

ЛАБОРАТОРНА РОБОТА № 3

Робота з одновимірними та багатовимірними масивами.

Мета: *Опрацювати завдання з масивами в JS*

Хід роботи:

Завдання 1.

1.1. Розмістіть на сторінці **div**-блок з ідентифікатором **“paragraphs”**. Розмістіть у ньому 5 абзаців з довільним текстом. Сформууйте масив, який міститиме 5 елементів - кількість символів, які входять до відповідного абзацу тексту.

Для пошуку абзаців використовуйте метод **querySelectorAll** та відповідний CSS-селектор. Для доступу до тексту, який входить в абзац використовуйте властивість **innerText** для DOM-елемента. Виведіть у консоль отриманий масив.

1.2. Дано масив цілих чисел:

```
let arr = [20, 17, 4, -4, 10, -9, 13, 4, 12, 22, 13, 19, 1, 3];
```

- Для заданого масиву визначте та виведіть у консоль:

- найменше та найбільше значення;
- кількість парних та непарних чисел;
- кількість чисел, що містять дві цифри;
- кількість додатних та кількість від'ємних чисел.

- Відсортуйте масив **arr** за допомогою методу **Sort** у порядку зростання значень та виведіть у консоль результат.

- Створіть масив з оберненим порядком елементів та виведіть його у консоль.

					ДУ «Житомирська політехніка».24.121.29.003 – Лр3							
Змн.	Арк.	№ докум.	Підпис	Дата								
Розроб.		Цимбалюк Р.І.			Звіт з лабораторної роботи				Літ.	Арк.	Аркушів	
Перевір.		Дмитренко І.А.									1	12
Керівник									ФІКТ Гр. ІПЗ-23-4[2]			
Н. контр.												
Зав. каф.												

Лістинг програми: HTML:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Task1</title>
</head>
<body>
<div id="paragraphs">
  <p>
    "Yare Yare Daze" – класична фраза Джотаро Куджо. Цей вираз,
    що перекладається як "Ну й ну...", відображає його незворушність і
    спокій у найскладніших ситуаціях. Незважаючи на всі труднощі,
    Джотаро завжди зберігає холоднокровність, показуючи, що навіть у найважчі
    моменти не варто втрачати самовладання.
  </p>
  <p>
    "ZA WARUDO! Toki wo tomare!" – коли Діо вимовляє ці слова, час буквально зупиняється.
    Його здатність контролювати час символізує абсолютну владу, яку він прагнув захопити.
    Це нагадує про те, що сила та амбіції можуть підкорити навіть неблаганний потік часу.
  </p>
  <p>
    "I reject my humanity, JoJo!" – зухвалий виклик Діо Брандо своєму ворогу і
    всьому людству. Це гучна заява про відмову від людської природи заради
    безсмертної сили, що показує безмежне бажання влади та егоїзм.
  </p>
  <p>
    "Your next line is..." – Жозеф Джостар обожнював здивовувати своїх супротивників
    цією фразою, коли він передбачав, що вони скажуть далі. Його хитрість і почуття гумору
    роблять його одним із найхаризматичніших персонажів серіалу. Це нагадування,
    що інтелект і стратегія можуть бути більш потужними, ніж груба сила.
  </p>
  <p>
    "Good grief..." – ще один вислів Джотаро Куджо, який символізує
    його постійне невдоволення та роздратування через непередбачувані обставини.
    Його фраза стала мемом серед фанатів, і багато хто використовує її для опису
    своїх власних "втомливих" ситуацій у житті.
  </p>
</div>
<script src="/JS/script1.js"></script>
</body>

</html>
```

		Цимбалюк Р.І.			ДУ «Житомирська політехніка».24.121.29.001 – Лр3	Арк.
		Дмитренко І.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		2

JS:

```
// Отримуємо всі абзаци та обчислюємо їх довжину
let paragraphs :NodeListOf<HTMLParagraphElement> = document.querySelectorAll( selectors: 'p');
let pArray :any[] = [];

paragraphs.forEach( callbackfn: paragraph :HTMLParagraphElement => {  Цимбалиук Руслан Іванович
    pArray.push(paragraph.innerText.length); // Додаємо довжину кожного абзаци в масив
});

console.log(pArray); // Виводимо масив з довжинами абзацив

// Масив чисел для подальших операцій
let arr :number[] = [20, 17, 4, -4, 10, -9, 13, 4, 12, 22, 13, 19, 1, 3];

// Знаходимо максимальне та мінімальне значення
let max :number = Math.max(...arr);
let min :number = Math.min(...arr);

console.log("Максимальне: " + max);
console.log("Мінімальне: " + min);

