**РЕСПУБЛИКА БЕЛАРУСЬ**

**КОМИТЕТ ПО ОБРАЗОВАНИЮ МИНГОРИСПОЛКОМА**

учреждение образования

МИНСКИЙ ГОСУДАРСТВЕННЫЙ КОЛЛЕДЖ ЦИФРОВЫХ ТЕХНОЛОГИЙ

##### Группа 73 ТП

## Отчет

## по лабораторной работе № 11

**"Использование** **функций при процедурном подходе в программировании на языке JavaScript"**

учебный предмет

“Программные средства создания Интернет приложений”

**Исполнитель: А.А. Артюшевский**

**Руководитель: Е.П. Сушкевич**

**Минск, 2025**

**Цель работы:**

Сформировать умение использовать функции в сценариях на языке JavaScript.

**Задание 1: Факториал числа**

Напишите функцию, которая вычисляет факториал заданного числа. Функция должна использовать рекурсию. Реализуйте проверку на корректность ввода (например, чтобы число было неотрицательным).

**Задание 2: Строка-палиндром**

Напишите функцию, которая проверяет, является ли заданная строка палиндромом. Учтите регистр и пробелы.

**Задание 3: Сортировка массива**

Создайте функцию, которая сортирует массив чисел по возрастанию. Используйте алгоритм сортировки пузырьком.

**Задание 4: Поиск уникальных элементов**

Напишите функцию, которая принимает массив и возвращает новый массив, содержащий только уникальные элементы.

**Задание 5: Генерация случайного пароля**

Создайте функцию, которая генерирует случайный пароль заданной длины. Пароль должен содержать буквы верхнего и нижнего регистра, цифры и специальные символы.

**Задание 6: Фибоначчи**

Напишите функцию, которая возвращает n-ное число Фибоначчи. Используйте как рекурсивный, так и итеративный подход.

**Задание 7: Сложение массивов**

Создайте функцию, которая принимает два массива и возвращает новый массив, содержащий сумму соответствующих элементов этих массивов.

**Задание 8: Фильтрация массива**

Напишите функцию, которая принимает массив чисел и фильтрует его, оставляя только четные числа.

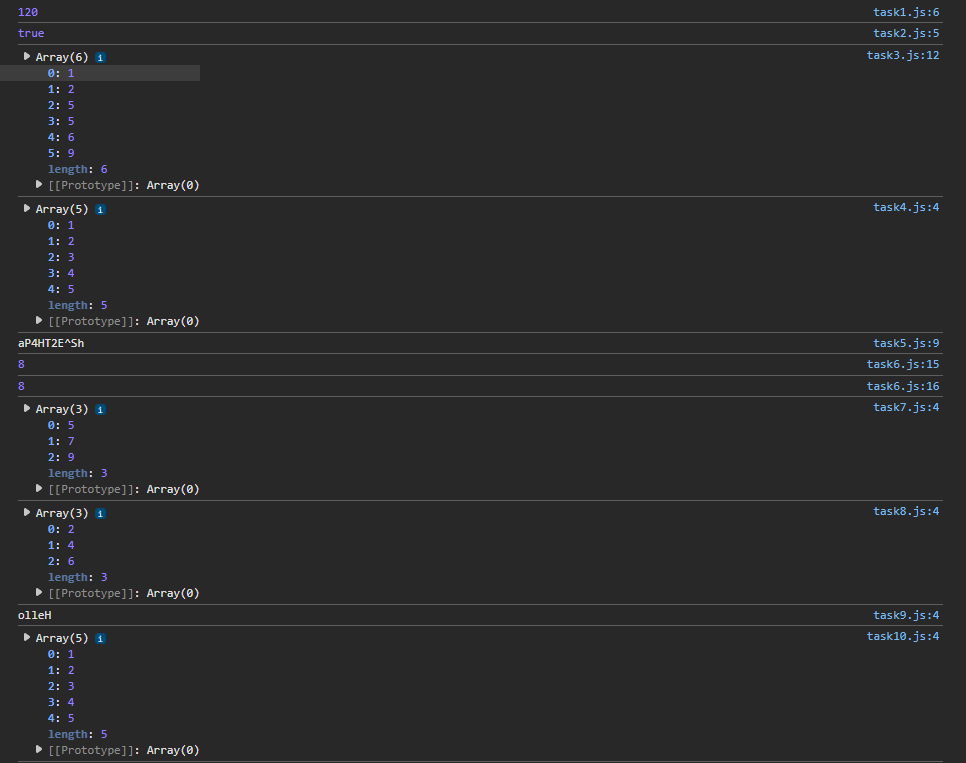
**Задание 9: Перевернуть строку**

Создайте функцию, которая принимает строку и возвращает её в перевернутом виде.

