

**Министерство Образования и исследований**

**Государственный Университет Молдовы**

**Кафедра информатики**

## **Отчет:**

**По дисциплине Java Script**

**Лабораторная работа №1**

**Тема: Введение в JavaScript**

**Выполнил:**

**Чубару Максим, gr. IA2503**

**Проверил:**

**N.Nartea lector universitar**

**Кишинев, 2025**

# Теоретическая часть

JavaScript — это интерпретируемый язык программирования, предназначенный для создания интерактивных элементов веб-страниц. Он выполняется на стороне клиента в браузере, а также может использоваться в серверной среде (например, с помощью платформы Node.js). JavaScript позволяет работать с пользовательским вводом, управлять содержимым страницы, выполнять вычисления, а также реализовывать логику принятия решений с помощью условий и циклов.

## Формулировка задачи

Изучить основы языка JavaScript, освоить способы выполнения кода в браузере и через внешний файл, а также научиться работать с переменными, типами данных, условными операторами и циклами.

## Описание цели и основные этапы работы

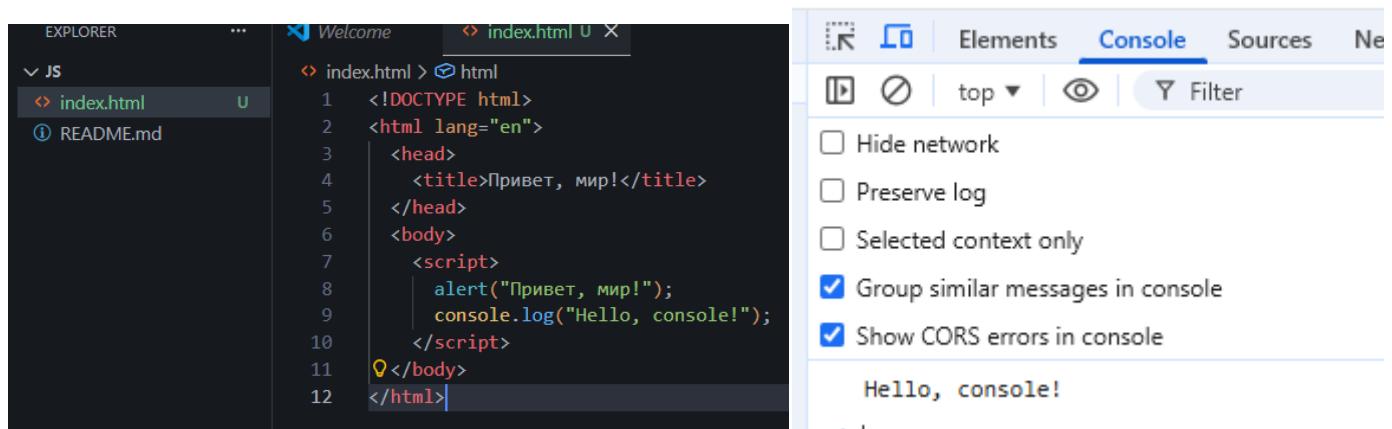
Целью лабораторной работы является формирование базовых навыков работы с JavaScript и понимание принципов его выполнения в браузере и локальной среде.

Основные этапы работы:

1. Подготовка среды разработки (установка редактора кода и Node.js).
2. Выполнение JavaScript-кода в консоли браузера.
3. Создание HTML-страницы со встроенным JavaScript-кодом.
4. Подключение внешнего JavaScript-файла.
5. Работа с переменными и типами данных.
6. Использование условных операторов и циклов.

## Практическая часть

В ходе практической части была создана HTML-страница `index.html`, в которую сначала был встроен JavaScript-код с использованием функций `alert()` и `console.log()`.



```
EXPLORER      ...
JS
index.html    u
README.md

Welcome      index.html ✘
index.html > html
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  <title>Привет, мир!</title>
5  </head>
6  <body>
7  <script>
8  | alert("Привет, мир!");
9  | console.log("Hello, console!");
10 | </script>
11 </body>
12 </html>
```

Console

Elements Sources Network

top Filter

Hide network

Preserve log

Selected context only

Group similar messages in console

Show CORS errors in console

Hello, console!