// Підрахунок кількості парних та непарних елементів
let evenCount :number = arr.filter(el :number => el % 2 === 0).length;
let oddCount :number = arr.filter(el :number => el % 2 !== 0).length;

console.log("Кількість:");
console.log("Парні: " + evenCount);
console.log("Непарні: " + oddCount);

// Функція для перевірки, чи число є двоцифровим
function hasTwoDigits(num) :boolean { Show usages  Цимбалиук Руслан Іванович
    num = Math.abs(num); // Перетворюємо на додатне для правильної перевірки
    return num.toString().length === 2; // Перевіряємо кількість цифр
}

// Підрахунок двоцифрових чисел
let twoDigitNumbersCount :number = arr.filter(el :number => hasTwoDigits(el)).length;
console.log("Двоцифрові числа: " + twoDigitNumbersCount);

// Підрахунок додатних та від'ємних чисел
let positiveCount :number = arr.filter(el :number => el > 0).length;
let negativeCount :number = arr.filter(el :number => el < 0).length;

console.log("Додатні: " + positiveCount);
console.log("Від'ємні: " + negativeCount);

// Сортування масиву по зростанню
console.log("Сортований масив: ", arr.sort( compareFn: (a :number , b :number ) => a - b));

// Реверс масиву після сортування
console.log("Реверсований масив: ", arr.reverse());
```

Результат виконання програми:

Рис.1. Результат виконання програми

		Цимбалиук Р.І.			ДУ «Житомирська політехніка».24.121.29.001 – Лр3	Арк.
		Дмитренко І.А.				3
Змн.	Арк.	№ докум.	Підпис	Дата		

Завдання 2.

- Розмістіть на сторінці 3 div-блоки з класами **"table"**. У кожному з них розмістіть таблицю. У першому блоці - таблицю розміром 3x3, другому - 4x4, третьому - 5x5.
- В CSS-файлі розмістіть стиль для класу **"selected"**, який передбачає жовтий фон.
- За допомогою **querySelectorAll** знайдіть **div**-блоки з класом **"table"**. Для кожного знайденого блоку у циклі знайдіть всі вкладені у нього теги **td**. Для кожного другого тега **td** додайте клас **"selected"**.

Завдання 3.

Створіть матрицю розміру N на M, яка містить цілі числа (наприклад, 7x5 та числами на власний розсуд). Знайдіть та виведіть у консоль:

- кількість додатних елементів;
- кількість рядків, які не містять жодного нульового елемента;
- кількість стовпців, які містять хоча б один нульовий елемент;
- номер рядка, в якому знаходиться найдовша серія однакових елементів;
- добуток елементів в тих рядках, які не містять від'ємних елементів;
- суму елементів в тих стовпцях, які не містять від'ємних елементів;
- суму елементів в тих стовпцях, які містять хоча б один від'ємний елемент;
- транспоновану матрицю;

		Цимбалюк Р.І.			ДУ «Житомирська політехніка».24.121.29.001 – Лр3	Арк.
		Дмитренко І.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		4

Лістинг програми:

HTML:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Task 2</title>
  <link rel="stylesheet" href="/CSS/Style1.css">
</head>
<body>
<div class="table">
  <table>
    <tr>
      <td></td>
      <td></td>
      <td></td>
    </tr>
    <tr>
      <td></td>
      <td></td>
      <td></td>
    </tr>
    <tr>
      <td></td>
      <td></td>
      <td></td>
    </tr>
  </table>
</div>
<div class="table">
  <table>
    <tr>
      <td></td>
      <td></td>
      <td></td>
      <td></td>
    </tr>
    <tr>
      <td></td>
      <td></td>
      <td></td>
      <td></td>
    </tr>
    <tr>
      <td></td>
      <td></td>
      <td></td>
      <td></td>
    </tr>
    <tr>
      <td></td>
      <td></td>
      <td></td>
      <td></td>
    </tr>
  </table>
</div>
```

		Цимбалюк Р.І.			ДУ «Житомирська політехніка».24.121.29.001 – ЛрЗ	Арк.
		Дмитренко І.А.				5
Змн.	Арк.	№ докум.	Підпис	Дата		

```

<div class="table">
  <table>
    <tr>
      <td></td>
      <td></td>
      <td></td>
      <td></td>
      <td></td>
    </tr>
    <tr>
      <td></td>
      <td></td>
      <td></td>
      <td></td>
      <td></td>
    </tr>
    <tr>
      <td></td>
      <td></td>
      <td></td>
      <td></td>
      <td></td>
    </tr>
    <tr>
      <td></td>
      <td></td>
      <td></td>
      <td></td>
      <td></td>
    </tr>
    <tr>
      <td></td>
      <td></td>
      <td></td>
      <td></td>
      <td></td>
    </tr>
  </table>
</div>
<script src="/JS/script2.js"></script>
</body>
</html>

