|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | |  |  |  | | МИНОБРНАУКИ РОССИИ | | | | Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | | | |
| Институт информационных технологий (ИТ) | |
| Кафедра инструментального и прикладного программного обеспечения (ИиППО) | |

|  |  |  |  |
| --- | --- | --- | --- |
| **ОТЧЕТ ПО ПРАКТИЧЕСКИМ РАБОТАМ** | | | |
| **по дисциплине** | | | |
| **«Разработка клиентских частей интернет-ресурсов»** | | | |
| Выполнил студент группы ИКБО-01-19 | | Иноземцев Н.С. | |
|  | |  | |
| Принял  *Ассистент* | | Коваленко М.А. | |
| Практические работы выполнены | «\_\_\_»\_\_\_\_\_\_\_2020 г. | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись студента) | |
| «Зачтено» | «\_\_\_»\_\_\_\_\_\_\_2020 г. | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись руководителя) | |
|  |  | |  | |

Москва 2020

**Оглавление**

[Практическая работа №5: «JavaScript. Работа с переменными, строками, массивами, циклами и условиями» 3](#_Toc57451448)

[**Задача 1.** 3](#_Toc57451449)

[**Задача 2.** 5](#_Toc57451450)

[**Задача 3.** 7](#_Toc57451451)

[**Задача 4.** 8](#_Toc57451452)

[**Задача 5.** 10](#_Toc57451453)

[**Задача 6.** 11](#_Toc57451454)

[**Задача 7.** 13](#_Toc57451455)

[**Задача 8.** 15](#_Toc57451456)

[**Задача 9.** 17](#_Toc57451457)

[**Задача 10.** 19](#_Toc57451458)

[**Задача 11.** 24](#_Toc57451459)

[**Задача 12.** 27](#_Toc57451460)

[**Задача 13.** 30](#_Toc57451461)

[**Задача 14.** 31](#_Toc57451462)

[**Задача 15.** 36](#_Toc57451463)

[**Задача 16.** 37](#_Toc57451464)

[**Задача 17.** 39](#_Toc57451465)

# Практическая работа №5: «JavaScript. Работа с переменными, строками, массивами, циклами и условиями»

**Цель работы:** Изучить принцип работы с переменными, строками, массивами, циклами и условиями в JavaScript.

**Задача 1.**

Работа с переменными:

* 1. Создать переменную *num* и присвоить ей значение 3. Вывести значение этой переменной на экран с помощью метода *alert;*
  2. Создать переменные *a=10* и *b=2*. Вывести на экран их сумму, разность, произведение и частное (результат деления);
  3. Создать переменные *c=15* и *d=2*. Просуммировать их, а результат присвоить переменной *result*. Вывести на экран значение переменной *result*;
  4. Создать переменные *a=10*, *b=2* и *c=5*. Вывести на экран их сумму;
  5. Создать переменные *a=17* и *b=10*. Отнять от *a* переменную *b* и результат присвоить переменной *c*. Затем создать переменную *d*, присвоить ей значение 7. Сложить переменные *c* и *d*, а результат записать в переменную *resul*t. Вывести на экран значение переменной *result;*

**Описание выполнения задания:**

Листинг 1.1 – создать переменную num и присвоить ей значение 3 и вывести через alert

let num = 3;

alert(num);

Результат выполнения кода на странице (см. Рисунок 1.1):

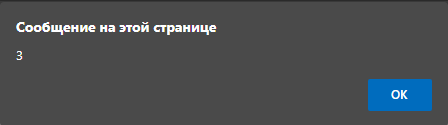


Рисунок 1.1 - результат выполнения кода

Листинг 1.2 - создать две переменные с заданными значениями и вывести все арифметические операции с ними.

let a = 10;

let b = 2;

alert(a+b);

alert(a-b);

alert(a\*b);

alert(a/b);

Результат выполнения кода на странице (см. Рисунок 1.2):

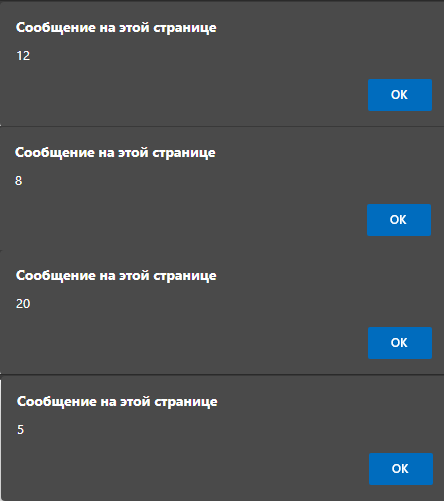


Рисунок 1.2

Листинг 1.3 - создать две переменные с заданными значениями, их сумму присвоить другой переменной и вывести ее.

let c = 15;

let d = 2;

let result = c + d;

alert(result);

Результат выполнения кода на странице (см. Рисунок 1.3):

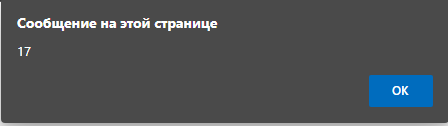


Рисунок 1.3

Листинг 1.4 - создать три переменные с заданными значениями, их сумму вывести в окне браузера.

let a = 10;

let b = 2;

let c = 5;

alert(a+b+c);

Результат выполнения кода на странице (см. Рисунок 1.4):

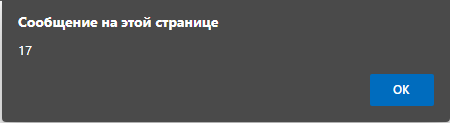


Рисунок 1.4

Листинг 1.5 - создать две переменные с заданными значениями, провести с ними цепочку арифметических операций и присвоить другой переменной и вывести на экран.

let a = 17;

let b = 10;

let c = a - b;

let d = 7;

let result = c + d;

alert(result);

Результат выполнения кода на странице (см. Рисунок 1.5):

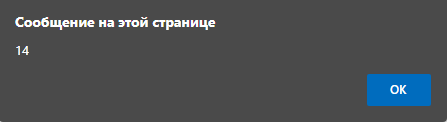


Рисунок 1.5

**Задача 2.**

Работа со строками:

1. Создать переменную *str* и присвоить ей значение *'Привет, Мир!'*. Вывести значение этой переменной на экран;
2. Создать переменные *str1='Привет, '* и *str2='Мир!'*. С помощью этих переменных и операции сложения строк вывести на экран фразу *'Привет, Мир!'*;
3. Создать переменную *nam*e и присвоить ей ваше имя. Вывести на экран фразу *'Привет, %Имя%!;*.
4. Создать переменную *age* и присвоить ей ваш возраст. Выведите на экран 'Мне %Возраст% лет!'.

