**Державний вищий навчальний заклад**

**Ужгородський національний університет**

**Факультет інформаційних технологій**

**ЛАБОРАТОРНА РОБОТА № 8**

**Тема:** “JavaScript ч. 1”

Виконав студент І курсу

спеціальності «Інженерія

програмного забезпечення»

Романюк Артем Романович

**Ужгород-2024**

***Мета:*** поглибити знання з скриптової мови **JavaScript**.

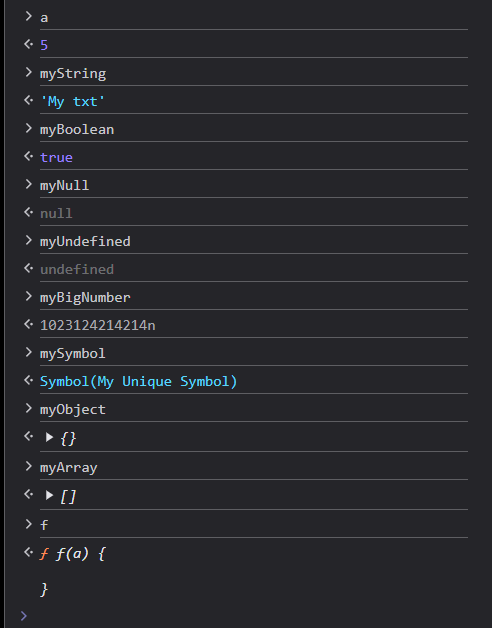
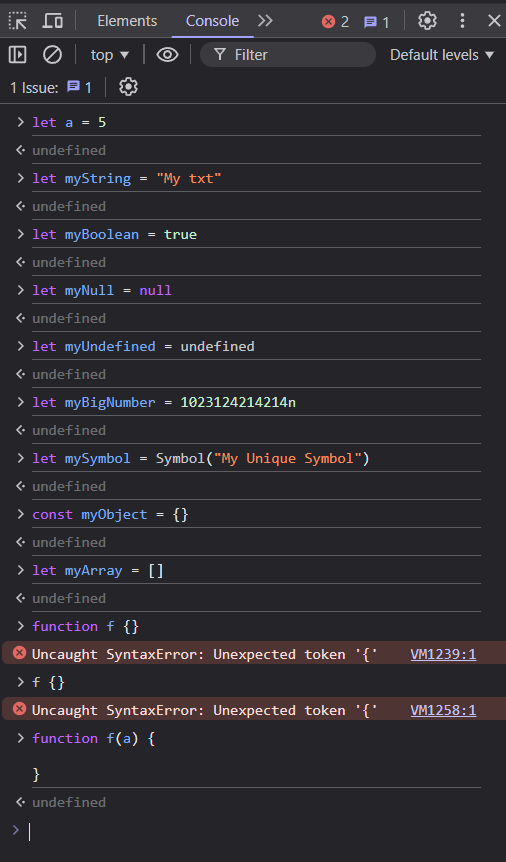
***Завдання до роботи:***

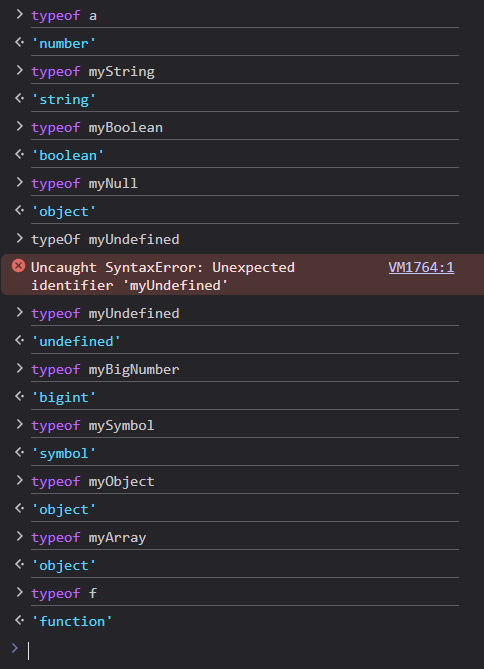
1. Виконати домашні завдання описані у теоретичних матеріалах до лекцій 1,2,3 та LEC\_ARRAYS
2. Оформити звіт за взірцем, де представити результати код, скріншоти виконання, пояснення коду чи вирішення проблеми (якщо таке потрібно)