```

CSS:

```

/* Основний стиль для таблиці */
div.table {
  margin: 15px;
}

/* Стиль таблиці та ячеек */
table {
  border-collapse: collapse; /* З'єднання меж таблиці */
}

table,
td {
  border: 1px solid black; /* Чорна межа для таблиці та ячеек */
}

td {
  padding: 20px; /* Відступи всередині ячеек */
}

/* Стиль для вибраної ячейки */
.selected {
  background-color: yellow; /* Жовтий фон для вибраних ячеек */
}

```

		Цимбалюк Р.І.			ДУ «Житомирська політехніка».24.121.29.001 – ЛрЗ	Арк.
		Дмитренко І.А.				6
Змн.	Арк.	№ докум.	Підпис	Дата		

JS:

```
// Отримуємо всі div з класом "table"
let divTags :NodeListOf<Element> = document.querySelectorAll( selectors: "div.table");

// Проходимо через кожен div
divTags.forEach( callbackfn: div :Element => {  Цимбалюк Руслан Іванович
    // Отримуємо всі таблиці в поточному div
    div.querySelectorAll( selectors: "table").forEach( callbackfn: table :HTMLTableElement => {
        // Отримуємо всі елементи td в таблиці
        let tdElements :NodeListOf<HTMLTableCellElement> = table.querySelectorAll( selectors: "td");

        // Проходимо через всі td, починаючи з другого, з кроком 2
        for (let i :number = 1; i < tdElements.length; i += 2) {
            // Додаємо клас "selected" до кожного другого td
            tdElements[i].className = "selected";
        }
    });
});
```

Результат виконання програми:

Рис.2. Результат виконання програми

ADVANCED TASKS:

- * максимум серед сум елементів діагоналей, паралельних головній діагоналі матриці (завдання на додаткові бали, приймається із обов'язковим усним поясненням та додатковими запитаннями від викладача);
- * мінімум серед сум модулів елементів діагоналей, паралельних побічній діагоналі матриці (завдання на додаткові бали, приймається із обов'язковим усним поясненням та додатковими запитаннями від викладача).

Лістинг програми:
HTML:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Task 3</title>
</head>
<body>
<script src="/JS/script3.js"></script>
</body>
</html>
```

		Цимбалюк Р.І.			ДУ «Житомирська політехніка».24.121.29.001 – Лр3	Арк.
		Дмитренко І.А.				8
Змн.	Арк.	№ докум.	Підпис	Дата		

JS:

```
// Визначення матриці
const matrix :number[] = [
  [3, 7, 0, 4, 0],
  [6, -8, 1, -9, 0],
  [12, 5, 6, 6, 7],
  [9, 7, 8, 8, 8],
  [-35, 12, 9, 0, 4]
];

// Виведення матриці
console.log("Матриця: ", matrix);

// Обчислення кількості додатних елементів
let positiveCount :number = matrix
  .map(row :number[] => row.filter(el :number => el > 0).length)
  .reduce((sum :number, el :number) => sum + el, 0);
console.log("Додатні елементи: " + positiveCount);

// Кількість рядків без нулів
let rowsWithoutZeroCount :number = matrix
  .filter(row :number[] => !row.includes(0)).length;
console.log("Кількість рядків, які не містять нуль: " + rowsWithoutZeroCount);

// Функція для обчислення кількості стовпців, що містять нуль
function getColsWithZeroCount(matrix) :number { Show usages Цимбалюк Руслан Іванович
  const rows = matrix.length;
  const cols = matrix[0].length;
  let count :number = 0;
  for (let i :number = 0; i < cols; i++) {
    for (let j :number = 0; j < rows; j++) {
      if (matrix[j][i] === 0) {
        count++;
        break; // Перериваємо, якщо знайшли нуль
      }
    }
  }
  return count;
}

console.log("Кількість стовпців, які містять нуль: " + getColsWithZeroCount(matrix));