**Описание выполнения задания:**

Листинг 2.1 – создать переменную и присвоить ей заданное строковое значение и вывести ее на экран.

     str = 'Привет, Мир!';

alert(str);

Результат выполнения кода на странице (Рисунок 2.1):

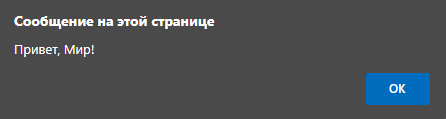


Рисунок 2.1

Листинг 2.2 – создать две переменные с заданными значениями, сложить их используя операцию сложения строк(конкатенацию) и вывести на экран.

str1='Привет, ';

str2='Мир!';

alert(str1 + str2);

Результат выполнения кода на странице (Рисунок 2.2):

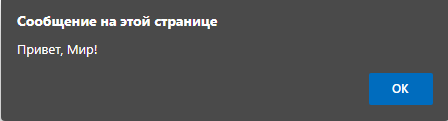


Рисунок 2.2

Листинг 2.3 – создать переменную со своим именем и вывести ее с заданной фразой.

let name = 'Никита';

alert('Привет, ' + name + '!');

Результат выполнения кода на странице (Рисунок 2.3):

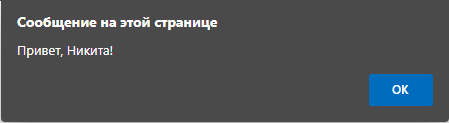


Рисунок 2.3

Листинг 2.4 – создать переменную со своим возрастом и вывести ее с заданной фразой.

let age = 18;

alert('Мне ' + age + ' лет!');

Результат выполнения кода на странице (Рисунок 2.4):

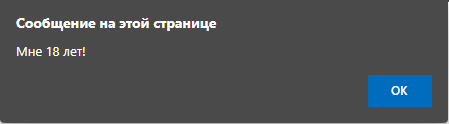


Рисунок 2.4

**Задача 3.**

Функция promt:

* 1. Спросить имя пользователя с помощью методы *prompt*. Вывести с помощью *alert* сообщение *'Ваше имя %имя%'*.
  2. Спросить у пользователя число. Выведите с помощью *alert* квадрат этого числа.

**Описание выполнения задания:**

Листинг 3.1 – спросить имя пользователя используя метод prompt и вывести заданное имя с заданной фразой.

let name = prompt('Введите имя', '');

alert('Ваше имя ' + name);

Результат выполнения кода на странице (Рисунок 3.1):

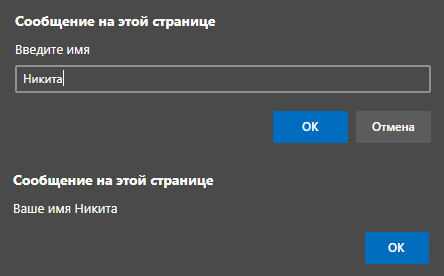


Рисунок 3.1

Листинг 3.2 –спросить у пользователя число и вывести его квадрат.

let num = prompt('Введите число', 0);

alert(num \* num);

Результат выполнения кода на странице (Рисунок 3.2):

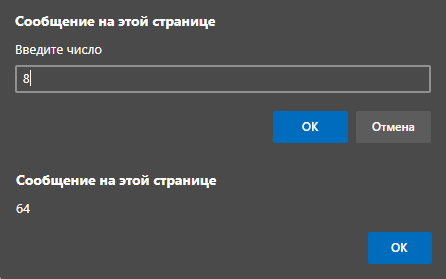


Рисунок 3.2

**Задача 4.**

Обращение к символам строки:

1. Создать переменную *str* и присвоить ей значение *'abcde'*. Обращаясь к отдельным символам этой строки вывести на экран символ *'a'*, символ *'c'*, символ *'e'*.
2. Создать переменную *num* и присвоить ей значение *'12345'*. Найти произведение (умножение) цифр этого числа.

**Описание выполнения задания:**

Листинг 4.1 – создать строку и с помощью обращения к элементам строки вывести символы.

let str = 'abcde';

alert(str[0]);

alert(str[2]);

alert(str[4]);

Результат выполнения кода на странице (Рисунок 4.1):

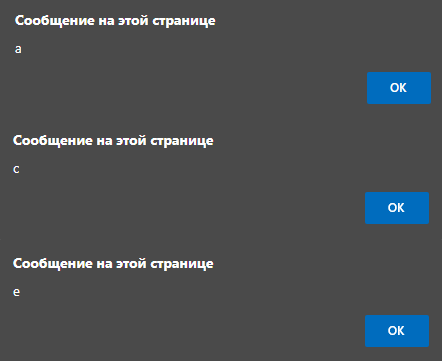
****

Рисунок 4.1

Листинг 4.2 – создать строку из цифр и найти произведение этих цифр.

let num = '12345';

let mul = 1;

for (let i = 0; i < num.length; i++) {

mul \*= num[i];

}

alert(mul);

Результат выполнения кода на странице (Рисунок 4.2):

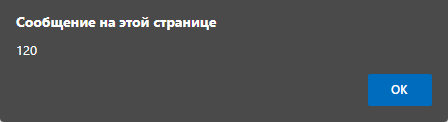
****

Рисунок 4.2

**Задача 5.**

Работа со временем:

1. Написать скрипт, который считает количество секунд в часе, в сутках, в месяце;
2. Создать три переменные - час, минута, секунда. С их помощью вывести текущее время в формате 'час:минута:секунда'.

