Министерство образования Республики Беларусь  
Учреждение образования  
«Белорусский государственный университет информатики и радиоэлектроники» Филиал  
«Минский радиотехнический колледж»

Учебная дисциплина «Программные средства создания Internet-приложений»

**Отчет**

по выполнению лабораторной работы  
«Создание HTML-страниц с использованием сценариев JavaScript. Использование управляющих конструкций в JavaScript»

Выполнил Гончаров П.В.

Проверила Терешко О.И.

Минск

2020

Лабораторная работа № 17

Номер учебной группы: 7K2492.

Дата выполнения работы: 10.11.2020

Тема работы: «Создание HTML-страниц с использованием сценариев JavaScript. Использование управляющих конструкций в JavaScript»

Цель работы: Формирование умений использования управляющих конструкций при написании сценариев JavaScript..

Результат выполнения работы:

Index.html

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

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

  <title>Document</title>

  <script src="script.js"></script>

</head>

<body>

  <button onclick="enterNumbers()">Ввод чисел</button>

  <button onclick="add()">Сложить</button>

  <button onclick="multiply()">Умножить</button>

</body>

</html>

Script.js

let [number1, number2] = [0, 0];

enterName();

function enterName() {

  const result = prompt("Введи имя:");

  if (result) {

    alert(`Привет, ${result}.`);

    askAge1(result);

  } else {

    alert("Такого имени не существует.");

  }

}

function askAge1(name) {

  const age = prompt("Возраст:");

  if (!age) {

    return;

  }

  let askMessage = "";

  let outputMessage = "";

  if (age >= 2 && age <= 5) {

    askMessage = "Сад:"

    outputMessage = "Сад ";

  }

  if (age >= 6 && age <= 16) {

    askMessage = "Школа:"

    outputMessage = "Школа ";

  }

  if (age >= 17 && age <= 21) {

    askMessage = "Учебное заведение:"

    outputMessage = "Учебное заведение ";

  }

  if (age >= 22 && age <= 65) {

    askMessage = "Место работы:"

    outputMessage = "Место работы ";

  }

  if (age > 66) {

    outputMessage = "На заслуженном отдыхе"

  }

  if (askMessage) {

    const result = prompt(askMessage);

    if (!result) {

      return;

    }

    alert(`${name}; ${outputMessage} ${result}`);

  }

}

function enterNumbers() {

  number1 = prompt("Первое число:");

  number2 = prompt("Второе число:");

}

function multiply() {

  alert(`${number1 \* number2}`);

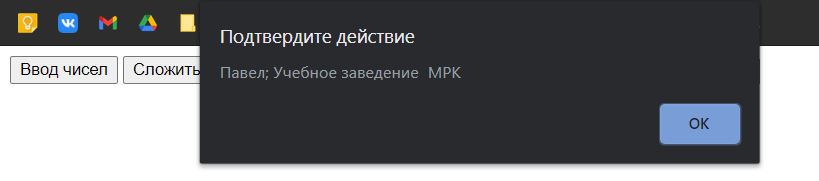
}

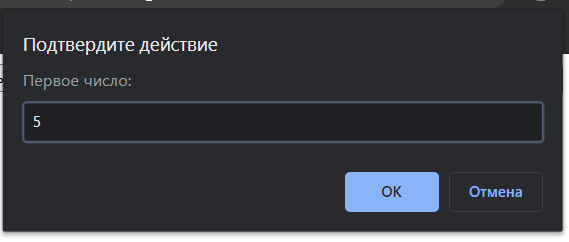
function add() {

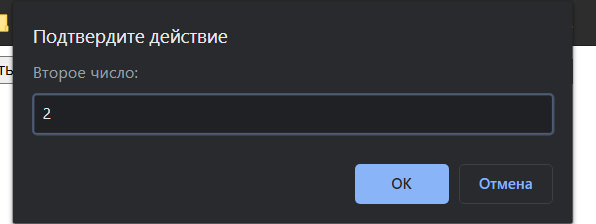
  alert(`${Number(number1) + Number(number2)}`);

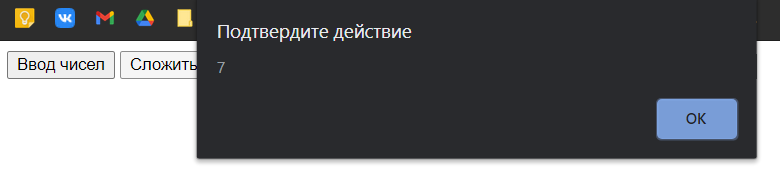
}

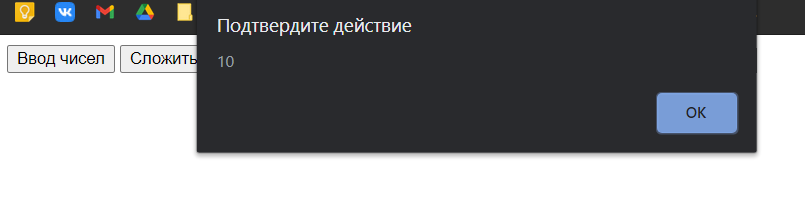
Cкриншоты:











Контрольные вопросы:

1. Какие управляющие конструкции поддерживает JavaScript?

Язык JavaScript поддерживает стандартный набор управляющих конструкций:

* последовательная структура.
* условные операторы: if (структура с единичным выбором); if/else (с двойным

выбором), else if... (с несколькими ветвями), тернарный оператор (?:), switch (с

* множественным выбором). Условные операторы - это операторы языка JavaScript

(ECMAScript), которые в зависимости от некоторого условия позволяют выполнить одно или несколько определённых инструкций.

* пять типов структур повторения: while, do/while, for, for/in, for...of.

1. Перечислите условные операторы, используемые в JavaScript.

Формы условных операторов в JavaScript:

* условный оператор if (с одной ветвью);
* условный оператор if...else (с двумя ветвями);
* условный оператор else if... (с несколькими ветвями);
* тернарный оператор (?:);
* оператор выбора switch.

1. Приведите пример использования структур повторения, используемые в JavaScript.

Пять типов структур повторения: while, do/while, for, for/in, for...of.

Для похожих повторяющихся действий, посчитать арифметическую прогрессию.

1. Охарактеризуйте типы данных JavaScript.

В JavaScript имеется восемь основных типов данных:

1) number для любых чисел: целочисленных или чисел с плавающей точкой,

целочисленные значения ограничены диапазоном ±253.

2) bigint для целых чисел произвольной длины.

3) string для строк. Строка может содержать один или больше символов, нет отдельного

символьного типа.

4) boolean для true/false.

5) null для неизвестных значений – отдельный тип, имеющий одно значение null.

6) undefined для неприсвоенных значений – отдельный тип, имеющий одно

значение undefined.

7) object для более сложных структур данных.

8) symbol для уникальных идентификаторов.