// Функція для знаходження індексу рядка з найдовшою серією повторюваних елементів
function getRowWithLongestSeriesIndex(matrix) :number { Show usages Цимбалюк Руслан Іванович
  let longestSeries :number = 1;
  let index :number = -1;

  for (let i :number = 0; i < matrix.length; i++) {
    let series :number = 1;
    for (let j :number = 1; j < matrix[i].length; j++) {
      if (matrix[i][j - 1] === matrix[i][j]) {
        series++;
      } else {
        if (series > longestSeries) {
          longestSeries = series;
          index = i; // Оновлення індексу рядка
        }
      }
    }
  }
}
```

		Цимбалюк Р.І.			ДУ «Житомирська політехніка».24.121.29.001 – ЛрЗ	Арк.
		Дмитренко І.А.				9
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        series = 1; // Скидання серії
    }
}

// Перевірка в кінці рядка
if (series > longestSeries) {
    longestSeries = series;
    index = i;
}
}
return index;
}

let index :number = getRowWithLongestSeriesIndex(matrix);
index === -1 ?
    console.log("Серії повторюваних елементів немає") :
    console.log("Індекс рядка з найдовшою серією: " + index);

// Обчислення добутку елементів у рядках без від'ємних елементів
let productArrayWithoutNegativeElementsInRows :number[] = matrix
    .filter(row :number[] => row.every(el :number => el >= 0)) // Вибір рядків без від'ємних елементів
    .map(row :number[] => row.reduce((product :number , el :number ) => product * el, 1)); // Обчислення добутку

console.log("Добуток елементів у рядках без від'ємних елементів: ", productArrayWithoutNegativeElementsInRows);

// Функція для видалення стовпців з матриці
function removeColsFromMatrix(matrix, colsToRemove) { Show usages Цимбалюк Руслан Іванович
    return matrix.map(row => row.filter((_, index :number ) => !colsToRemove.includes(index)));
}

// Функція для обчислення сум у стовпцях матриці
function getColSumsInMatrix(matrix) :any[] { Show usages Цимбалюк Руслан Іванович
    const sumsArray :any[] = [];
    const rows = matrix.length;
    const cols = matrix[0].length;

    for (let j :number = 0; j < cols; j++) {
        let sum :number = 0;
        for (let i :number = 0; i < rows; i++) {
            sum += matrix[i][j]; // Додавання значення
        }
        sumsArray.push(sum);
    }
    return sumsArray;
}

// Функція для отримання суми елементів у стовпцях без від'ємних елементів
function getSumArrayWithoutNegativeElementsInCols(matrix) :[] { Show usages Цимбалюк Руслан Іванович
    const colsToRemove :any[] = [];
    const rows = matrix.length;
    const cols = matrix[0].length;

```

		Цимбалюк Р.І.			ДУ «Житомирська політехніка».24.121.29.001 – Лр3	Арк.
		Дмитренко І.А.				10
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    for (let j :number = 0; j < cols; j++) {
        for (let i :number = 0; i < rows; i++) {
            if (matrix[i][j] < 0) {
                colsToRemove.push(j);
                break; // Переривання, якщо знайдено від'ємний елемент
            }
        }
    }

    const updatedMatrix = removeColsFromMatrix(matrix, colsToRemove);
    return getColSumsInMatrix(updatedMatrix);
}

console.log("Сума елементів у стовпцях без від'ємних елементів: ", getSumArrayWithoutNegativeElementsInCols(matrix));

// Функція для отримання суми елементів у стовпцях з від'ємними елементами
function getSumArrayWithNegativeElementsInCols(matrix) :[] { Show usages Цимбалюк Руслан Іванович
    const colsToRemove :any[] = [];
    const rows = matrix.length;
    const cols = matrix[0].length;

    for (let j :number = 0; j < cols; j++) {
        let isIncludeNegative :boolean = false;

        for (let i :number = 0; i < rows; i++) {
            if (matrix[i][j] < 0) {
                isIncludeNegative = true;
                break; // Переривання, якщо знайдено від'ємний елемент
            }
        }

        if (!isIncludeNegative) {
            colsToRemove.push(j);
        }
    }

    const updatedMatrix = removeColsFromMatrix(matrix, colsToRemove);
    return getColSumsInMatrix(updatedMatrix);
}

console.log("Сума елементів у стовпцях з від'ємними елементами: ", getSumArrayWithNegativeElementsInCols(matrix));

// Транспонування матриці
let transposedMatrix :(...):[] = matrix[0].map((_:number, colIndex :number) :number[] => matrix.map(row :number[] => row[colIndex]));
console.log("Транспонована матриця:", transposedMatrix);

// Функція для обчислення максимальних сум елементів діагоналей, паралельних головній діагоналі
function calculateMaxParallelMainDiagonalsSum(matrix) :number { Show usages Цимбалюк Руслан Іванович
    let maxParallelMainDiagonalsSum :number = Number.NEGATIVE_INFINITY;

```

		Цимбалюк Р.І.			ДУ «Житомирська політехніка».24.121.29.001 – Лр3	Арк.
		Дмитренко І.А.				11
Змн.	Арк.	№ докум.	Підпис	Дата		