**Описание выполнения задания:**

Листинг 5.1 – реализовать подсчет количества секунд в часе, в сутках, в месяце.

let hour = 60 \* 60;

let day = hour \* 24;

let month = day \* 30;

alert(hour);

alert(day);

alert(month);

Результат выполнения кода на странице (Рисунок 5.1):

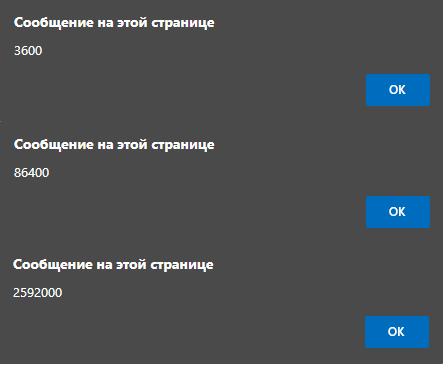


Рисунок 5.1

Листинг 5.2 – создать три переменные отвечающие за секунды, минуты и часы соответственно и вывести их в нужном формате.

let date = new Date();

let hours = date.getHours();

let mins = date.getMinutes();

let secs = date.getSeconds();

alert(hours + " : " + mins + " : " + secs );

Результат выполнения кода на странице (Рисунки 5.2):

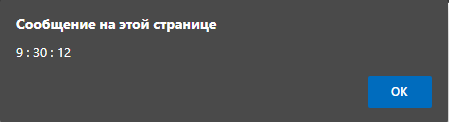
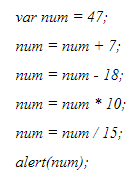
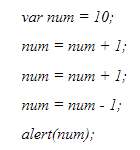


Рисунок 5.2

**Задача 6.**

Работа с присваиванием и декрементами:

1. Переделать ниже представленный код так, чтобы в нем использовались операции *+=, -=, \*=, /=*. Количество строк кода при этом не должно измениться.   
   
2. Переделать ниже представленный код так, чтобы в нем использовались операции *++* и *--*. Количество строк кода при этом не должно измениться./  
   

**Описание выполнения задания:**

Листинг 6.1 – изменить представленный код используя сокращенные операции.

var num = 47;

num += 7;

num -= 18;

num \*= 10;

num /= 15;

alert(num);

Результат выполнения кода на странице (Рисунок 6.1):

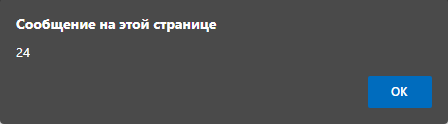


Рисунок 6.1

Листинг 6.2 – изменить представленный код используя сокращенные операции.

var num = 10;

num++;

num++;

num--;

alert(num);

Результат выполнения кода на странице (Рисунок 6.2):

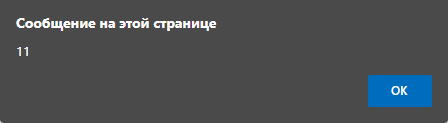


Рисунок 6.2

**Задача 7.**

Работа с массивами:

1. Создать массив *arr = ['a', 'b', 'c']*. Вывести его на экран с помощью функции *alert*.
2. С помощью массива *arr* из предыдущего номера вывести на экран содержимое первого, второго и третьего элементов.
3. Создать массив *arr = ['a', 'b', 'c', 'd']* и с его помощью вывести на экран строку *'a+b, c+d'*.
4. Создать массив *arr* с элементами *2, 5, 3, 9.* Умножить первый элемент массива на второй, а третий элемент на четвертый. Результаты сложить, присвоить переменной *result*. Вывести на экран значение этой переменной.

**Описание выполнения задания:**

Листинг 7.1 - создать массив и вывести его на экран.

let arr = ['a', 'b', 'c'];

alert(arr);

Результат выполнения кода на странице (Рисунок 7.1):

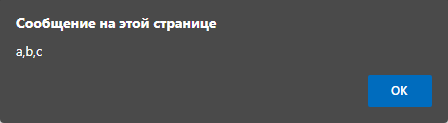


Рисунок 7.1

Листинг 7.2 - с помощью созданного раннее массива вывести каждый его элемент.

alert(arr[0]);

alert(arr[1]);

alert(arr[2]);

Результат выполнения кода на странице (Рисунок 7.2):

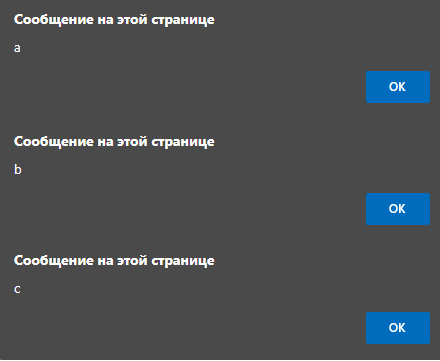


Рисунок 7.2

Листинг 7.3 - создать массив и вывести заданную строку, используя его элементы.

let arr = ['a', 'b', 'c', 'd'];

alert(arr[0] + ' + ' + arr[1] + ', ' + arr[2] + ' + ' + arr[3]);

Результат выполнения кода на странице (Рисунок 7.3):

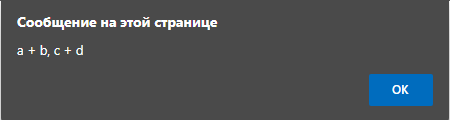


Рисунок 7.3

Листинг 7.4 - провести цепочку арифметических операций с элементами массива и вывести результат.

  let arr = [2, 5, 3, 9];

let result = arr[0] \* arr[1] + arr[2] \* arr[3];

alert(result);

