Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет информационных технологий и управления

Кафедра информационных технологий автоматизированных систем

Отчет

по лабораторной работе №2

«Работа с *JavaScript*»

Вариант №10

|  |  |
| --- | --- |
| Выполнил:  ст. гр. 820603  Дрозд В.А. | Проверил:  А. Л. Гончаревич |

Минск 2021

**СОДЕРЖАНИЕ**

Введение 3

1 Постановка задачи 4

2 Аналитический обзор литературы 5

3 Ход работы 8

Заключение 17

## **ВВЕДЕНИЕ**

*JavaScript* – [мультипарадигменный](https://ru.wikipedia.org/wiki/%D0%9C%D1%83%D0%BB%D1%8C%D1%82%D0%B8%D0%BF%D0%B0%D1%80%D0%B0%D0%B4%D0%B8%D0%B3%D0%BC%D0%B5%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) [язык программирования](https://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F). Поддерживает [объектно-ориентированный](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5), [императивный](https://ru.wikipedia.org/wiki/%D0%98%D0%BC%D0%BF%D0%B5%D1%80%D0%B0%D1%82%D0%B8%D0%B2%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) и [функциональный](https://ru.wikipedia.org/wiki/%D0%A4%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) стили.

*JavaScript* обычно используется как встраиваемый язык для программного доступа к объектам [приложений](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0). Наиболее широкое применение находит в [браузерах](https://ru.wikipedia.org/wiki/%D0%91%D1%80%D0%B0%D1%83%D0%B7%D0%B5%D1%80) как язык сценариев для придания [интерактивности](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%B0%D0%BA%D1%82%D0%B8%D0%B2%D0%BD%D0%BE%D1%81%D1%82%D1%8C) [веб-страницам](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D1%82%D1%80%D0%B0%D0%BD%D0%B8%D1%86%D0%B0). В браузере для *JavaScript* доступно всё, что связано с манипулированием веб-страницами, взаимодействием с пользователем и веб-сервером. Язык *JS* – это то, что «оживляет» разметку страниц и пользовательский функционал сайтов. С помощью этого языка реализуется возможность реакции страницы или отдельных ее элементов на действия посетителя. *JavaScript* полностью совместим с операционными системами *Windows*, *Linux*, *Mac* *OS*, а также всеми популярными мобильными платформами.

# 1 ПОСТАНОВКА ЗАДАЧИ

В данной лабораторной работе необходимо изучить базовые возможности языка *JavaScript*.

Научиться использовать переменные, массивы, методы языка программирования. Полученные знания использовать для написания калькулятора с поддержкой нескольких операций. Создать веб-страницы, демонстрирующие базовую функциональность языка *JavaScript*.

# 2 АНАЛИТИЧЕСКИЙ ОБЗОР ЛИТЕРАТУРЫ

*JavaScript* – [мультипарадигменный](https://ru.wikipedia.org/wiki/%D0%9C%D1%83%D0%BB%D1%8C%D1%82%D0%B8%D0%BF%D0%B0%D1%80%D0%B0%D0%B4%D0%B8%D0%B3%D0%BC%D0%B5%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) [язык программирования](https://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F). Поддерживает [объектно-ориентированный](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5), [императивный](https://ru.wikipedia.org/wiki/%D0%98%D0%BC%D0%BF%D0%B5%D1%80%D0%B0%D1%82%D0%B8%D0%B2%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) и [функциональный](https://ru.wikipedia.org/wiki/%D0%A4%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) стили. Является реализацией спецификации [*ECMAScript*](https://ru.wikipedia.org/wiki/ECMAScript).

*JavaScript* обычно используется как встраиваемый язык для программного доступа к объектам [приложений](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0). Наиболее широкое применение находит в [браузерах](https://ru.wikipedia.org/wiki/%D0%91%D1%80%D0%B0%D1%83%D0%B7%D0%B5%D1%80) как язык сценариев для придания [интерактивности](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%B0%D0%BA%D1%82%D0%B8%D0%B2%D0%BD%D0%BE%D1%81%D1%82%D1%8C) [веб-страницам](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D1%82%D1%80%D0%B0%D0%BD%D0%B8%D1%86%D0%B0).

Для добавления кода на страницу можно использовать теги <*script*></*script*>, которые рекомендуется, но не обязательно, помещать внутри контейнера <*head*>. Контейнеров <*script*> в одном документе может быть сколько угодно. Есть и другая возможность подключения *JavaScript* – написать скрипт в отдельном файле, а потом подключить его с помощью атрибута *src*.

