**Лабораторная работа №14**

**Циклы для массивов в PHP**

Пусть у нас дан вот такой массив:

<?php

$arr = [1, 2, 3, 4, 5];

?>

Давайте переберем этот массив циклом foreach:

<?php

$arr = [1, 2, 3, 4, 5];

foreach ($arr as $elem) {

echo $elem;

}

?>

На самом деле элементы массива можно перебрать и обычным циклом for. Такое нужно редко, но иногда бывает полезно. Давайте выполним такой перебор:

<?php

$arr = [1, 2, 3, 4, 5];

$length = count($arr);

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

echo $arr[$i];

}

?>

**Задача №1**

Дан массив:

<?php

$arr = ['a', 'b', 'c', 'd', 'e'];

?>

С помощью цикла for выведите все эти элементы на экран.

## **Применение**

Выведем на экран все элементы массива, кроме нулевого. Для этого начнем счетчик цикла не с нуля, а с единицы:

<?php

$arr = [1, 2, 3, 4, 5];

$length = count($arr);

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

echo $arr[$i];

}

?>

**Задача №2**

Дан массив:

<?php

$arr = ['a', 'b', 'c', 'd', 'e'];

?>

С помощью цикла for выведите на экран все элементы этого массива, кроме последнего.

**Задача №3**

Дан массив:

<?php

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

?>

С помощью цикла for выведите на экран первую половину элементов этого массива.

## **Распространенная ошибка**

В следующем коде подсчет элементов массива вынесен в условие цикла:

<?php

$arr = [1, 2, 3, 4, 5];

for ($i = 1; $i < count($arr); $i++) {

echo $arr[$i];

}

?>

На первый взгляд кажется, что этот код лучше, так как мы избавились от лишней переменной $length. На самом деле это не так. Дело в том, что в приведенном коде функция count будет заново подсчитывать количество элементов массива каждую интерацию цикла. Это не оптимально.

**Задача №4**

В следующем коде некоторый программист находит сумму всех элементов массива, кроме нулевого:

<?php

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

$sum = 0;

for ($i = 1; $i < count($arr); $i++) {

$sum += $arr[$i];

}

echo $sum;

?>

Что не так с этим кодом? Исправьте ошибку автора кода.

# Заполнение массивов через цикл в PHP

Пусть у нас есть какой-то массив:

<?php

$arr = [];

?>

Давайте заполним этот массив какими-нибудь элементами. Можно сделать это вот так:

<?php

$arr = [1, 2, 3, 4, 5];

?>

А можно изначально создать массив пустым, а затем добавить в него данные вот так:

<?php

$arr = [];

$arr[] = 1;

$arr[] = 2;

$arr[] = 3;

$arr[] = 4;

$arr[] = 5;

?>

**Задача №5**

Объявите пустой массив, а затем заполните его числами от 1 до 10.

**Задача №6**

Объявите пустой массив, а затем заполните его 5-ю буквами x.

## **Заполнение массива в цикле**

Пусть мы заполняем массив вот таким образом:

<?php

$arr = [];

$arr[] = 1;

$arr[] = 2;

$arr[] = 3;

$arr[] = 4;

$arr[] = 5;

?>

Очевидно, что данный способ не очень удобен, особенно если элементов в массиве очень много. Давайте перепишем наш код так, чтобы заполнением массива занимался цикл for:

<?php

$arr = [];

for ($i = 0; $i <= 4; $i++) {

$arr[] = $i + 1;

}

var\_dump($arr);

?>

**Задача №7**

С помощью цикла заполните массив числами от 1 до 100.

**Задача №8**

С помощью цикла заполните массив нечетными числами в промежутке от 1 до 99.

# Изменение массива в цикле PHP

Пусть у нас дан вот такой массив с числами:

<?php

$arr = [1, 2, 3, 4, 5];

?>

Давайте увеличим каждый элемент этого массива в два раза:

<?php

$arr = [1, 2, 3, 4, 5];

$arr[0] = $arr[0] \* 2;

$arr[1] = $arr[1] \* 2;

$arr[2] = $arr[2] \* 2;

$arr[3] = $arr[3] \* 2;

$arr[4] = $arr[4] \* 2;

var\_dump($arr); // выведет [2, 4, 6, 8, 10]

?>

Давайте теперь то же самое выполним в цикле:

<?php

$arr = [1, 2, 3, 4, 5];

$length = count($arr);

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

$arr[$i] = $arr[$i] \* 2;

}

var\_dump($arr); // выведет [2, 4, 6, 8, 10]

?>

**Задача №9**

Дан массив с числами. Переберите этот массив циклом и возведите каждый элемент этого массива в квадрат.

## **Сокращенные операции**

При изменениях массивов можно использовать сокращенные операции, применяя их прямо к элементу массива (то есть к $arr[i]). Давайте для примера увеличим каждый элемент массива на единицу, используя операцию ++:

<?php

$arr = [1, 2, 3, 4, 5];

$length = count($arr);

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

$arr[$i]++;

}

var\_dump($arr); // выведет [2, 3, 4, 5, 6]

?>

А теперь увеличим каждый элемент массива на 5, используя операцию +=:

<?php

$arr = [1, 2, 3, 4, 5];

$length = count($arr);

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

$arr[$i] += 5;

}

var\_dump($arr); // выведет [6, 7, 8, 9, 10]

?>

**Задача №10**

Дан массив с числами. Переберите этот массив циклом и отнимите от каждого элемента единицу.

**Задача №11**

Дан массив с числами. Переберите этот массив циклом и прибавьте к каждому элементу 10.

# Практика на массивы в цикле в PHP

**Задача №12**

Дан следующий массив с работниками и их зарплатами:

<?php

$arr = [

'employee1' => 100,

'employee2' => 200,

'employee3' => 300,

'employee4' => 400,

'employee5' => 500,

'employee6' => 600,

'employee7' => 700,

];

?>

Увеличьте зарплату каждого работника на 10%.

**Задача №13**

Модифицируйте предыдущую задачу так, чтобы зарплата увеличивалась только тем работникам, у которых она меньше или равна 400.

**Задача №14**

Дан следующий массив:

<?php

$arr = [1 => 6, 2 => 7, 3 => 8, 4 => 9, 5 => 10];

?>

Найдите сумму ключей этого массива и поделите ее на сумму значений.

**Задача №15**

Дан следующий массив:

<?php

$arr = ['a' => 1, 'b' => 2, 'c' => 3, 'd' => 4, 'e' => 5];

?>

Запишите ключи этого массива в один массив, а значения - в другой.

**Задача №16**

Дан следующий массив:

<?php

$arr = [

1 => 125,

2 => 225,

3 => 128,

4 => 356,

5 => 145,

6 => 281,

7 => 452,

];

?>

Запишите в новый массив элементы, значение которых начинается с цифры 1 или цифры 2. То есть у вас в результате получится вот такой массив:

<?php

[

125,

225,

128,

145,

281,

];

?>