Результат выполнения кода на странице (Рисунок 7.4):

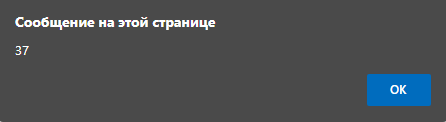


Рисунок 7.4

**Задача 8.**

Объекты(ассоциативные массивы):

1. Создать объект *obj*. Вывести на экран элемент с ключом *'c'* двумя способами: через квадратные скобки и как свойство объекта:  
   
2. Создать массив заработных плат *obj*. Вывести на экран зарплату Пети и Коли:  
   
3. Создать объект с днями недели. Ключами в нем должны служить номера дней от начала недели (понедельник - первый и т.д.). Вывести на экран текущий день недели.
4. Пусть теперь номер дня недели хранится в переменной *day*, например, там лежит число 3. Вывести день недели, соответствующий значению переменной *day*.

**Описание выполнения задания:**

Листинг 8.1 – создать объект с заданными полями и вывести один из его элементов по ключу 2 способами.

var obj = {a: 1, b: 2, c: 3};

alert(obj.c);

alert(obj['c']);

Результат выполнения кода на странице (Рисунок 8.1):

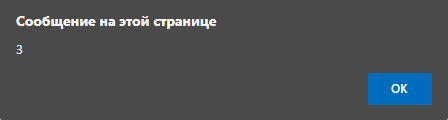


Рисунок 8.1

Листинг 8.2 – создать массив заработных плат и вывести определенные зарплаты.

var obj = {Коля: '1000', Вася: '500', Петя: '200'};

alert(obj.Петя + ' ' + obj.Коля);

Результат выполнения кода на странице (Рисунок 8.2):

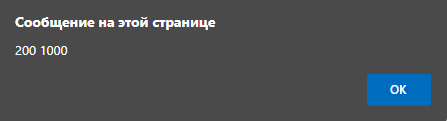


Рисунок 8.2

Листинг 8.3 – создать объект с днями недели, ключ- номера дней, значение - день недели, вывести текущий день.

var week = {1:'понедельник', 2:'вторник', 3:'среда', 4:'четверг', 5:'пятница', 6:'суббота', 7:'воскресенье'};

         let date = new Date();

         for (let key in week) {

             if (key == date.getDay()%7) {

                 alert(week[key]);

                 break;

             }

         }

Результат выполнения кода на странице (Рисунок 8.3):

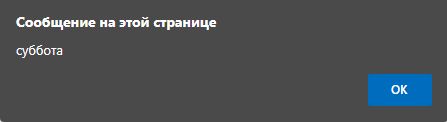


Рисунок 8.3

Листинг 8.4 – вывести день недели соответствующий номеру, лежащем в заданной переменной.

let day = 3;

alert(week[day]);

Результат выполнения кода на странице (Рисунок 8.4):

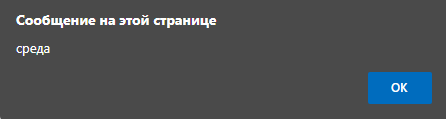


Рисунок 8.4

**Задача 9.**

Многомерные массивы:

* 1. Дан массив [ [1, 2, 3], [4, 5, 6], [7,8,9] ]. Вывести на экран цифру 4 из этого массива.
  2. Дан объект {js:['jQuery', 'Angular'], php: 'hello', css: 'world'}. Вывести с его помощью слово 'jQuery'.
  3. Создать двухмерный массив. Первые два ключа - это 'ru' и 'en'. Пусть первый ключ содержит элемент, являющийся массивом названий дней недели по-русски, а второй - по-английски. Вывести с помощью этого массива понедельник по-русски и среду по-английски (пусть понедельник - это нулевой день).
  4. Пусть теперь в переменной lang хранится язык (она принимает одно из значений или 'ru', или 'en' - либо то, либо то), а в переменной day - номер дня. Вывести словом день недели, соответствующий переменным lang и day. То есть: если, к примеру, lang = 'ru' и day = 3 - то вывести 'среда'.

**Описание выполнения задания:**

Листинг 9.1 - найти и вывести заданную цифра из массива.

let arr = [ [1, 2, 3], [4, 5, 6], [7,8,9] ];

arr.forEach(a => a.indexOf(4) !== -1 ? alert(a.filter(b => b ===4)) : a);

Результат выполнения кода на странице (Рисунок 9.1):

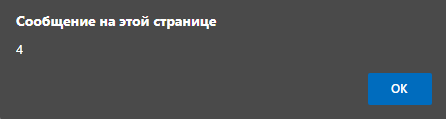


Рисунок 9.1

Листинг 9.2 - вывести заданный объект из массива.

let arr = {js:['jQuery', 'Angular'], php: 'hello', css: 'world'};

alert(arr.js[0]);

Результат выполнения кода на странице (Рисунок 9.2):

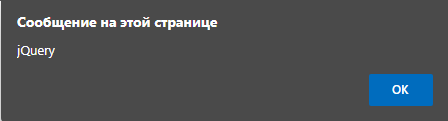


Рисунок 9.2

Листинг 9.3 - создать двумерный массив, где ключи - 'en' и 'ru', каждый из ключей содержит массив дней недели на своем языке соответственно.

let arr = {ru:['понедельник', 'вторник', 'среда', 'четверг', 'пятница', 'суббота', 'воскресенье'], en: ['monday', 'tuesday', 'wednesday', 'thursday', 'friday', 'saturday', 'sunday']};

alert(arr.ru[0] + ' ' + arr.en[2]);

Результат выполнения кода на странице (Рисунок 9.3):

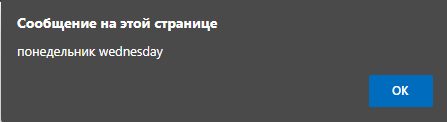


Рисунок 9.3

Листинг 9.4 - задать переменную, хранящую ключ, и переменную, хранящую значение, по ним найти соответствующий элемент в массиве.

