**Лабораторная работа №11**

**Циклы в PHP**

В данном уроке мы с вами начнем изучение *циклов*.

Циклы используются для того, чтобы некоторый участок кода выполнился несколько раз подряд. Зачем это нужно - представьте, что вам нужно возвести в квадрат 100 элементов массива. Если обращаться к каждому элементу отдельно по его ключу - это займет 100 строчек кода, и для того, чтобы написать этого код, нужно будет потратить довольно много времени.

Но это не нужно - у нас есть возможность сделать так, чтобы PHP за нас выполнил некоторую операцию нужное количество раз. Например, возвел все элементы массива в квадрат. Это и делается с помощью циклов.

Начнем наше изучение циклов с цикла while.

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

<?php

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.

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

<?php

$i = 1; // задаем какую-нибудь переменную

while ($i <= 5) {

echo $i; // выводим содержимое $i в консоль

$i++; // увеличиваем $i на единицу при каждом проходе цикла

}

?>

Каждый проход цикла по-научному называется *итерацией* цикла. Можно сказать, что мы увеличиваем нашу переменную $i на единицу в каждой итерации цикла.

Сама переменная $i называется *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.

**Задача №1**

Выведите на экран числа от 1 до 100.

**Задача №2**

Выведите на экран числа 11 до 33.

**Более сложный цикл**

К счетчику не обязательно прибавлять единицу. Давайте для примера выведем столбец четных чисел от 2 до 10. Для этого начальное значение переменной $i зададим как 2 и будем прибавлять двойку:

<?php

$i = 2;

while ($i <= 10) {

echo $i;

$i += 2; // увеличиваем $i на 2 при каждом проходе цикла

}

?>

**Задача №3**

Выведите на экран четные числа в промежутке от 0 до 100.

**Задача №4**

Выведите на экран нечетные числа в промежутке от 1 до 99.

**Обратный отсчет**

Счетчик не обязательно должен увеличиваться в цикле. Бывают и обратные ситуации, когда счетчик, наоборот, уменьшается. Давайте для примера выведем столбец чисел от 10 до 1:

<?php

$i = 10; // начальное значение 10

while ($i >= 1) { // пока $i больше 1

echo $i;

$i--; // уменьшаем $i на единицу

}

?>

**Задача №5**

Выведите на экран числа от 30 до 0.

Цикл while может выполняться бесконечно (это приведет к зависанию скрипта и страницы браузера), достаточно передать ему выражение, которое *никогда не станет ложным*. Например, так:

<?php

$test = true;

while ($test === true) {

/\*

Написанный здесь код будет выполняться "вечно"

(пока скрипт не будет остановлен принудительно).

\*/

}

?>

Начинающие программисты время от времени загоняют свою программу в бесконечный цикл (с вами это тоже обязательно случится пару раз).

Давайте рассмотрим распространенные ошибки начинающих.

Ошибка 1

Пусть есть вот такой цикл, выводящий числа от 1 до 10:

<?php

$i = 1;

while ($i <= 10) {

echo $i;

$i++;

}

?>

Представим теперь, что программист забыл сделать увеличение счетчика в цикле:

<?php

$i = 1;

while ($i <= 10) {

echo $i;

}

?>

В этом случае цикл будет выполнятся бесконечно, так как условие, при котором цикл закончится, никогда не будет достигнуто.

Ошибка 2

Пусть опять наш программист хотел выводить числа от 1 до 10, однако, вместо ++ написал --:

<?php

$i = 1;

while ($i <= 10) {

echo $i;

$i--; // перепутал ++ и --

}

?>

В этом случае, хотя счетчик и изменяется, но он изменяется не в ту сторону - и опять условие никогда не будет достигнуто.

Ошибка 3

Бывает и обратная проблема - можно так написать код, что условие цикла уже изначально будет неверно и цикл просто не запустится. Пусть, к примеру, программист хотел вывести столбец чисел от 1 до 10, однако, вместо знака меньше написал знак больше:

<?php

$i = 1;

while ($i > 10) { // правильно было написать $i < 10

echo $i;

$i++;

}

?>

В таком случае цикл просто не начнется, так как условие изначально неверно.