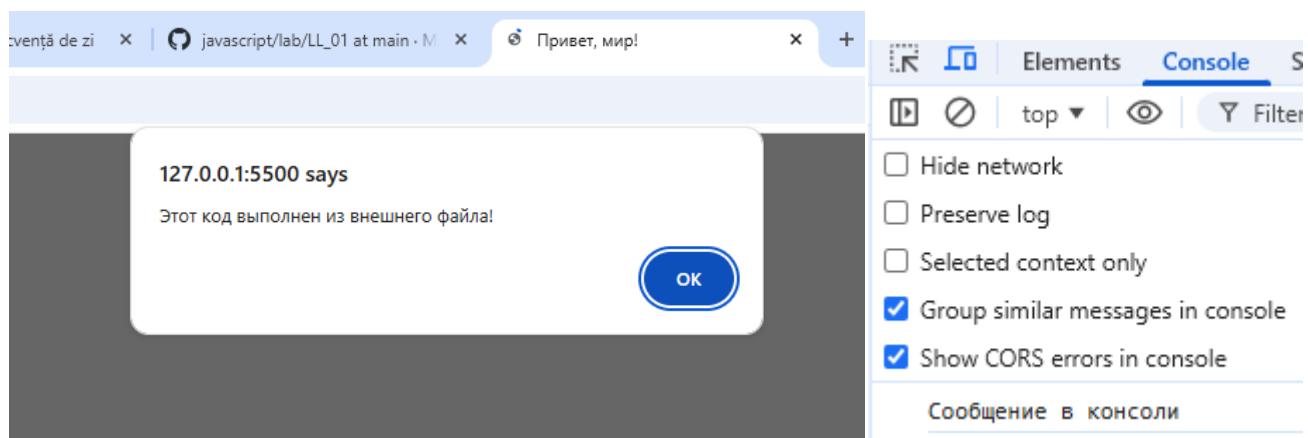


Затем код был вынесен во внешний файл `script.js` и подключён к HTML-документу с помощью тега `<script>`.

The screenshot shows a code editor with two files:

- `index.html` contains the HTML structure with a `<script src="script.js">` tag.
- `script.js` contains the JavaScript code: `1 alert("Этот код выполнен из внешнего файла!");` and `2 console.log("Сообщение в консоли");`

```
index.html > html > head > script
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <title>Привет, мир!</title>
5     <script src="script.js"></script>
6   </head>
7   <body>
8     <script>
9       alert("Привет, мир!");
10      console.log("Hello, console!");
11    </script>
12  </body>
13 </html>
```



Также в файле `script.js` были объявлены переменные различных типов данных (строка, число, логический тип),