let lang = 'en';

let day = '6';

alert(arr[lang][--day]);

Результат выполнения кода на странице (Рисунок 9.4):

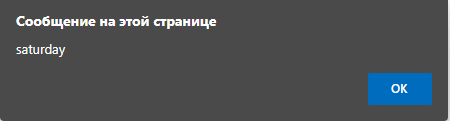


Рисунок 9.4

**Задача 10.**

Работа с if-else:

1. Если переменная a равна нулю, то вывести 'Верно', иначе выведите 'Неверно'. Проверьте работу скрипта при a, равном 1, 0, -3.
2. Если переменная a больше нуля, то вывести 'Верно', иначе вывести 'Неверно'. Проверьте работу скрипта при a, равном 1, 0, -3.
3. Если переменная a меньше нуля, то вывести 'Верно', иначе вывести 'Неверно'. Проверьте работу скрипта при a, равном 1, 0, -3.
4. Если переменная a больше или равна нулю, то вывести 'Верно', иначе вывести 'Неверно'. Проверьте работу скрипта при a, равном 1, 0, -3.
5. Если переменная a меньше или равна нулю, то вывести 'Верно', иначе вывести 'Неверно'. Проверьте работу скрипта при a, равном 1, 0, -3.
6. Если переменная a не равна нулю, то вывести 'Верно', иначе вывести 'Неверно'. Проверьте работу скрипта при a, равном 1, 0, -3.
7. Если переменная a равна 'test', то вывести 'Верно', иначе вывести 'Неверно'. Проверьте работу скрипта при a, равном 'test', 'тест', 3.
8. Если переменная a равна '1' и по значению и по типу, то вывести 'Верно', иначе вывести 'Неверно'. Проверьте работу скрипта при a, равном '1', 1, 3.

**Описание выполнения задания:**

Листинг 10.1 – проверить работу условия равенства нулю.

let a = 0;

if(a == 0) {

alert("Верно");

}

else {

alert("Неверно");

}

Результат выполнения кода на странице (Рисунок 10.1):

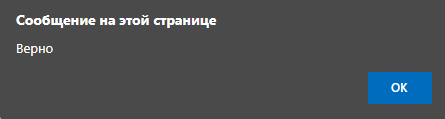


Рисунок 10.1

Листинг 10.2 – проверить работу условия больше нуля.

let a = 1;

if(a > 0) {

alert("Верно");

}

else {

alert("Неверно");

}

Результат выполнения кода на странице (Рисунок 10.2):

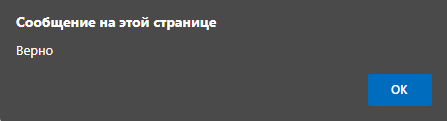


Рисунок 10.2

Листинг 10.3 – проверить работу условия меньше нуля.

let a = -5;

if(a < 0) {

alert("Верно");

}

else {

alert("Неверно");

}

Результат выполнения кода на странице (Рисунок 10.3):

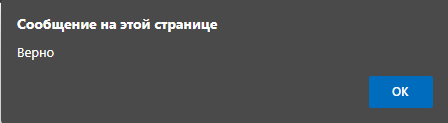


Рисунок 10.3

Листинг 10.4 – проверить работу условия больше или равно нулю.

let a = -2;

if(a >= 0) {

alert("Верно");

}

else {

alert("Неверно");

}

Результат выполнения кода на странице (Рисунок 10.4):

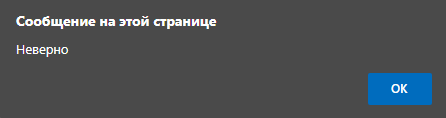


Рисунок 10.4

Листинг 10.5 – проверить работу условия меньше или равно нулю.

let a = 1;

if(a <= 0) {

alert("Верно");

}

else {

alert("Неверно");

}

Результат выполнения кода на странице (Рисунок 10.5):

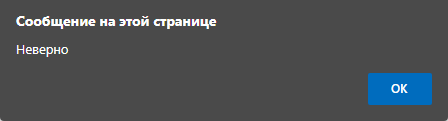


Рисунок 10.5

Листинг 10.6 – проверить работу условия не равно нулю.

let a = 1;

if(a != 0) {

alert("Верно");

}

else {

alert("Неверно");

}

Результат выполнения кода на странице (Рисунок 10.6):

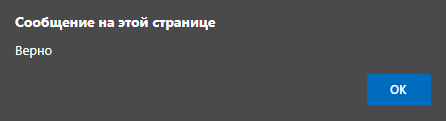


Рисунок 10.6

Листинг 10.7 – создать массив со строковами и целочисленными элементами и проверить на равенство заданной строке.

let a = 'test';

if(a == 'test') {

alert("Верно");

}

else {

alert("Неверно");

}

Результат выполнения кода на странице (Рисунок 10.7):

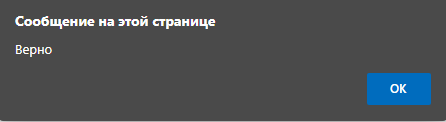


Рисунок 10.7

Листинг 10.8 – создать массив со строковами и целочисленными элементами и проверить на равенство по типу заданному значению.

let a = 3;

if(a == '1' || a == 1) {

alert("Верно");

}

else {

alert("Неверно");

}

Результат выполнения кода на странице (Рисунок 10.8):

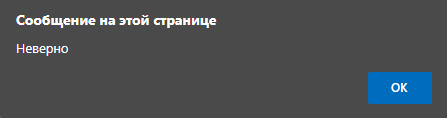
****

Рисунок 10.8

**Задача 11.**

Работа с логическими переменными:

* 1. Если переменная test равна true, то вывести 'Верно', иначе вывести 'Неверно'. Проверьте работу скрипта при test, равном true, false. Написать два варианта скрипта - с короткой записью и с длинной.
  2. Если переменная test не равна true, то вывести 'Верно', иначе вывести 'Неверно'. Проверить работу скрипта при test, равном true, false. Написать два варианта скрипта - с короткой записью и с длинной.

