# Лабораторна робота №4

# Робота з масивами JavaScript

#### Мета роботи: вивчити можливості JavaScript для створення та обробки масивів.

## Теоретична частина

Для роботи із наборами даних призначені масиви. Для створення масиву застосовується літерал масиву або конструкція new Array():

let array\_name1 = [item1, item2, ...];

let array\_name2 = новий Array([item1, item2, ...]);

Для підвищення продуктивності та читабельності програмного коду рекомендується використовувати літерал масиву.

Для отримання доступу до елементів масиву використовується індекс. Індексація елементів починається з нуля:

let cars = ["Saab", "Volvo", "BMW"];

console.log(cars[0]); // Saab

Індех використовується як для читання, так запису елемента масиву. Причому на відміну від інших мов, таких як C# або Java, можна встановити елемент, який спочатку не встановлено:

let cars = ["Saab", "Volvo", "BMW"];

cars[10] = "Toyota";

console.log(cars[10]); // Toyota

console.log (cars [3]); // undefined

На відміну від низки мов програмування JavaScript масиви не є строго типізованими, один масив може зберігати дані різних типів:

var objects = ["Tom", 12, true, 3.14, false];

console.log(objects.toString());

Масиви можуть бути одновимірними та багатовимірними. Кожен елемент у багатовимірному масиві може бути окремим масивом.

const students = [

 ["Зеленський", 20, 5.5],

 ["Залужний", 18, 8.2],

 ["Кривоніс", 21, 7.8]

];

students[0][1] = 19; // привласнюємо окреме значення

console.log(students[0][1]);

У мові JavaScript всі властивості та методи обробки масивів зібрані у глобальному об'єкті Array.prototype, від якого автоматично успадковуються всі масиви, що створюються.

Всі масиви мають властивість length, якою встановлює або повертає кількість елементів у масиві:

let cars = ["Saab", "Volvo", "BMW"];

console.log(cars.length); // 3

cars.length = 5;

console.log(cars[4]); // undefined

## Методи масивів

Для додавання та видалення елементів масиву використовуються такі методи:

* push(...items) – додає елементи в кінець.
* pop() – витягує елемент з кінця.
* shift() – витягує елемент з початку.
* unshift(...items) – додає елементи до початок.

Для видалення елементу масиву можна використовувати оператор delete. Однак цей оператор видаляє лише значення елемента із заданим ключем без переіндексації:

let cars = ["Saab", "Volvo", "BMW"];

delete cars [1];

console.log(cars.length); // 3

console.log(cars[1]); // undefined

Універсальний метод splice() використовується для додавання, видалення та заміни елементів масиву:

splice(index[, deleteCount, elem1, ..., elemN])

Він починає з позиції index видаляти deleteCount елементи і вставляти elem1, ..., elemN на їх місце. Повертає масив із віддалених елементів.

Метод slice() повертає новий масив, який копіює елементи, починаючи з індексу start і до end (не включаючи end). Обидва індекси start і end можуть бути від’ємними. У такому разі відлік буде здійснюватися з кінця масиву:

slice([start], [end])

Метод forEach() дозволяє запускати функцію кожного елемента масиву. Його синтаксис:

forEach(function(item, index, array) {

 // ... робити щось з item

});

Функція зворотного виклику (callback) викликається по черзі кожному за елемента масиву і приймає такі параметри:

* item – черговий елемент;
* index – його індекс;
* array - сам масив.

Для пошуку елементів у масиві використовуються такі методи:

* indexOf(item, from) шукає item, починаючи з індексу from, і повертає індекс, на якому було знайдено шуканий елемент, інакше -1.
* lastIndexOf(item, from) – те саме, але шукає справа наліво.
* includes(item, from) – шукає item, починаючи з індексу from, та повертає true, якщо пошук успішний.

Методи find, findIndex і filter як умови пошуку використовують функцію-предикат:

let result = arr.find(function(item, index, array) {

// якщо true - повертається поточний елемент і перебір переривається

// якщо всі ітерації виявилися хибними, повертається undefined

});

let result = arr.findIndex(function(item, index, array) {

// якщо true - повертається індекс, у якому було знайдено елемент, і перебір переривається

// Якщо всі ітерації виявилися помилковими, повертається -1

});

let results = arr.filter(function(item, index, array) {

// якщо true - елемент додається до результату, і перебір продовжується

// Повертається порожній масив у разі, якщо нічого не знайдено

});

Метод map() є одним з найкорисніших і найчастіше використовуваних. Він викликає функцію для кожного елемента масиву та повертає масив результатів виконання цієї функції:

let result = arr.map(function(item, index, array) {

// Повертається нове значення замість елемента

});

Метод sort(fn) сортує масив на місці, змінюючи у ньому порядок елементів. Він повертає відсортований масив, але зазвичай значення, що повертається, ігнорується, оскільки змінюється сам масив.

Повний список методів є у [довіднику MDN](https://developer.mozilla.org/ru/docs/Web/JavaScript/Reference).

## Практична частина

**Завдання 1.** У файлі зі скриптом lab4\_1.html:

1. Створіть масив Treag «трикутних» чисел, тобто, чисел виду n(n+1)/2 (де n=1,2,… 10) і виведіть значення цього масиву на екран у рядок (через 2 пробіли).
2. Створіть масив kvd квадратів натуральних чисел від 1 до 10, виведіть значення цього масиву на екран у рядок.
3. Об'єднайте ці 2 масиви в масив rez, виведіть результат на екран.
4. Відсортуйте масив rez у зворотному порядку, виведіть результат на екран.