Для пояснения кода *JavaScript* добавляются комментарии. Однострочные комментарии начинаются с //, многострочные комментарии начинаются с /\* и заканчиваются \*/.

В языке *JavaScript* предусмотрены следующие типы литералов: целого типа, вещественного, логического типа и строковые литералы. Литералы часто называют данными. Данные хранятся в переменных. К объектам относятся данные основных системных типов, например, числовые значения и массивы. Свойствами являются значения, принадлежащие объектам. Методы выполняют некоторые манипуляции над данными. Выражения – это манипуляции над литералами, переменными, свойствами и методами, в результате вычисления которых, так же как при использовании методов, получается единственное значение. Эти комбинации создаются с помощью операций (или знаков операций). Операторы – это некоторая синтаксически законченная конструкция языка, аналогичная предложению в русском языке, состоящая из последовательности всех вышеперечисленных элементов. И, наконец, программа представляет собой последовательность операторов.

Функции – ключевая концепция в *JavaScript*. Функция в *JavaScript* –специальный тип объектов, позволяющий формализовать средствами языка определённую логику поведения и обработки данных. Любая функция – это объект, и, следовательно, ею можно манипулировать как объектом, в частности.

Рассмотрим основные методы языка *JavaScript*.

*Alert* (предупреждение) – служит для вывода информации.

*Confirm* – предназначен для вывода информации и позволяет пользователю сделать выбор в форме ответа да/нет на вопрос.

*Prompt* – служит для вывода информации и позволяет пользователю ввести ответ с клавиатуры.

Оператор *typeof* позволяет нам увидеть, какой тип данных сохранён в переменной.

События – это действия или случаи, возникающие в программируемой вами системе, о которых система сообщает вам для того, чтобы пользователи могли с ними взаимодействовать. *JavaScript* позволяет при обнаружении нужного события выполнить необходимый код. Для этого у *HTML* элементов есть специальные атрибуты – обработчики событий, в которые и можно добавить *JavaScript* код.

Рассмотрим основные события.

*Onclick* – пользователь кликнул мышкой на *HTML* элемент.

*Onload* – браузер закончил загружать страницу.

*Onmouseover* – пользователь навел мышку на *HTML* элемент.

Для создания переменной в *JavaScript* используется ключевое слово *let* и устаревшее *var*. Объявление *const* задаёт константу, то есть переменную, которую нельзя менять.

В *JavaScript* любые текстовые данные являются строками. Строки начинаются и оканчиваются одинарными, либо двойными кавычками. При вставке кавычек в середину строки, она будет урезана. Чтобы решить эту проблему, в *JavaScript* предусмотрены специальные символы. Приведем некоторые из них.

Перенос строки – \*n.*

Двойная кавычка – \".

Одинарная кавычка – \'.

Табуляция – \*t*.

Возврат на одну позицию – *\b.*

Переход на следующую страницу – *\f.*

Возврат каретки *– \r.*

Одинарная кавычка – *\’.*

Обратная косая черта – *\\.*

Массивы представляют собой отдельные объекты, которые содержат несколько значений, хранящихся в списке. Объекты массива могут храниться в переменных и обрабатываться во многом так же, как и любой другой тип значения. Элементы массива нумеруются, начиная с нуля. Можно получить доступ к элементу по его порядковому номеру. *JavaScript* предоставляет методы для работы с массивами. Рассмотрим самые популярные методы.

*Pop()* – извлекает элемент из конца.

*Push*() – добавляет элементы в конец массива.

*Shift()* – извлекает элемент из начала.

*Unshift*() – добавляет элементы в начало.

Объект *Math* предоставляет ряд математических функций, которые можно использовать при вычислениях. Рассмотрим основные из них.

*Math*.*min*() и *Math*.*max*() – возвращают соответственно минимальное и максимальное значение из набора чисел.

Кроме методов, объект *Math* также определяет набор встроенных констант, которые можно использовать в различных вычислениях.

# ХОД РАБОТЫ

* 1. **Создание окон сообщений**

Требуется показать вывод и ввод информации в виде разнообразных окон сообщений с помощью методов *alert*(), *confirm*() и *prompt*(). Были добавлены три кнопки, при нажатии на которые будут вызываться методы. Ниже приведен код методов.

