Молдавский Государственный Университет

Факультет Математики и Информатики

Департамент Информатики

Отчет по лабораторной работе № 1

по предмету „JavaScript”

Выполнил: студент группы IA2403

Божик Дениc

Проверил преподаватель:

Murinet A., Dr., Conf.univ.

Кишинев, 2024

**Цель работы**

Познакомиться с основами JavaScript, научиться писать и выполнять код в браузере и в локальной среде, разобраться с базовыми конструкциями языка.

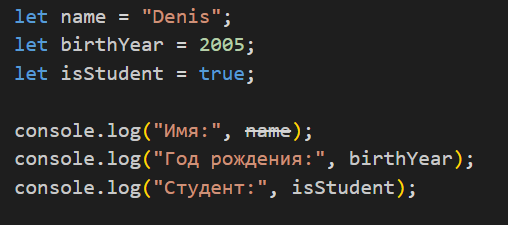


Стандартный код для html документа в который подключаем:

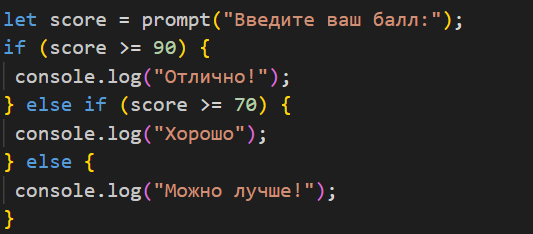
<script src="script.js"></script> - ссылка на js файл со скриптом

alert("Привет, мир!") – выводит сообщение с текстом

console.log("Hello, console!") – выводит в консоле

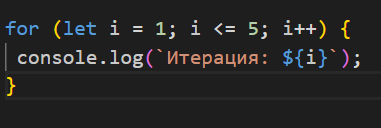


Создаем переменные в который вводим имя, год рождения и указываем true если мы студент. Затем выводим на в консоль все переменные.



Создаем переменную в которую записываем значения.

С условных опереаторов if и else if вывдится : отлично, хорошо или можно лучше.



Цикл for

Цикл выполняется **5 раз**, начиная с i = 1 и заканчивая i = 5.

На каждой итерации в консоль выводится текущий номер итерации:

**Контрольные вопросы:**

**Чем отличается var от let и const?**

**Различия между var, let и const**

1. **var** – устаревший способ объявления переменных:
   * Доступен во всей функции, даже до объявления (hoisting).
   * Может быть повторно объявлен в той же области видимости.
   * Не имеет блочной области видимости.
2. **let** – современный способ объявления переменных:
   * Доступен только после объявления.
   * Нельзя повторно объявить в той же области.
   * Имеет блочную область видимости {}.
3. **const** – для констант:
   * Нельзя изменить после присваивания.
   * Также имеет блочную область видимости.
   * Однако можно изменять свойства объектов, объявленных через const.

**Что такое неявное преобразование типов в JavaScript?**

Неявное преобразование (coercion) — это автоматическое изменение типа переменной в зависимости от контекста.

**Как работает оператор == в сравнении с ===?**

**азница между == и ===**

* **== (нестрогое равенство) – сравнивает значения, но приводит их к одному типу:**

**console.log(5 == "5"); // true (строка "5" приводится к числу)**

**console.log(false == 0); // true (false превращается в 0)**

* **=== (строгое равенство) – сравнивает значения и типы:**

**console.log(5 === "5"); // false (число не равно строке)**

**console.log(false === 0); // false (разные типы)**

**Вывод:  
Лабораторная работа продемонстрировала базовые конструкции JavaScript: получение данных от пользователя, условия и циклы. Также важно учитывать особенности работы с типами данных для предотвращения логических ошибок.**