**Описание выполнения задания:**

Листинг 11.1 – проверить эквивалентность булевых переменных.

let test = true;

if(test) {

alert('Верно');

}

else {

alert('Неверно');

}

if(test == true) {

alert('Верно');

}

else {

alert('Неверно');

}

Результат выполнения кода на странице (Рисунок 11.1):

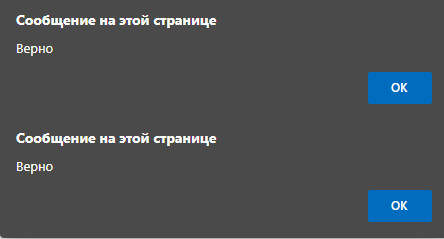


Рисунок 11.1

Листинг 11.2 – проверить неравность булевых переменных.

let test = false;

if(!test) {

alert('Верно');

}

else {

alert('Неверно');

}

if(test != true) {

alert('Верно');

}

else {

alert('Неверно');

}

Результат выполнения кода на странице (Рисунок 11.2):

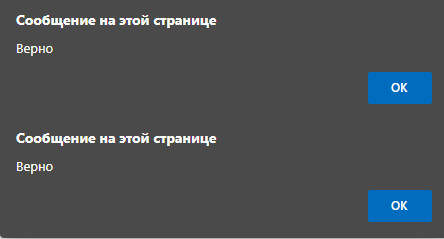


Рисунок 11.2

**Задача 12.**

Работа с && (и) и || (или):

1. Если переменная a больше нуля и меньше 5-ти, то вывести 'Верно', иначе вывести 'Неверно'. Проверить работу скрипта при a, равном 5, 0, -3, 2.
2. Если переменная a равна нулю или равна двум, то прибавить к ней 7, иначе поделить ее на 10. Вывести новое значение переменной на экран. Проверить работу скрипта при a, равном 5, 0, -3, 2.
3. Если переменная a равна или меньше 1, а переменная b больше или равна 3, то вывести сумму этих переменных, иначе вывести их разность (результат вычитания). Проверить работу скрипта при a и b, равном 1 и 3, 0 и 6, 3 и 5.
4. Если переменная a больше 2-х и меньше 11-ти, или переменная b больше или равна 6-ти и меньше 14-ти, то вывести 'Верно', в противном случае вывести 'Неверно'.

**Описание выполнения задания:**

Листинг 12.1 – проверить, что переменная больше нуля и меньше 5-ти.

let a = 5;

if(a > 0 && a < 5) {

alert("Верно");

}

else {

alert("Неверно");

}

Результат выполнения кода на странице (Рисунок 12.1):

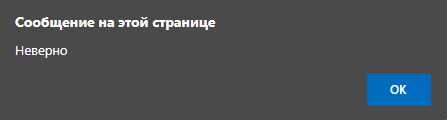


Рисунок 12.1

Листинг 12.2 – проверить, что переменная либо равна нулю, либо 2-м, и выполнить соответствующие арифметическое действие.

let a = 5;

if(a == 0 || a == 2) {

a += 7;

}

else {

a /= 10;

}

alert(a);

Результат выполнения кода на странице (Рисунок 12.2):

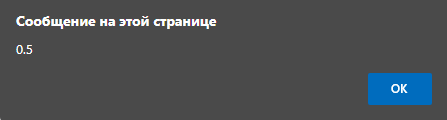


Рисунок 12.2

Листинг 12.3 –определить 2 массива, при выполнении условий для элементов из первого и второго массива выполнить с ними соответствующие арифметические операции.

let a = 1;

let b = 3;

if(a <= 1 && b >= 3) {

alert(a + b);

}

else {

alert(a - b);

}

Результат выполнения кода на странице (Рисунок 12.3):

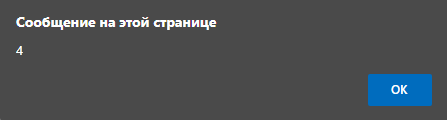


Рисунок 12.3

Листинг 12.4 – определить сложное условие для каждого элемента из двух массивом и, при выполнении какого-либо из них, вывести соответствующую надпись.

let a = 4;

let b = 3;

if((a > 2 && a < 11) || b >= 6 && b < 14) {

alert("Верно");

}

else {

alert("Неверно");

}

Результат выполнения кода на странице (Рисунок 12.4):

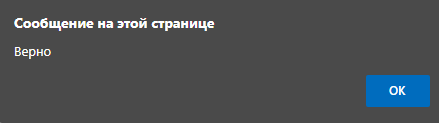


Рисунок 12.4

**Задача 13.**

Работа со switch-case: Переменная num может принимать 4 значения: 1, 2, 3 или 4. Если она имеет значение '1', то в переменную result записать 'зима', если имеет значение '2' – 'весна' и так далее. Решить задачу через switch-case.

**Описание выполнения задания:**

Листинг 13 – используя конструкцию switch-case выбирать в соответствие с цифрой время года.

let num = 3;

let result = "";

switch(num) {

case 1:

result = 'зима';

break;

case 2:

result = 'весна';

break;

case 3:

result = 'лето';

break;

case 4:

result = 'осень';

break;

}

alert(result);

Результат выполнения кода на странице (Рисунок 13):

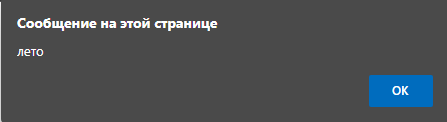


Рисунок 13

**Задача 14.**

Циклы while и for:

1. Вывести столбец чисел от 1 до 100.
2. Вывести столбец чисел от 11 до 33.
3. Вывести столбец четных чисел в промежутке от 0 до 100.
4. С помощью цикла найти сумму чисел от 1 до 100.