*const showAlert = () => alert("Салют Влад и Тимур");*

*const showPrompt = () => {*

*const name = prompt("Как вас зовут?", "Дрозд Владислав Андреевич");*

*document.querySelector('#name').textContent = name;*

*};*

*const showConfirm = () => {*

*const result = confirm("Это confirm-подтверждение. Сделайте выбор: yes/no");*

*/\* Я, студент Владислав изменил, то что было в тексте окна подтверждения*

*(да/нет) на yes/no \*/*

*alert(result);*

*};*

Внешний вид кнопок показан на рисунке 1.



Рисунок 1 – Кнопки вызова методов

Результат вызова функций показан на рисунках 2-4.

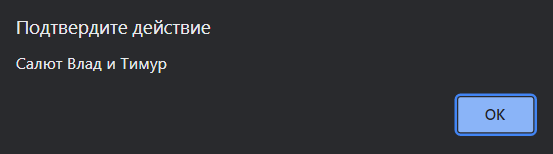


Рисунок 2 – Метод *alert*()

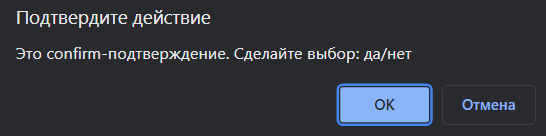


Рисунок 3 – Метод *confirm*()

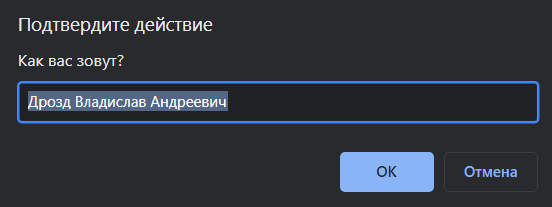


Рисунок 4 – Метод *prompt*()

* 1. **Использование переменных при вызове методов**

В следующем задании изменил пример, показанный в пункте 3.1, добавив переменные, содержащие информацию об авторе. Использовались эти переменные в методе *alert*(). Ниже приведен фрагмент программного кода.

*// Объявление переменных*

*const name = "Влад";*

*const year = "2000";*

*const city = "Минск";*

*// Вывод на экран*

*alert(`Меня зовут ${name}, я ${year} года рождения, проживаю в городе ${city}`);*

Результат вывода информации показан на рисунке 5.

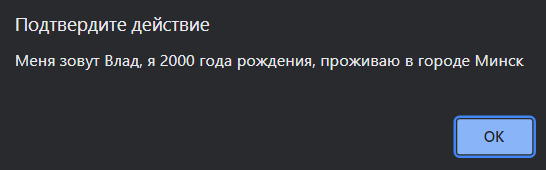
**

Рисунок 5 – Окно вывода информации

* 1. **Переопределение переменной в методе prompt**

В следующем примере изменил пример, показанный в пункте 3.2, добавив ввод переменных о себе с помощью метода prompt, а затем вывел, введенную информацию с помощью метода alert.

*// Ввод переменных с помощью prompt*

*const name = prompt('Введите ваше имя', 'Влад');*

*const year = prompt('Введите год рождения', '1990');*

*const city = prompt('Введите город проживания', 'Москва');*

*// Вывод на экран*

*alert(`Меня зовут ${name}, я ${year} года рождения, проживаю в городе ${city}`);*

На рисунке 6 показан пример окна ввода информации.

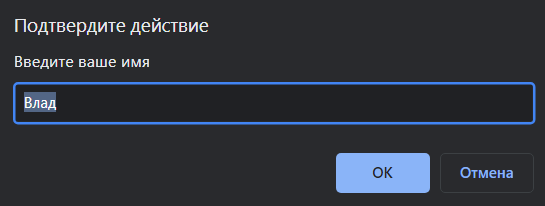


Рисунок 6 – Окно ввода имени

На рисунке 7 показан текст, который будет выведен после ввода всей информации.

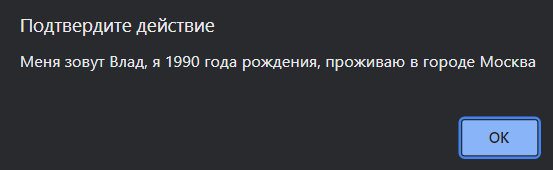


Рисунок 7 – Вывод нового текста

* 1. **Освоение приемов работы с массивами**

Был изменен предыдущий пример. Для этого создал массив, содержащий номер и названия месяцев, а также массив, содержащий информацию о каждом месяце, которую будем выводить в окне сообщения.

