var time12 = hours - 12 + ":" + minute + " pm ";
            }
            console.log("поточний час -", time12);
            alert("поточний час -" + time12);
        </script>
    </body>
</html>

```

У скрипті, використовуючи аргументи size, day та color функції weekday(), відобразіть назви днів тижня різними розмірами та кольорами:

понеділок

вівторок

середа

четвер

п'ятниця

субота

неділя

```

.*****
<!DOCTYPE HTML>
<html>

    <head>

```

```

        <meta charset="UTF-8">

        <title>Прізвище, група</title>
    </head>
    <body>
        <script>
            "use strict";

            let today = new Date();

            let newYearDay = new Date( today.getFullYear () + 1, 0,
1);

            const day = 86400000;
            const hour = 3600000;
            const minute = 60000;

            let time = newYearDay.getTime() - today.getTime();
            let days = Math.floor(time/day);

            let hours = Math.floor((time-days*day)/hour);

            let minutes = Math.floor((time-days*day-
hours*hour)/minute);

            let str = days + " днів, " + hours + " годин, " +
minutes + " минут.";

            console.log("До Нового року залишилося ", str);

            alert("До Нового року залишилося " + str);
        </script>
    </body>
</html>

```

## Створення користувацьких функцій

**Завдання 3.** Створіть файл lab3\_3.html, що демонструє створення функції користувача з аргументами і звернення до неї з основної частини скрипту:

```

<!DOCTYPE HTML>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Прізвище, група</title>
    </head>
    <body>
        <script>

```

```

        //Опис функції виведення рядка з переходом на новий
        function PrintBR(txt) {
            document.write(txt, "<br />");
        }
        // Звертання до функції з передачею фактичних параметрів
        PrintBR ("Це перший рядок");
        PrintBR ("Це другий рядок");
        PrintBR ("Це ще один рядок");
    </script>
</body>
</html>

```

**Завдання 4.** Створіть файл lab3\_4.html, що демонструє створення користувацької функції, що повертає значення.

```

<!DOCTYPE HTML>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Прізвище, група</title>
    </head>
    <body>
        <script>
            function AddNums(firstnum,secondnum) {
                return firstnum + secondnum;
            }
            document.write("3 + 5 = ", AddNums (3, 5));
            let a = 5;
            let b = 25;
            document.write("<br>", a, " + ", b, " = ", AddNums(a, b));
        </script>
    </body>
</html>

```

**Завдання 5.** Створіть файл lab3\_5.html, що демонструє використання функції з необов'язковими параметрами та параметрами за замовчуванням:

```

<!DOCTYPE HTML>
<html>
    <head>
        <meta charset="UTF-8">

```

```

        <title>Прізвище, група</title>
</head>
<body>
    <script>
        "use strict";
        function printText(txt, size = 14) {
            document.write(`<div style="font-
size:${size}pt">${txt}</div>`);
        }
        printText("<p>Великий шрифт", 32);
        printText("<p>Шрифт за замовченням - перший рядок");
        printText("<p>Шрифт за замовченням - другий рядок");
        printText("<p>Малий шрифт", 8);
    </script>
</body>
</html>

```

**Завдання 6.** Створіть файл lab3\_6.html, що демонструє передачу функцію як аргумент іншої функції:

```

<!DOCTYPE HTML>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Прізвище, група</title>
    </head>
    <body>
        <script>
            function kvadrat(a) {
                return a * a;
            }
            function polinom(a, k) {
                return k(a) + a + 5;
            }
            let result=polinom (4, kvadrat);
            document.write("<h2> result =", result);
        </script>
    </body>
</html>

```

**Завдання 7.** Створіть файл lab3\_7.html, що демонструє використання функції як змінної:

```
<!DOCTYPE HTML>

<html>

  <head>

    <meta charset="UTF-8">

    <title>Прізвище, група</title>

  </head>

  <body>

    <script>

      let i = 5;

      function f(a,b,c) {

        if (a > b) return c;

      }

      document.write("Значення змінної i: " + i.valueOf());

      document.write("<p>Значення змінної f:<BR>" +

f.valueOf());

    </script>

  </body>

</html>
```

**Завдання 8.** Створіть файл lab3\_8.html, що демонструє використання масиву arguments:

```
<!DOCTYPE HTML>

<html>

  <head>

    <meta charset="UTF-8">

    <title>Прізвище, група</title>

  </head>

  <body>

    <script>

      function mean(a, b) {

        if (arguments.length > mean.length) {

          alert("Аргументів більше, ніж треба");

        }

        var result = 0;

        for (var i = 0; i < arguments.length; i++) {

          result += arguments [i];

        }

        return result/arguments.length ;

      }

    </script>

  </body>

</html>
```

```
    }  
    document.write ("Середнє -", mean (2, 9), " ");  
    document.write ("Середнє -", mean (7, 4, 6));  
  </script>  
</body>  
</html>
```

### ***Контрольні питання***

1. Що таке функція?
2. Як створити функцію користувача?
3. Що таке формальні та фактичні параметри?
4. Чим відрізняється опис функцій, які не повертають та повертають значення?
5. У чому особливість використання функції як змінної?
6. У чому особливість використання функції як об'єкт?
7. Навіщо використовується функція eval?
8. Чим відрізняються глобальні та локальні змінні?
9. Як встановити значення параметра функції за промовчанням?
10. Навіщо використовується масив arguments?