**Хід роботи:**

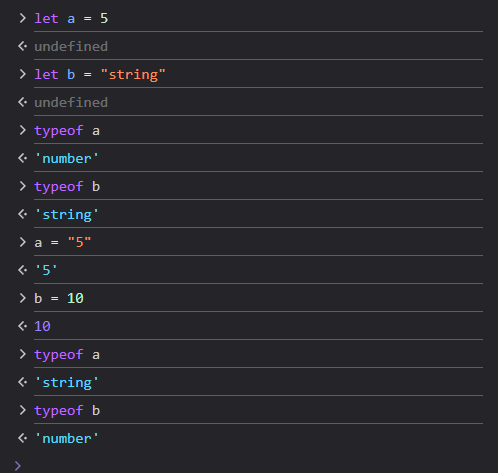
**1. Завдання до лекції №1:**

*1.1. Вивести всі види змінних з їх типами. Як мінімум мають бути об'єкт, масив, рядок, число, функція. //var varString…*





*1.2. Для рядка і числа поміняти значення так, щоб їх типи помінялися.*



*1.3. Порівняти між собою рядок і число з "однаковими" значеннями.*



*1.4. Простими способами зашифрувати / дешифрувати числову інформацію*



**2. Завдання до лекції №2:**

*2.1. Записати код в консолі браузера/NodeJS. Яким буде результат виконання?*

var foo = 1;

function bar() {

if (!foo) {

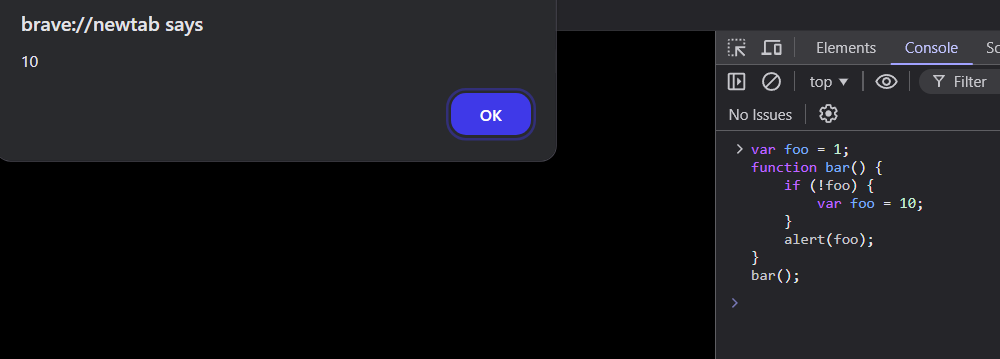
var foo = 10;

}

alert(foo);

}

bar();



*2.2. Обгрунтуйте будь-ласка результат виконання вищезгаданого коду.*

Так як **функція bar()** не приймає значення **foo**, **if** буде завжди виконуватись. Тому задається нова локальна змінна **var foo = 10**. Згодом вона ж алертиться, бо **alert** знаходиться всередині **функції**

*2.3. Обгрунтуйте будь-ласка результат виконання коду.*

var a = 1;

function b() {

a = 10;

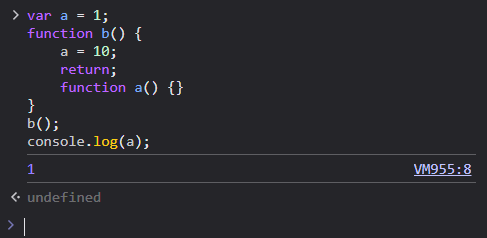
return;

function a() {}

}

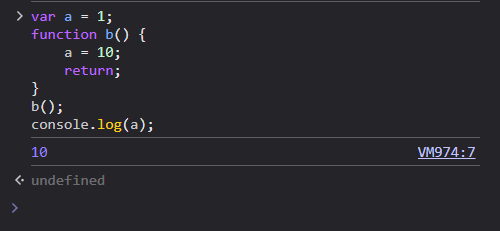
b();

console.log(a);



Через те, що в **функції b** спочатку ініціалізується **функція а**, а потім ця функція перетворюється в локальну змінну-число **а = 10**, console.log(a) виводить глобальну змінну **a**, яка ніяк не змінилась при виклику **функції b**

*2.4. Якщо результатом виконання коду буде число відмінне від 10, виправити помилку. Виправити мінімальною кількістю операцій.*



Щоб виправити, потрібно прибрати ініціалізацію **функції а**. Тоді в **функції b** **глобальна змінна а** набуде нового значення 10.