**Завдання 2.** У файлі зі скриптом lab4\_2.html:

Сформуйте одновимірний масив (не менше 15 елементів, для генерації значень масиву можна скористатися функцією random об'єкта Math), виведіть значення цього масиву на екран у рядок.

Виконайте завдання відповідно до вашого номера в журналі, не використовуючи вбудовані функції.

1. Знайти максимальний елемент та поміняти його місцями з останнім елементом масиву.
2. Знайти мінімальний елемент та поміняти його місцями з попереднім елементом масиву.
3. Знайти мінімальний елемент та поміняти його місцями з наступним елементом масиву.
4. Знайти максимальний елемент та поміняти його місцями з шостим елементом масиву.
5. Знайти максимальний елемент, надати його значення останньому елементу масиву, а замість максимального числа записати – 1.
6. Знайти мінімальний елемент, присвоїти його значення першому елементу масиву, а замість мінімального елемента записати число 9999.
7. Знайти мінімальний елемент та поміняти його місцями з третім елементом масиву.
8. Знайти мінімальний елемент та замінити його на суму першого та останнього елементів.
9. Знайти максимальний елемент та поміняти його місцями з передостаннім елементом масиву.
10. Знайти мінімальний елемент, присвоїти його значення останньому елементу масиву, а замість мінімального елемента записати значення 3N.
11. Знайти мінімальний елемент і поміняти його місцями з елементом масиву, номер якого задано.
12. Знайти максимальний елемент і поміняти його місцями зі другим елементом масиву.
13. Знайти мінімальний елемент та поміняти його місцями з останнім елементом масиву.
14. Знайти мінімальний елемент та поміняти його місцями з передостаннім елементом масиву.
15. Знайти мінімальний елемент та поміняти його місцями з третім елементом масиву.

**Завдання 3.** У файлі зі скриптом lab4\_3.html:

Сформуйте одновимірний масив (не менше 15 елементів, для генерації значень масиву можна скористатися функцією random об'єкта Math), виведіть значення цього масиву на екран у рядок.

Виконайте завдання відповідно до вашого номера в журналі, використовуючи вбудовані методи масивів.

1. Знайти кількість чисел, що належать проміжку [a,b], та суму чисел, що стоять на місцях, кратних 3.
2. Знайти суму чисел, менших за задане D, і кількість чисел, що стоять на парних місцях і більших за задане С.
3. Знайти добуток усіх чисел, що стоять на місцях, кратних 4, та кількість чисел, не більше за задане A.
4. Знайти кількість чисел, менших за задане X, і добуток усіх від’ємних чисел, що знаходяться на непарних місцях.
5. Знайти кількість чисел, що не належать до проміжку (X,Y], та суму від’ємних чисел, що знаходяться на парних місцях.
6. Знайти кількість невід'ємних чисел і визначити суму чисел, що знаходяться на місцях, кратних 3 і не рівних заданому F.
7. Знайти середнє арифметичне від’ємних чисел і визначити кількість чисел, що за величиною великих А і стоять на парних місцях.
8. Знайти середнє арифметичне додатних чисел, що стоять на непарних місцях, та кількість чисел, менших за задане B.
9. Знайти середнє арифметичне чисел, що належать проміжку [A,B), та кількість додатних чисел, що стоять на місцях, кратних 4.
10. Знайти середнє арифметичні чисел, нерівних заданому С, і добуток не додатних чисел, що стоять на парних місцях.
11. Знайти середнє арифметичне чисел, більших за задане D і які стоять на непарних місцях, і визначити кількість чисел, не більших за задане F.
12. Знайти середнє арифметичне чисел, що не потрапляють у проміжок [A,B], та кількість додатних чисел, що стоять на місцях, кратних 3.
13. Знайти середнє арифметичне ненульових чисел і кількість чисел, що за величиною не більше A і що стоять на парних місцях.
14. Обчислити добуток чисел, що належать проміжку (A,B], та кількість від’ємних чисел, що стоять на місцях, кратних 3.
15. Знайти середнє арифметичне додатних чисел, що стоять на непарних місцях, і добуток чисел, менших за задане С.

**Завдання 4.** У файлі зі скриптом lab4\_4.html створіть масиви

fruits=['apple','pineapple','mango','melon','grape'];

citrus=['orange','lemon','lime'].

Послідовно, у кожному окремому рядку, виведіть:

* масив fruits;
* масив citrus;
* об'єднаний масив fruits;
* масив fruits, в якому після mango вставлено: pear, cherry, plum, raspberry, strawberry;
* масив fruits, в якому видалено останні 3 елементи;
* виведіть відсортований в алфавітному порядку масив fruits;
* виведіть масив fruits у зворотному алфавітному порядку;
* виведіть масив fruits, відсортований за довжиною слів.

## Контрольні питання для захисту:

1. Що таке масив?
2. Які масиви використовуються в JavaScript?
3. Як індексуються елементи масиву?
4. Як створюється багатовимірний масив?
5. Як переглянути структуру та значення елементів масиву?
6. Як вивести значення елементів масиву у вікно браузера?
7. Як відсортувати масив за зростанням та спаданням?
8. Які методи об'єкту Array ви знаєте?
9. Як додати елементи в масив?
10. Як об'єднати кілька масивів?
11. Як видалити елементи масиву?