Код программы:

*const onPrompt = () => {*

*// Обновляем значение в переменной*

*const selectetLanguage = prompt('Выберите язык (ru/en)', 'ru');*

*const months = new Array(12).fill(0).map((\_, i) => {*

*return new Date(`${i + 1}/1`).toLocaleDateString(selectetLanguage, { month: 'long' })*

*});*

*alert(months);*

*}*

*onPrompt();*

При открытии страницы будет предложено ввести язык, на котором будет выведен список месяцев. Пример ввода показан на рисунке 8.

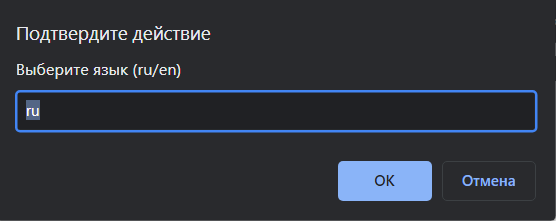
**

Рисунок 8 – Ввод языка.

На рисунке 9 показан вывод месяцев на экраны.

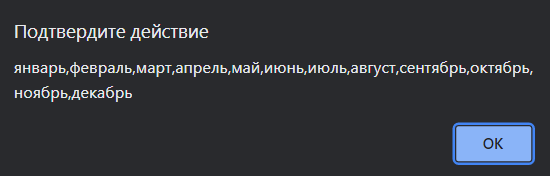


Рисунок 9 – Сведения о месяце

* 1. ***Escape*-последовательности**

В следующем задании приведен пример использования *escape*-последовательностей. Для этого, при загрузке страницы, показываем модальное окно с примерами для каждой последовательности.

Код программы:

*const sequences = [*

*['\b', '\\b'],*

*['\t', '\\t'],*

*['\f', '\\f'],*

*['\n', '\\n'],*

*['\r', '\\r'],*

*['\"', '\\"'],*

*['\'', '\\\''],*

*['\\', '\\\\'],*

*];*

*alert(*

*sequences.map(*

*sequence => `${sequence[1]} = ${sequence[0]}`*

*).join('\n')*

*);*

На рисунке 10 показано применение escape-последовательностей.

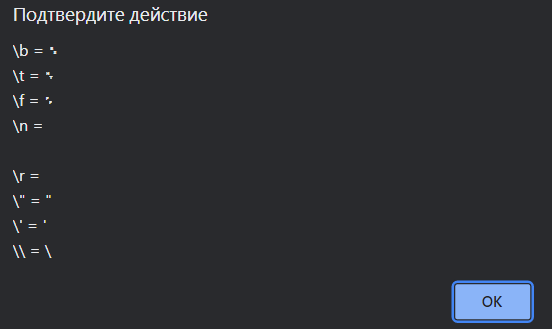


Рисунок 10 – Применение специальных символов

* 1. **Создание калькулятора**

В задании 4.1 требовалось написать калькулятор, используя тернарную операцию, который поддерживал бы также постфиксные и префиксные операции.

Код программы:

*let operator = operatorInput.value;*

*const setOperation = () => {*

*operator = operatorInput.value;*

*if (operator === '++' || operator === '--') {*

*numInput2.style.display = 'none';*

*} else {*

*numInput2.style.display = 'block';*

*}*

*}*

*const calc = () => {*

*let num1 = +numInput1.value;*

*let num2 = +numInput2.value;*

*switch(operator) {*

*case '+':*

*alert(num1 + num2);*

*break;*

*case '-':*

*alert(num1 - num2);*

*break;*

*case '\*':*

*alert(num1 \* num2);*

*break;*

*case '/':*

*alert(num1 / num2);*

*break;*

*case '%':*

*alert(num1 % num2);*

*break;*

*case '++':*

*alert(++num1);*

*break;*

*case '--':*

*alert(--num1);*

*break;*

*}*

*}*

В зависимости от введенной операции (арифметической или тернарной) будет показано два или одно поле для ввода чисел соответственно. После заполнения пользователем всех полей и нажатия кнопки «*Calculate*» пользователь получит результат в виде модального окна. Поля ввода операций и чисел показаны на рисунке 11.