**3. Завдання до лекції №3:**

*3.1. Наведіть приклади з використанням унарних, бінарних, тернарних операторів (якомога більше різних операторів)*

**Унарні оператори**

Унарні оператори застосовуються до одного операнда.

* **+** - перетворює значення на число.
* **-** - змінює знак числа.
* **++**  - збільшує значення на 1.
* **--**  - зменшує значення на 1.
* **typeof** - визначає тип даних.
* **!**  - логічне заперечення.
* **void**  - завжди повертає undefined.
* **delete**  - виконує видалення об’єкта, властивості об’єкта, або ж елемента масива з заданим індексом.

**Бінарні оператори**

Бінарні оператори застосовуються до двох операндів.

* **Арифметичні оператори**: +, -, \*, /, %, \*\*
* **Порівняння**: ==, !=, ===, !==, >, <, >=, <=
* **Логічні оператори**: &&, ||
* **Побітові оператори**: &, |, ^, ~, <<, >>, >>>
* **Оператори присвоєння**: =, +=, -=, \*=, /=, %=, \*\*=

**Тернарний оператор**

Тернарний оператор — це скорочена форма умовного виразу **if-else**. Він має вигляд:

**умова ? вираз1 : вираз2;**

**condition? expression if true : expression if false;**

Якщо **умова** істинна, повертається **вираз1**, інакше — **вираз2**.

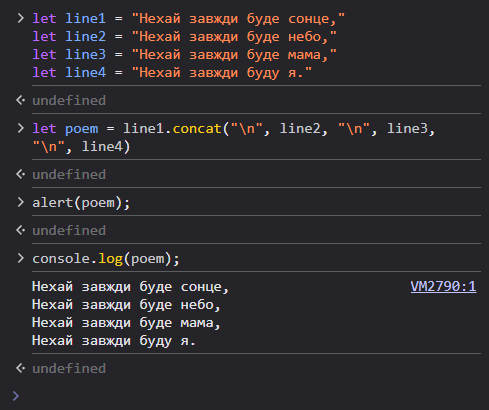
*3.2. Написати скрипт використовуючи метод concat() на тему:*

Нехай завжди буде сонце,

Нехай завжди буде небо,

Нехай завжди буде мама,

Нехай завжди буду я.



**4. Завдання до лекції : LEC\_ARRAYS**

*4.1. Знайти неперервний підмасив масиву , сума елементів якого є максимальною.*

function maxSubArray(arr) { //Алгоритм Кадане

let maxSum = arr[0];

let currentSum = arr[0];

let subArray = arr[0];

let startIndex = 0;

let endIndex = 0;

let tempStartIndex = 0;

for (let i = 1; i < arr.length; i++) {

if (arr[i] > currentSum + arr[i]) {

tempStartIndex = i;

currentSum = arr[i];

} else {

currentSum += arr[i];

}

if (currentSum > maxSum) {

maxSum = currentSum;

startIndex = tempStartIndex;

endIndex = i;

}

}

subArray = arr.slice(startIndex, endIndex + 1)

return {

maxSum,

subArray

}

}

const arr1 = [214, 55, 0, 1, 3, -2];

const arr2 = [8, -2, 5, -8];

const arr3 = [-4, -1, 4];

let result = maxSubArray(arr1);

console.log("Максимальна сума:", result.maxSum);

console.log("Неперерваний підмасив із максимальною сумою:", result.subArray);

console.log();

result = maxSubArray(arr2);

console.log("Максимальна сума:", result.maxSum);

