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

**Справочная информация**

**Цикл while**

Цикл **while** будет выполняться до тех пор, пока **верно** (истинно) выражение, переданное ему параметром. Смотрите синтаксис:

while ( пока выражение истинно ) {

выполняем этот код циклически;

в начале каждого цикла проверяем выражение в круглых скобках

}

/\*

Цикл закончится, когда выражение перестанет быть истинным.

Если оно было ложным изначально - то он не выполнится ни разу!

\*/

В принципе, цикл **while** может выполняться **бесконечно** (но это приведет к зависанию скрипта!), достаточно передать ему выражение, которое **никогда не станет ложным**. Например, так:

var test **=** true;

while (test **===** true) {

/\*

Написанный здесь код будет выполняться 'вечно'

(пока скрипт не будет остановлен принудительно).

Не стоит это повторять - это приведет к зависанию сервера!

\*/

}

Давайте последовательно выведем с помощью цикла while числа от одного до пяти:

var i **=** 0; //счетчик цикла

while (i **<** 5) {

/\*

С помощью оператора ++ увеличиваем i на единицу

при каждом проходе цикла.

\*/

i**++**;

alert(i);

}

Обратите внимание на переменную **i** – она является так называемым **счетчиком цикла**. Счетчики используются для того, чтобы подсчитывать, сколько раз выполнился цикл. Кроме того, они выполняют вспомогательную роль - в нашей задаче мы использовали счетчик, чтобы вывести цифры от 1 до 5.

Для счетчиков принято использовать буквы **i**, **j** и **k**.

**Цикл for**

Цикл **for** является **альтернативой while**. Он более сложен для понимания, но чаще всего его любят больше, чем while за то, что он занимает меньше строчек.

for ( начальные команды; условие окончания цикла; команды после прохода цикла ) {

тело цикла

}

**Начальные команды** - это то, что выполнится перед стартом цикла. Они выполнятся только один раз. Обычно там размещают начальные значения счетчиков, пример:   
**i = 0**.

**Условие окончания цикла** - пока оно истинное, цикл будет работать, пример:   
**i <10**.

**Команды после прохода цикла** - это команды, которые будут выполнятся каждый раз при окончании прохода цикла. Обычно там увеличивают счетчики, например: **i++**.

Давайте с помощью цикла **for** выведем последовательно числа от 0 до 9:

/\*

В начале цикла i будет равно нулю,

цикл будет выполнятся пока i < 10,

после каждого прохода к i прибавляется единица:

\*/

for (var i **=** 0; i **<** 10; i**++**) {

alert(i); //выведет 0, 1, 2... 9

}

**Инструкция break**

Иногда нам необходимо прервать выполнение цикла досрочно, в случае с циклом for это значит до того, как цикл переберет все элементы массива.

Зачем такое может понадобится? Например, перед нами стоит задача выводить элементы массива до тех пор, пока не встретится число 3. Как только встретится - цикл должен завершить свою работу.

Такое можно сделать с помощью инструкции **break** - если выполнение цикла дойдет до нее, цикл закончит свою работу.

Давайте решим приведенную выше задачу - оборвем цикл, как только нам встретится число 3:

var arr **=** [1, 2, 3, 4, 5];

for (var i **=** 0; i **<** arr.length; i**++**) {

if (arr[i] **===** 3) {

break; //выходим из цикла

} else {

alert(arr[i]);

}

}

**Задания**

**Задание 14: «Циклы while и for»**

Решить эти задачи сначала через цикл while, а затем через цикл for:

**Задача 14.1** Вывести столбец чисел от 1 до 100.

**Задача 14.2** Вывести столбец чисел от 11 до 33.

**Задача 14.3** Вывести столбец четных чисел в промежутке от 0 до 100.

**Задача 14.4** С помощью цикла найти сумму чисел от 1 до 100.

**Справочная информация**

**Цикл for для массивов**

С помощью цикла **for** можно последовательно перебрать элементы массива. Делается это следующим образом:

var arr **=** [1, 2, 3, 4, 5];

for (var i **=** 0; i **<=** arr.length-1; i**++**) {

alert(arr[i]); //выведет 1, 2, 3, 4, 5

}

Ключевым моментом является то, что мы делаем перебор от нуля до длины массива минус 1 (так как номер последнего элемента массива на единицу меньше его длины).

Можно не отнимать единицу, а место **<=** сделать **<**:

var arr **=** [1, 2, 3, 4, 5];

for (var i **=** 0; i **<** arr.length; i**++**) {

alert(arr[i]); //выведет 1, 2, 3, 4, 5

}

**Задания**

**Задание 15: «Работа с for для массивов»**

**Задача 15.1** Дан массив с элементами [1, 2, 3, 4, 5]. С помощью цикла for вывести все эти элементы на экран.

**Задача 15.2** Дан массив с элементами [1, 2, 3, 4, 5]. С помощью цикла for найти сумму элементов этого массива. Записать ее в переменную result.

**Справочная информация**

**Цикл for-in**

Для перебора объекта используется так называемый цикл **for-in**. Давайте посмотрим, как он работает.

Пусть у нас дан такой объект:

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

Давайте выведем его ключи. Для этого используем такую конструкцию: **for (key in obj)**, где **obj** - это объект, который мы перебираем, а **key** - это переменная, в которую последовательно будут ложиться ключи объекта (ее название может быть любым, какое придумаете - такое и будет).

То есть в данном цикле не надо указывать условие окончания - он будет перебирать ключи объекта, пока они не закончатся.

Итак, вот так мы выведем все ключи объекта (по очереди):

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

for (key in obj) {

alert(key); //выведет 'Коля', 'Вася', 'Петя'

}

Если нам нужны не ключи, а значения, нужно обратиться к нашему объекту по ключу, вот так: **obj[key]**.

Как это работает: в переменной **key** сначала будет 'Коля', то есть **obj[key]** в данном случае все равно, что **obj['Коля']**, при следующем проходе цикла в переменной **key** будет 'Вася' и так далее.

Итак, выведем все элементы объекта:

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

for (key in obj) {

alert(obj[key]); //выведет 200, 300, 400

}

**Задания**

**Задание 16: Работа с for-in**

**Задача 16.1** Дан объект obj. С помощью цикла for-in вывести на экран ключи и элементы этого объекта.

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

**Задача 16.2** 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.'.

**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. Делите 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.