**Задание 10: Удаление дубликатов из массива**

Напишите функцию, которая принимает массив и удаляет из него все дубликаты, сохраняя порядок элементов.

Решение:



index.html:

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

    <script src="/scripts/task1.js"></script>

    <script src="/scripts/task2.js"></script>

    <script src="/scripts/task3.js"></script>

    <script src="/scripts/task4.js"></script>

    <script src="/scripts/task5.js"></script>

    <script src="/scripts/task6.js"></script>

    <script src="/scripts/task7.js"></script>

    <script src="/scripts/task8.js"></script>

    <script src="/scripts/task9.js"></script>

    <script src="/scripts/task10.js"></script>

    <script src="/scripts/task11.js"></script>

</body>

</html>

task1.js:

function factorial(n) {

if (n < 0) return "Введите неотрицательное число!";

if (n === 0 || n === 1) return 1;

return n \* factorial(n - 1);

}

console.log(factorial(5)); // Выведет 120

task2.js:

function isPalindrome(str) {

let cleanStr = str.toLowerCase().replace(/\s/g, '');

return cleanStr === cleanStr.split('').reverse().join('');

}

console.log(isPalindrome("А роза упала на лапу Азора")); // true

task3.js:

function bubbleSort(arr) {

let len = arr.length;

for (let i = 0; i < len - 1; i++) {

for (let j = 0; j < len - 1 - i; j++) {

if (arr[j] > arr[j + 1]) {

[arr[j], arr[j + 1]] = [arr[j + 1], arr[j]];

}

}

}

return arr;

}

console.log(bubbleSort([5, 2, 9, 1, 5, 6]));

task4.js:

function uniqueElements(arr) {

return [...new Set(arr)];

}

console.log(uniqueElements([1, 2, 3, 3, 4, 5, 5]));

task5.js:

function generatePassword(length) {

let chars = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789!@#$%^&\*()";

let password = "";

for (let i = 0; i < length; i++) {

password += chars.charAt(Math.floor(Math.random() \* chars.length));

}

return password;

}

console.log(generatePassword(10));

task6.js:

function fibonacciRecursive(n) {

if (n <= 1) return n;

return fibonacciRecursive(n - 1) + fibonacciRecursive(n - 2);

}

function fibonacciIterative(n) {

let a = 0, b = 1, temp;

for (let i = 2; i <= n; i++) {

temp = a + b;

a = b;

b = temp;

}

return b;

}

console.log(fibonacciRecursive(6)); // 8

console.log(fibonacciIterative(6)); // 8

task7.js:

function sumArrays(arr1, arr2) {

return arr1.map((num, index) => num + (arr2[index] || 0));

}

console.log(sumArrays([1, 2, 3], [4, 5, 6])); // [5, 7, 9]

task8.js:

function filterEvenNumbers(arr) {

return arr.filter(num => num % 2 === 0);

}

console.log(filterEvenNumbers([1, 2, 3, 4, 5, 6]));

task9.js:

function reverseString(str) {

return str.split('').reverse().join('');

}

console.log(reverseString("Hello")); // "olleH"

task10.js:

function removeDuplicates(arr) {

return arr.filter((item, index) => arr.indexOf(item) === index);

}

console.log(removeDuplicates([1, 2, 2, 3, 4, 4, 5]));

Контрольные вопрсоы:

1. Что такое функция в JavaScript?

Ответ:

Функция — это блок кода, который выполняет определенную задачу и может быть вызван многократно.

2. Каковы основные способы определения функций в JavaScript?

Ответ:

В JavaScript есть несколько способов объявить функцию:

* Обычная (Function Declaration)
* Функциональное выражение (Function Expression)
* Стрелочная функция (Arrow Function)
* Анонимная функция (чаще используется в обработчиках событий)

3. Что такое параметры функции и как они могут использоваться?

Ответ:

* Параметры — это входные данные, которые передаются в функцию
* Можно задавать параметры по умолчанию
* Можно передавать неограниченное количество параметров с arguments

4. Что происходит, если функция не возвращает значение?

Ответ:

Если в функции нет return, она возвращает undefined

5. Объясните разницу между глобальной и локальной областью видимости.

Ответ:

* **Глобальная область видимости** — переменные доступны **везде** в коде
* **Локальная область** — переменная существует **только внутри функции**

6. Что такое замыкание и как оно работает?

Ответ:

Замыкание — это функция, которая **запоминает** переменные из своей области видимости, даже если она вызывается за ее пределами.