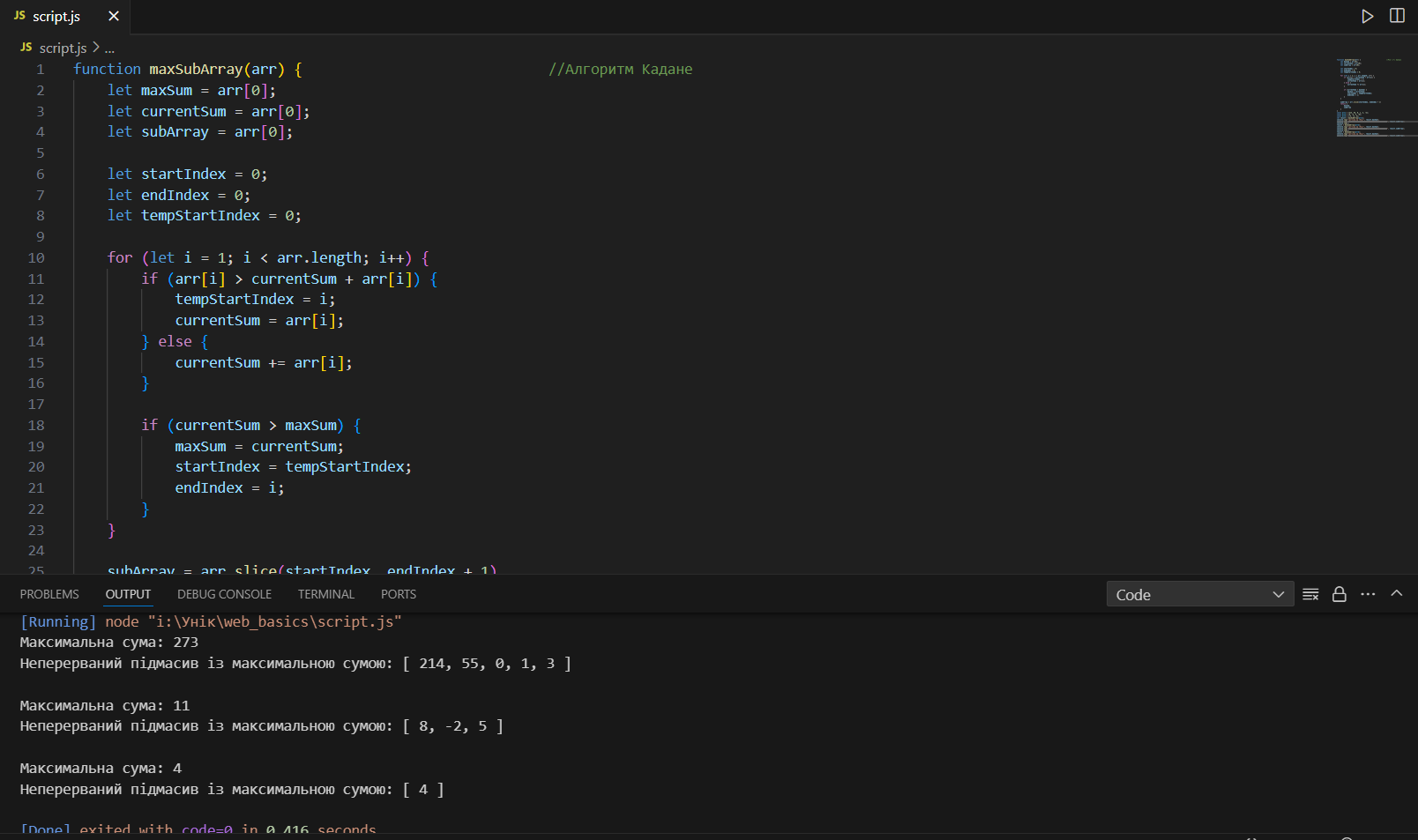
console.log("Неперерваний підмасив із максимальною сумою:", result.subArray);

console.log();

result = maxSubArray(arr3);

console.log("Максимальна сума:", result.maxSum);

console.log("Неперерваний підмасив із максимальною сумою:", result.subArray);

**

*4.2. Створити функцію сумування двох дуже довгих чисел представлених строкою.*

function sumBigNums(num1, num2) {

let carry = 0;

let result = [];

let maxLenghtOfNums = Math.max(num1.length, num2.length);

num1 = num1.padStart(maxLenghtOfNums, "0");

num2 = num2.padStart(maxLenghtOfNums, "0");

for (let i = maxLenghtOfNums - 1; i >= 0; i--) {

let sum = parseInt(num1[i]) + parseInt(num2[i]) + carry;

carry = Math.floor(sum / 10);

result.unshift(sum % 10);

