МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образование «Полоцкий государственный университет»

Факультет информационных технологий

Кафедра технологий программирования

**Лабораторная работа №10**

по дисциплине: «Веб-технологии»

на тему: «Составление и отладка программ на языке JavaScript.»

Вариант 10

|  |  |
| --- | --- |
| Выполнил | Студент группы 22-ИТ-1  Короткевич П.А. |
|  |  |
| Проверил | Макарычева В.А |

Полоцк 2024

**Цель:** Изучить принцип отладки программ на JavaScript, научиться применять полученные знания на практике.

**Задание 1**

Создайте функцию которая будет принимать первым аргументом массив чисел, перебирать его, выводя на каждой итерации в консоль сообщение содержащее в себе текст Элемент N, где N - элемент массива на текущей итерации. Если элемент массива равен вашему варианту, то вместо console.log следует использовать console.error c тем же содержимым.

Литсинг 1

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Array Iteration</title>

</head>

<body>

<script>

function iterateArray(arr) {

const variant = 10;

arr.forEach((element, index) => {

if (element === variant) {

console.error(`Элемент ${element}`);

} else {

console.log(`Элемент ${element}`);

}

});

}

const myArray = [5, 10, 7, 3, 10, 8];

iterateArray(myArray);

</script>

</body>

</html>

Этот HTML-код представляет собой веб-страницу с заголовком "Array Iteration". Он включает в себя скрипт JavaScript, который содержит функцию iterateArray, принимающую массив в качестве аргумента.

Внутри функции iterateArray определена константа variant, которая равна числу 10. Затем функция перебирает каждый элемент переданного массива с помощью метода forEach. На каждой итерации проверяется, равен ли текущий элемент значению константы variant. Если да, то выводится сообщение об ошибке в консоль с помощью console.error, содержащее текст "Элемент N", где N - значение элемента. Если элемент не равен variant, то сообщение выводится с помощью console.log.

После определения функции iterateArray, создается массив myArray, содержащий числа. Этот массив передается в качестве аргумента функции iterateArray, что вызывает ее выполнение. В результате вызова функции выводятся сообщения в консоль для каждого элемента массива в формате "Элемент N", где N - значение элемента. Если элемент равен 10, сообщение выводится через console.error, в противном случае - через console.log.

На следующем рисунке представлена реализация:



Рисунок 1 – Реализация первого задания

**Задание 2**

Напишите функцию, предлагающую пользователю ввести строку (используйте метод prompt). Если пользователь ввел строку Дата, то отобразите в консоль сообщение содержащее текущую дату (для получения текущей даты используйте new Date()), иначе – следует вывести в консоль саму строку и предложить пользователю снова ввести строку(предлагать до тех пор, пока пользователь не введет строку Дата.

Листинг 2

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Проверка ввода</title>

</head>

<body>

<script>

function checkInput() {

while (true) {

const userInput = prompt("Введите строку:");

if (userInput === "Дата") {

const currentDate = new Date();

console.log(`Текущая дата: ${currentDate}`);

break;

} else {

console.log(`Вы ввели: ${userInput}`);

}

}

}

checkInput();

</script>

</body>

</html>

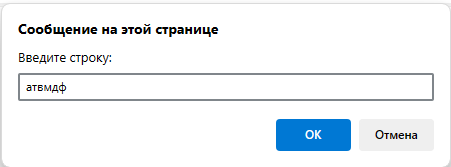
Этот HTML-код представляет собой веб-страницу с заголовком "Проверка ввода". Код включает в себя скрипт JavaScript, который содержит функцию checkInput. При загрузке страницы вызывается функция checkInput(). Внутри функции запускается бесконечный цикл while(true), который будет выполняться до тех пор, пока не будет прерван с помощью оператора break.

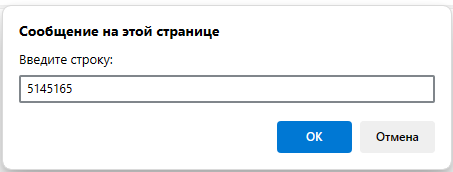
На каждой итерации цикла вызывается функция prompt, которая выводит диалоговое окно с запросом ввода строки пользователем. Введенное пользователем значение сохраняется в переменной userInput.