**Описание выполнения задания:**

Листинг 14.1 – вывести с помощью конструкций цикла for и while числа от 1 до 100.

let i = 1;

while(i != 101) {

console.log(i);

i++;

}

for (let i = 1; i < 101; i++) {

console.log(i);

}

Результат выполнения кода на странице(первые 50 цифр) (Рисунок 14.1):

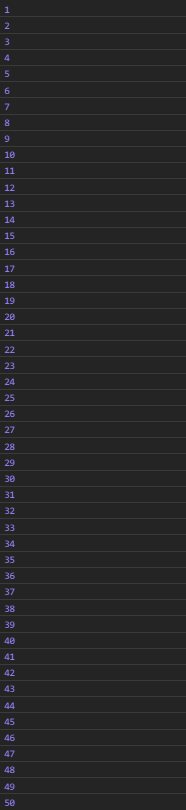


Рисунок 11.1

Листинг 14.2 – вывести с помощью конструкций цикла for и while числа от 11 до 33.

let i = 11;

while(i != 34) {

console.log(i);

i++;

}

for (let i = 11; i < 34; i++) {

console.log(i);

}

Результат выполнения кода на странице (Рисунок 14.2):

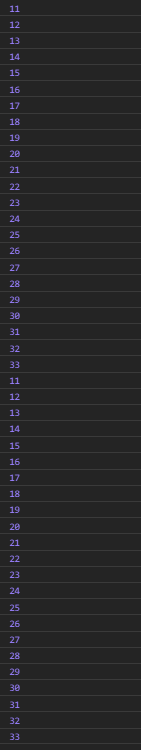


Рисунок 14.2

Листинг 14.3 – вывести с помощью конструкций цикла for и while четные числа от 0 до 100

let i = 0;

while(i != 100) {

console.log(i);

i += 2;

}

for (let i = 0; i < 100; i += 2) {

console.log(i);

}

Результат выполнения кода на странице(первая половина цифр) (Рисунок 14.3):

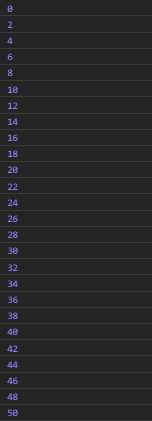


Рисунок 14.2

Листинг 14.4 – вывести с помощью конструкций цикла for и while сумму чисел от 0 до 100.

let i = 1;

let sum = 0;

while(i != 100) {

sum += i;

i++;

}

alert(sum);

sum = 0;

for (let i = 0; i < 100; i += 2) {

sum += i;

}

Результат выполнения кода на странице (Рисунок 14.4):

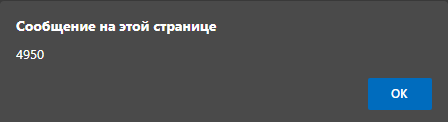


Рисунок 14.4

**Задача 15.**

Циклы while и for:

1. Дан массив с элементами [1, 2, 3, 4, 5]. С помощью цикла for вывести все эти элементы на экран.
2. Дан массив с элементами [1, 2, 3, 4, 5]. С помощью цикла for найти сумму элементов этого массива. Записать ее в переменную result.

**Описание выполнения задания:**

Листинг 15.1 – вывести элементы массива с помощью цикла.

let a = [1, 2, 3, 4, 5];

for (let i = 0; i < a.length; i++) {

console.log(a[i]);

}

Результат выполнения кода на странице (Рисунок 15.1):



Рисунок 15.1

Листинг 15.2 – вывести сумму элементов массива.

let a = [1, 2, 3, 4, 5];

let result = 0

for (let i = 0; i < a.length; i++) {

result += a[i];

}

alert(result);

Результат выполнения кода на странице (Рисунок 15.2):

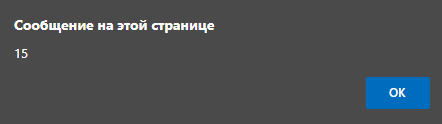


Рисунок 15.2

**Задача 16.**

Работа с for-in:

1. Дан объект obj. С помощью цикла for-in вывести на экран ключи и элементы этого объекта.  
   
2. Дан объект obj с ключами Коля, Вася, Петя с элементами '200', '300', '400'. С помощью цикла for-in вывести на экран строки такого формата: 'Коля - зарплата 200 долларов.'.

**Описание выполнения задания:**

Листинг 16.1 – вывести с помощью цикла for-in ключи и элементы заданного объекта.

var obj = {green: 'зеленый', red: 'красный', blue: 'голубой'};

for (var val in obj) {

alert(val + ' ' + obj[val]);

}

Результат выполнения кода на странице (Рисунок 16.1):

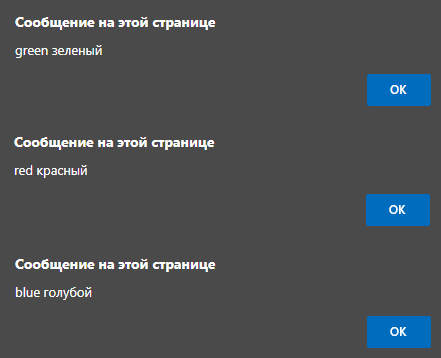


Рисунок 16.1

Листинг 16.2 – вывести с помощью цикла for-in ключи и элементы заданного объекта в определенном формате.

var obj = {Коля: '200', Вася: '300', Петя: '400'};

for (var val in obj) {

alert(val + ' - зарплата ' + obj[val] + ' долларов');

}

Результат выполнения кода на странице (Рисунок 16.2):

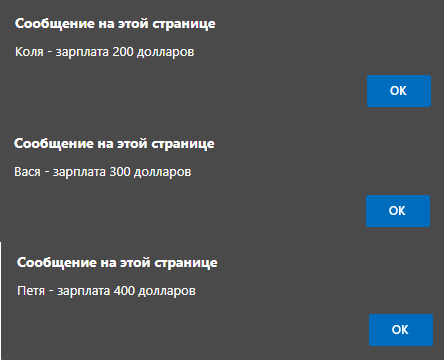


Рисунок 16.2

**Задача 17.**

Задачи с циклами:

1. Дан массив с элементами 2, 5, 9, 15, 0, 4. С помощью цикла for и оператора if вывести на экран столбец тех элементов массива, которые больше 3-х, но меньше 10..
2. Дан массив с числами. Числа могут быть положительными и отрицательными. Найти сумму положительных элементов массива.
3. Дан массив с элементами 1, 2, 5, 9, 4, 13, 4, 10. С помощью цикла for и оператора if проверить есть ли в массиве элемент со значением, равным 4. Если есть - вывести на экран 'Есть!' и выйти из цикла. Если нет - ничего делать не надо.
4. Дан массив числами, например: [10, 20, 30, 50, 235, 3000]. Вывести на экран только те числа из массива, которые начинаются на цифру 1, 2 или 5.
5. Дан массив с элементами 1, 2, 3, 4, 5, 6, 7, 8, 9. С помощью цикла for создать строку '-1-2-3-4-5-6-7-8-9-'.
6. Составить массив дней недели. С помощью цикла for вывести все дни недели, а выходные дни вывести жирным.
7. Составить массив дней недели. С помощью цикла for вывести все дни недели, а текущий день вывести курсивом. Текущий день должен храниться в переменной day.
8. Дано число n=1000. Делить его на 2 столько раз, пока результат деления не станет меньше 50. Какое число получится? Посчитать количество итераций необходимых для этого (итерация - это проход цикла), и записать его в переменную num.

**Описание выполнения задания:**

Листинг 17.1 – вывести с помощью конструкций цикла for те элементы, которые больше 3-х и меньше 10-и.

let a = [2, 5, 9, 15, 0, 4];

for(let i = 0; i < a.length; i++) {

if(a[i] > 3 && a[i] < 10) {

console.log(a[i]);

}

}

Результат выполнения кода на странице (Рисунок 17.1):



Рисунок 17.1

Листинг 17.2 – с помощью конструкций цикла for найти сумму положительныъ элементом массива.

let a = [2, 5, -10, 15, 0, 4];

let sum = 0;

for(let i = 0; i < a.length; i++) {

if(a[i] > 0) {

sum += a[i];

}

}

alert(sum);

Результат выполнения кода на странице (Рисунок 17.2):

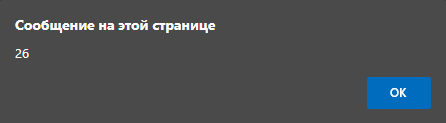


Рисунок 17.2

Листинг 17.3 – с помощью конструкций цикла for проверить существует ли заданное число в массиве и вывести соответствующий текст.

let a = [1, 2, 5, 9, 4, 13, 4, 10];

for(let i = 0; i < a.length; i++) {

if(a[i] == 4) {

alert('Есть!');

break;

}

}

Результат выполнения кода на странице (Рисунок 17.3):

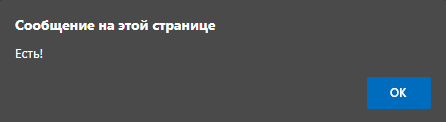


Рисунок 17.3

Листинг 17.4 – вывести на экран только те числа массива, которые начинаются 1, 2 или 5.

let a = [10, 20, 30, 50, 235, 3000];

for(let i = 0; i < a.length; i++) {

let str = String(a[i]);

if (str[0] == "1" || str[0] == "2" || str[0] == "5") {

console.log(a[i]);

}

}

Результат выполнения кода на странице (Рисунок 17.4):

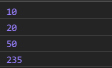


Рисунок 17.4

Листинг 17.5 – вывести строку с элементами массива в заданном формате.

let a = [1, 2, 3, 4, 5, 6, 7, 8, 9];

let res = '-';

for(let i = 0; i < a.length; i++) {

res += a[i];

res += '-';

}

alert(res);

Результат выполнения кода на странице (Рисунок 17.5):

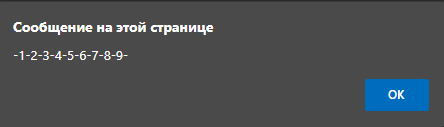


Рисунок 17.5

Листинг 17.6 – вывести массив дней недели, выделив выходные жирным.

let a = ['понедельник', 'вторник', 'среда', 'четверг', 'пятница', 'суббота', 'воскресенье'];

for(let i = 0; i < a.length; i++) {

if (i > 4) {

console.log(a[i].bold());

}

else {

console.log(a[i]);

}

}

Результат выполнения кода на странице (Рисунок 17.6):

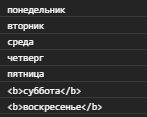


Рисунок 17.6

Листинг 17.7 – вывести массив дней недели, выделив текущий день курсивом.

let a = ['понедельник', 'вторник', 'среда', 'четверг', 'пятница', 'суббота', 'воскресенье'];

let day = 'суббота'

for(let i = 0; i < a.length; i++) {

if (a[i] == day) {

console.log(a[i].italics());

}

else {

console.log(a[i]);

}

}

Результат выполнения кода на странице (Рисунок 17.7):

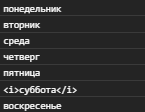


Рисунок 17.7

Листинг 17.8 – узнать какое число получится, если число 1000 делить на 2 пока оно больше 50, посчитать сколько пройдет итераций.

let n = 1000;

let num = 0;

while(n > 50) {

num++;

n /= 2;

}

alert(num + ' ' + n);

Результат выполнения кода на странице (Рисунок 17.8):

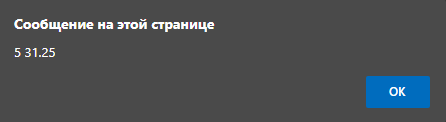


Рисунок 17.8