}

if (carry > 0) {

result.unshift(carry);

}

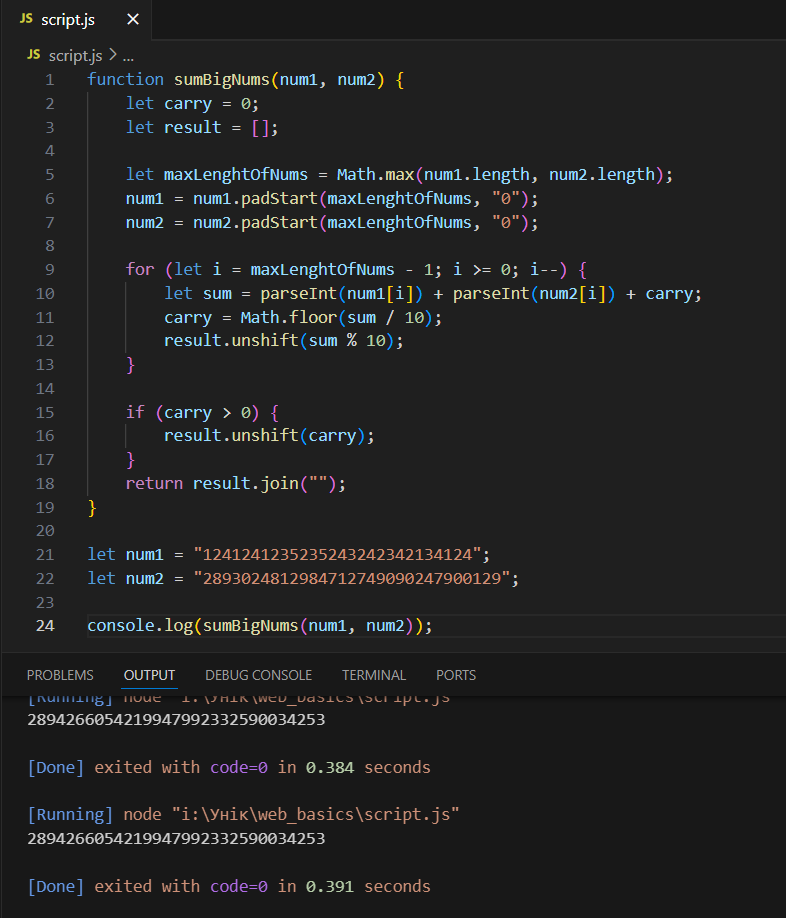
return result.join("");

}

let num1 = "1241241235235243242342134124";

let num2 = "2893024812984712749090247900129";

console.log(sumBigNums(num1, num2));

**

*4.3. Написати метод реалізуючий різницю двох масивів. Причому різниця має враховувати кількість однакових елементів.*

function arrayDifference(arr1, arr2) {

const countsElements = (arr) => {

return arr.reduce((count, item) => {

count[item] = (count[item] || 0) + 1;

return count;

}, {});

};

const count1 = countsElements(arr1);

const count2 = countsElements(arr2);

const result = [];

for (let item in count1) {

const remaining = count1[item] - (count2[item] || 0);

if (remaining > 0) {

result.push(...Array(remaining).fill(item));

}

}

return result;

}

let arr1 = [1, 2, 3, 3, 4, 5];

let arr2 = [1, 3, 5];

console.log(arrayDifference(arr1, arr2));

**

**Висновок:**

В ході даної лабораторної роботи було поглиблено знання про скриптову мову JavaScript. Ми закріпили навички з роботи з різними типами змінних, навчилися виконувати порівняння між ними та застосовувати методи шифрування та дешифрування числових даних. Також ми розглянули особливості функцій та локальних змінних у JavaScript, зокрема, як відбувається обробка змінних у межах функцій. Використання унарних, бінарних та тернарних операторів дозволило поглибити розуміння операцій, а також ознайомитися з методами для роботи з масивами та рядками, включаючи алгоритми для пошуку підмасивів із максимальною сумою, операції над великими числами та різницю між масивами.