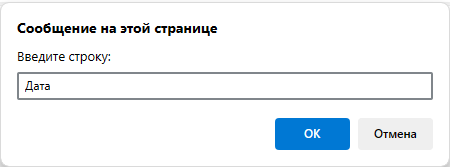
Затем проверяется, равна ли введенная пользователем строка "Дата". Если да, то создается новый объект Date с текущей датой и временем. Значение текущей даты выводится в консоль с помощью console.log.

Если введенная строка не равна "Дата", то выводится сообщение в консоль, содержащее текст "Вы ввели:" и введенное пользователем значение.

На следующем рисунке представлена реализация:







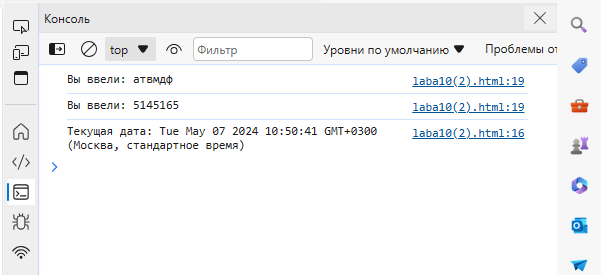


Рисунок 2 – Реализация второго задания

**Задание 3**

Напишите функцию, которая будет принимать первым аргументом число N, создавать массив длиной N, где каждый элемент массива равен его индексу, и выводить поочередно все элементы массива через 3 секунды после вызова(для задержки используйте setTimeout).

Литсинг 3

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Вывод массива с задержкой</title>

</head>

<body>

<script>

function printArrayWithDelay(N) {

const array = new Array(N).fill().map((\_, index) => index);

array.forEach((element, index) => {

setTimeout(() => {

console.log(element);

}, index \* 3000);

});

}

printArrayWithDelay(10);

</script>

</body>

</html>

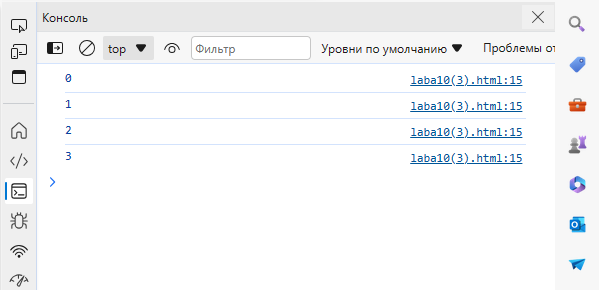
Этот HTML-код представляет собой веб-страницу с заголовком "Вывод массива с задержкой". Внутри тега <script> находится JavaScript код.

Когда страница загружается, вызывается функция printArrayWithDelay, которая принимает аргумент N. Внутри этой функции создается массив array длиной N, заполненный значениями от 0 до N-1.

Затем для каждого элемента массива вызывается функция setTimeout. Эта функция используется для установки задержки перед выполнением определенной операции. Каждый элемент массива выводится в консоль через определенное время после вызова функции printArrayWithDelay.

Задержка между выводом элементов массива увеличивается на 3 секунды с каждой итерацией. Это достигается путем умножения индекса элемента на 3000 миллисекунд (так как 1 секунда = 1000 миллисекунд).

На следующем рисунке представлена реализация:



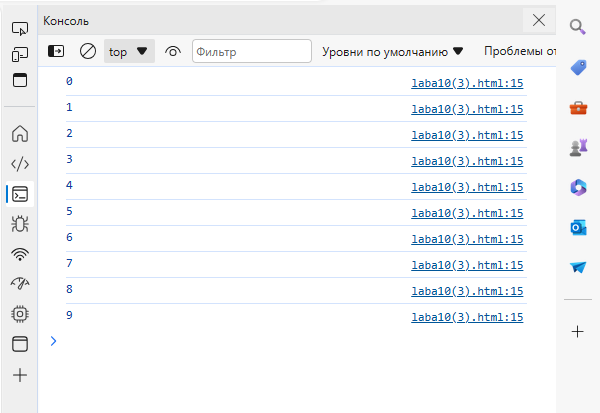


Рисунок 3 – Реализация третьего задания

**Вывод:** Изучение принципов отладки программ на JavaScript и их практическое применение являются важной частью процесса обучения этому языку программирования. Понимание методов отладки позволяет программистам обнаруживать и исправлять ошибки в своем коде более эффективно. В ходе изучения этой темы мы освоили различные методы отладки JavaScript-программ, такие как использование инструментов браузера для анализа и отладки кода, включая консоль разработчика, точки останова (breakpoints) и отслеживание выполнения кода строка за строкой.