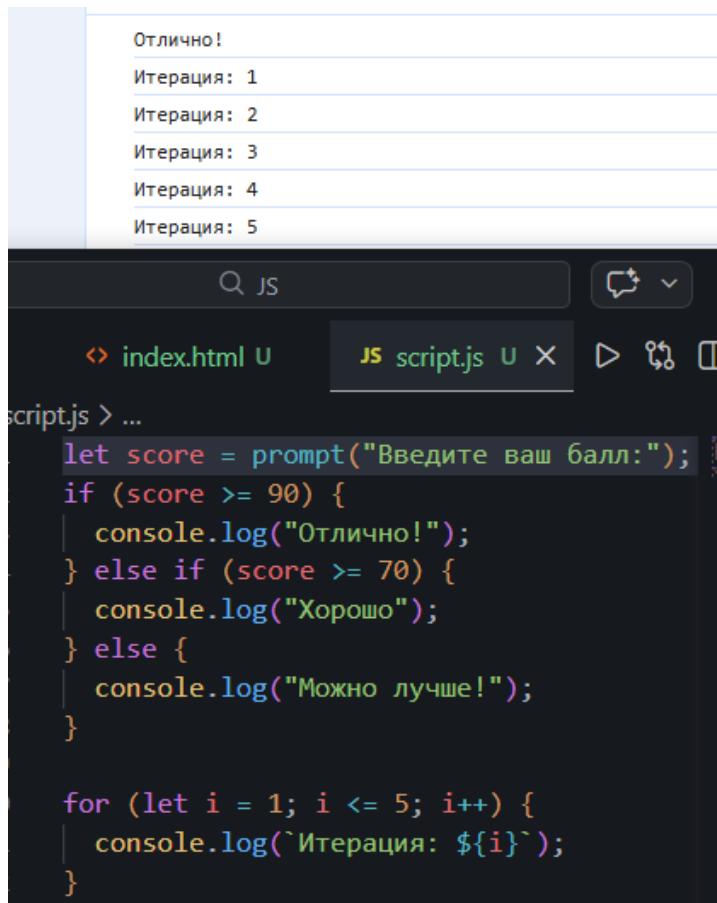
The screenshot shows a code editor with `script.js` containing:

```
1 console.log("Введите свои данные:\n");
2
3 let Name = "Максим";
4 let birthYear = 2006;
5 let isStudent = true;
```

The developer console shows the output of these log statements:

- "Введите свои данные:"
- "Имя: Максим"
- "Год рождения: 2006"
- "Студент: true"

реализована проверка условий с использованием конструкции `if-else if-else`, а также организован цикл `for` для многоократного вывода информации в консоль.



The screenshot shows the DevTools console tab for a file named 'script.js'. The console output displays five iterations of the script's execution:

```
Отлично!
Итерация: 1
Итерация: 2
Итерация: 3
Итерация: 4
Итерация: 5
```

The code itself is as follows:

```
let score = prompt("Введите ваш балл:");
if (score >= 90) {
    console.log("Отлично!");
} else if (score >= 70) {
    console.log("Хорошо");
} else {
    console.log("Можно лучше!");
}

for (let i = 1; i <= 5; i++) {
    console.log(`Итерация: ${i}`);
}
```

## Краткое описание особенностей реализации

Особенностью реализации является использование встроенных средств браузера (DevTools) для выполнения и отладки JavaScript-кода. Для вывода сообщений применялась функция `console.log()`, что позволяет отслеживать выполнение программы без вмешательства в интерфейс страницы.

Пользовательский ввод реализован с помощью функции `prompt()`, которая возвращает строку и автоматически участвует в неявном преобразовании типов при сравнении значений.

## Вывод и ссылка на репозиторий Git

В ходе выполнения лабораторной работы были изучены основы языка JavaScript, освоены способы выполнения кода в браузере и через внешний файл, а также получены практические навыки работы с переменными, условиями и циклами. Поставленная цель работы была достигнута.

Ссылка на репозиторий Git:

<https://github.com/maximcv246-a11y/JS-Lab1>

# Ответы на контрольные вопросы

## 1. Чем отличается `var` от `let` и `const`?

`var` имеет функциональную область видимости и допускает повторное объявление переменной.  
`let` имеет блочную область видимости и может изменяться, но не может быть объявлен повторно в том же блоке.

`const` также имеет блочную область видимости, но значение переменной нельзя изменить после инициализации.

## 2. Что такое неявное преобразование типов в JavaScript?

Неявное преобразование типов — это автоматическое приведение одного типа данных к другому при выполнении операций, например при сравнении строки и числа.

## 3. Как работает оператор `==` в сравнении с `===`?

Оператор `==` сравнивает значения с учётом неявного преобразования типов, а оператор `===` сравнивает и значение, и тип данных, не выполняя преобразования.

---

## Список использованных источников

1. Официальная документация JavaScript — <https://developer.mozilla.org>
2. JavaScript.info — <https://javascript.info>
3. Учебные материалы по дисциплине «Веб-программирование» - [https://github.com/MSU-Courses/javascript\\_typescript](https://github.com/MSU-Courses/javascript_typescript)