```

// Обробка діагоналей вище головної
for (let k :number = 1; k < matrix[0].length; k++) {
    let parallelOverMainDiagonalSum :number = 0;
    for (let i :number = 0; i < matrix.length && i + k < matrix[0].length; i++) {
        parallelOverMainDiagonalSum += matrix[i][i + k];
    }
    maxParallelMainDiagonalsSum = Math.max(maxParallelMainDiagonalsSum, parallelOverMainDiagonalSum);
}

// Обробка діагоналей нижче головної
for (let k :number = 1; k < matrix.length; k++) {
    let parallelUnderMainDiagonalSum :number = 0;
    for (let i :number = 0; i < matrix[0].length && i + k < matrix.length; i++) {
        parallelUnderMainDiagonalSum += matrix[i + k][i];
    }
    maxParallelMainDiagonalsSum = Math.max(maxParallelMainDiagonalsSum, parallelUnderMainDiagonalSum);
}

return maxParallelMainDiagonalsSum;
}

console.log("Максимум серед сум елементів діагоналей, паралельних головній діагоналі матриці = " +
    calculateMaxParallelMainDiagonalsSum(matrix));

// Функція для обчислення мінімальних сум модулів елементів діагоналей, паралельних побічній діагоналі
function calculateMinAbsParallelAntiDiagonalsSum(matrix) :number { Show usages Цимбалиук Руслан Іванович
    let minAbsParallelAntiDiagonalsSum :number = Number.POSITIVE_INFINITY;

    // Обробка діагоналей вище побічної
    for (let k :number = 1; k < matrix[0].length; k++) {
        let parallelOverAntiDiagonalSum :number = 0;
        for (let i :number = 0; i < matrix.length && i + k < matrix[0].length; i++) {
            let j :number = matrix[0].length - 1 - (i + k);
            parallelOverAntiDiagonalSum += Math.abs(matrix[i][j]);
        }
        minAbsParallelAntiDiagonalsSum = Math.min(minAbsParallelAntiDiagonalsSum, parallelOverAntiDiagonalSum);
    }

    // Обробка діагоналей нижче побічної
    for (let k :number = 1; k < matrix.length; k++) {
        let parallelUnderAntiDiagonalSum :number = 0;
        for (let i :number = 0; i < matrix[0].length && i + k < matrix.length; i++) {
            let j :number = matrix[0].length - 1 - i;
            parallelUnderAntiDiagonalSum += Math.abs(matrix[i + k][j]);
        }
        minAbsParallelAntiDiagonalsSum = Math.min(minAbsParallelAntiDiagonalsSum, parallelUnderAntiDiagonalSum);
    }

    return minAbsParallelAntiDiagonalsSum;
}

console.log("Мінімум серед сум модулів елементів діагоналей, паралельних побічній діагоналі матриці = " +
    calculateMinAbsParallelAntiDiagonalsSum(matrix));

```

		Цимбалиук Р.І.			ДУ «Житомирська політехніка».24.121.29.001 – Лр3	Арк.
		Дмитренко І.А.				12
Змн.	Арк.	№ докум.	Підпис	Дата		

Результат виконання програми:

```

▶ Removing unpermitted intrinsics lockdown-install.js:1
Матриця: ▶ Array(5) script3.js:11
Додатні елементи: 18 script3.js:17
Кількість рядків, які не містять нуль: 2 script3.js:22
Кількість стовпців, які містять нуль: 3 script3.js:41
Індекс рядка з найдовшою серією: 3 script3.js:74
Добуток елементів у рядках без від'ємних елементів: ▶ Array(3) script3.js:81
Сума елементів у стовпцях без від'ємних елементів: ▶ Array(2) script3.js:123
Сума елементів у стовпцях з від'ємними елементами: ▶ Array(3) script3.js:150
Транспонована матриця: ▶ Array(5) script3.js:154
Максимум серед сум елементів діагоналей, паралельних головній
діагоналі матриці = 28 script3.js:181
Мінімум серед сум модулів елементів діагоналей, паралельних побічній
діагоналі матриці = 3 script3.js:211
>

```

Рис.3. Результат використання програми

Висновки: Під час виконання лабораторної роботи №3, я опрацював завдання з використанням масивів у JS.

Посилання на Git: https://git.ztu.edu.ua/ipz/2023-2027/ipz-23-4/tsimbalyuck-ruslan/frontend/front_lab3.git

		Цимбалюк Р.І.			ДУ «Житомирська політехніка».24.121.29.001 – Лр3	Арк.
		Дмитренко І.А.				13
Змн.	Арк.	№ докум.	Підпис	Дата		