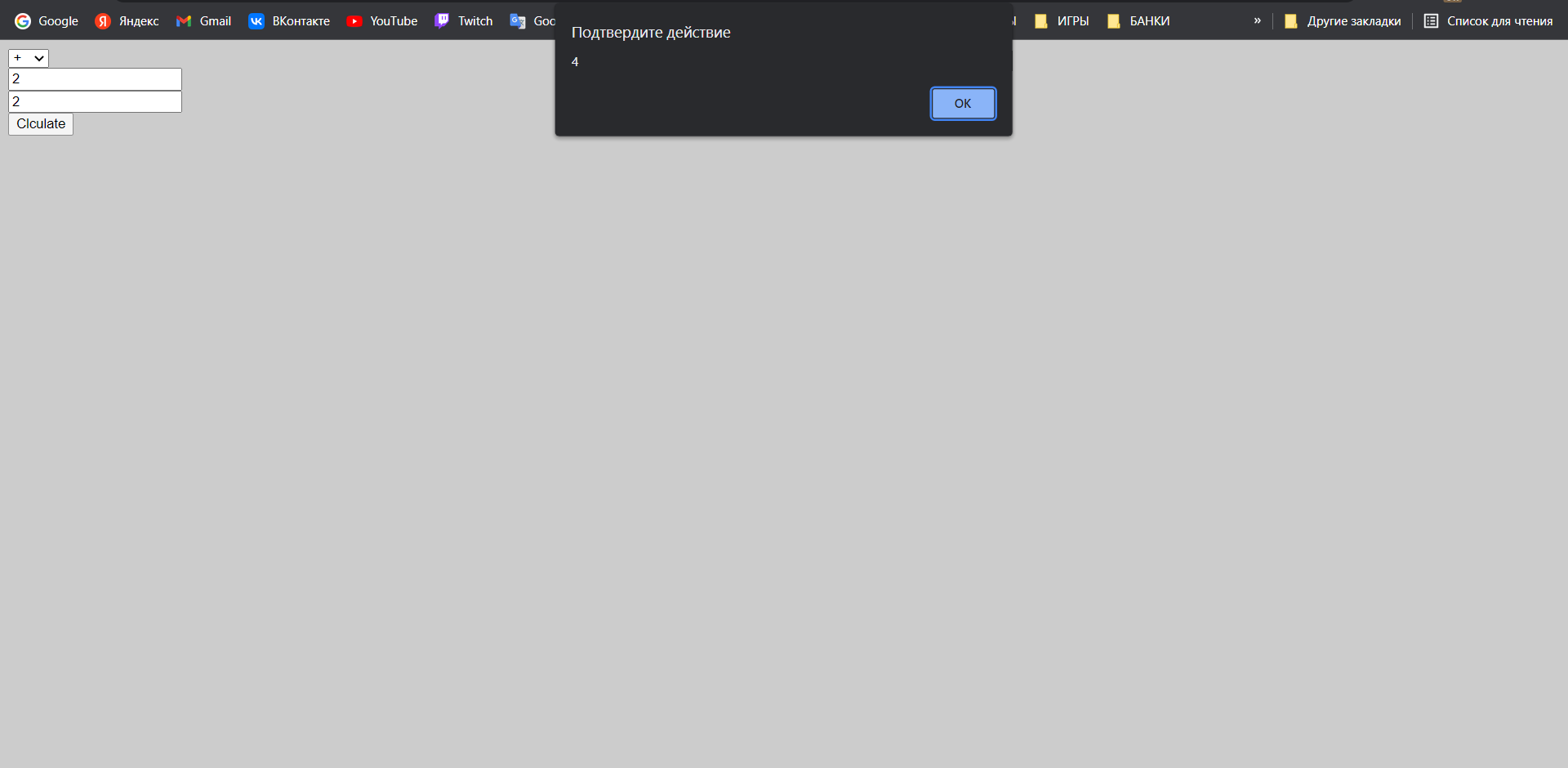


Рисунок 11 – Страница с калькулятором

# 

# ЗАКЛЮЧЕНИЕ

В ходе выполнения данной лабораторной работы мною были изучены основы работы с языком программирования *JavaScript*. Были реализованы примеры использования *JavaScript* для создания веб-страниц. В ходе работы были применены основные методы языка: *prompt*, *alert*, *confirm* и другие.