Ошибка 4

Бывает и так, что некоторым начинающим кажется, что цикл крутится не *пока условие верно*, а наоборот - цикл крутится пока условие не верно и остановится, когда оно станет верным.

Вот пример такого кода:

<?php

$i = 1;

while ($i == 10) { // зачем-то написано ==

echo $i;

$i++;

}

?>

При таком условии цикл вообще не начнется, так как условие просто изначально не верно.

**Задача №6**

В следующем коде программист вывел числа от 10 до 1:

<?php

$i = 10;

while ($i >= 1) {

echo $i;

}

?>

В коде, однако, была допущена ошибка, которая привела к тому, что цикл выполняется бесконечно. Исправьте ошибку программиста.

**Задача №7**

В следующем коде программист вывел числа от 10 до 1:

<?php

$i = 10;

while ($i >= 1) {

echo $i;

$i++;

}

?>

В коде, однако, была допущена ошибка, которая привела к тому, что цикл выполняется бесконечно. Исправьте ошибку программиста.

**Задача №8**

В следующем коде программист вывел числа от 10 до 1:

<?php

$i = 10;

while ($i <= 0) {

echo $i;

$i--;

}

?>

В коде, однако, была допущена ошибка, которая привела к тому, что на экран ничего не вывелось. Исправьте ошибку программиста.

**Задача №9**

В следующем коде программист вывел числа от 10 до 1:

<?php

$i = 10;

while ($i == 0) {

echo $i;

$i--;

}

?>

В коде, однако, была допущена ошибка, которая привела к тому, что на экран ничего не вывелось. Исправьте ошибку программиста.

**Задача №10**

В следующем коде программист вывел числа от 10 до 1:

<?php

$i = 10;

while ($i == 10) {

echo $i;

$i--;

}

?>

В коде, однако, была допущена ошибка, которая привела к тому, что цикл вывел только число 10 и закончил свою работу. Исправьте ошибку программиста.

**Цикл FOR в PHP**

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

Вот его синтаксис:

<?php

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

тело цикла

}

?>

Начальные команды - это то, что выполнится перед стартом цикла. Они выполнятся только один раз. Обычно там размещают начальные значения счетчиков. Условие окончания - это условие, при котором цикл будет крутится, пока оно истинное. Команды после прохода - это команды, которые будут выполнятся каждый раз при окончании прохода цикла. Обычно там увеличивают счетчики.

Давайте с помощью цикла for выведем последовательно числа от 1 до 9:

<?php

/\*

В начале цикла $i будет равно нулю,

цикл будет выполнятся пока $i <= 9,

после каждого прохода к $i прибавляется единица:

\*/

for ($i = 0; $i <= 9; $i++) {

echo $i; // выведет 1, 2... 9

}

?>

**Задача №11**

С помощью цикла for выведите на экран числа от 1 до 100.

**Задача №12**

С помощью цикла for выведите на экран числа от 11 до 33.

**Задача №13**

С помощью цикла for выведите на экран четные числа в промежутке от 0 до 100.

**Задача №14**

С помощью цикла for выведите на экран нечетные числа в промежутке от 1 до 99.

**Задача №15**

С помощью цикла for выведите на экран числа от 100 до 0.

Давайте с помощью цикла найдем сумму целых чисел от 1 до 100. Для таких задач существует стандартное решение, которое мы сейчас разберем. Это решение заключается в том, что циклом перебираются числа и их сумма последовательно записывается в какую-то переменную:

<?php

$result = 0;

for ($i = 1; $i <= 100; $i++) {

$result = $result + $i;

}

echo $result; // искомая сумма

?>

Как это работает: изначально переменная $result имеет значение 0, затем при первом проходе цикла в нее записывается ее текущее содержимое плюс значение счетчика. Получится, что каждую итерацию наша переменная будет расти, постепенно накапливая в себе результат.

**Задача №16**

Найдите произведение целых чисел от 1 до 20.

**Задача №17**

Найдите сумму четных чисел от 2 до 100.

**Задача №18**

Найдите сумму нечетных чисел от 1 до 99.