**Лабораторная работа №9**

**Порядок элементов и длинна массива**

Как вы уже знаете, в обычных массивах элементы располагаются в строгом порядке, ведь позиция каждого элемента определяет его ключ. В ассоциативных массивах мы сами назначаем ключи, поэтому порядок следования элементов не имеет значения.

Для примера рассмотрим вот такой ассоциативный массив:

<?php

$arr = [1 => 'value1', 2 => 'value2', 3 => 'value3'];

echo $arr[1]; // выведет 'value1'

echo $arr[2]; // выведет 'value2'

echo $arr[3]; // выведет 'value3'

?>

Если переставить элементы этого массива в произвольном порядке (конечно же, вместе с их ключами), то ничего от этого в работе нашего скрипта не изменится:

<?php

$arr = [3 => 'value3', 1 => 'value1', 2 => 'value2'];

echo $arr[1]; // выведет 'value1'

echo $arr[2]; // выведет 'value2'

echo $arr[3]; // выведет 'value3'

?>

Кроме того, числовые ключи не обязательно должны иметь все значения без дырок. У нас могут быть произвольные числа и это не будет приводить ни к каким проблемам:

<?php

$arr = [7 => 'value1', 50 => 'value2', 23 => 'value3'];

?>

**Длина массива в PHP**

Длина массива находится с помощью функции count:

<?php

$arr = [1, 2, 3];

echo count($arr); // выведет 3

?>

## Последний элемент массива

Последний элемент массива также получается аналогично последнему символу строки:

<?php

$arr = [1, 2, 3];

echo $arr[count($arr) - 1]; // выведет 3

?>

**Изменение элементов массива в PHP**

Элементы массивов можно изменять подобно изменению символов строки:

<?php

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

$arr[0] = '!';

var\_dump($arr); // выведет ['!', 'b', 'c']

?>

## Перезапись элементов

Можно также прочитывать текущее значение элемента, выполнять с ним какие-то операции и записывать измененное значение обратно в этот элемент:

<?php

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

$arr[0] = $arr[0] . '!';

$arr[1] = $arr[1] . '!';

$arr[2] = $arr[2] . '!';

var\_dump($arr); // выведет ['a!', 'b!', 'c!']

?>

Предыдущий код можно переписать через оператор .=:

<?php

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

$arr[0] .= '!';

$arr[1] .= '!';

$arr[2] .= '!';

var\_dump($arr); // выведет ['a!', 'b!', 'c!']

?>

## Инкремент и декремент

Можно также применять операции инкремента и декремента:

<?php

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

$arr[0]++;

++$arr[1];

$arr[2]--;

--$arr[3];

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

?>

**Заполнение массивов в PHP**

Элементы в массив не обязательно добавлять сразу в момент объявления этого массива. Можно вначале объявить этот массив пустым, а затем добавить в него необходимые элементы, вот так:

<?php

$arr = []; // создаем пустой массив

$arr[] = 'a'; // элемент добавится в ключ 0

$arr[] = 'b'; // элемент добавится в ключ 1

$arr[] = 'c'; // элемент добавится в ключ 2

var\_dump($arr); // выведет ['a', 'b', 'c']

?>

Массив не обязательно должен быть изначально пустым - там уже что-то может быть, но мы все равно можем добавлять новые элементы:

<?php

$arr = ['a', 'b', 'c']; // объявляем массив с элементами

$arr[] = 'd'; // элемент добавится в ключ 3

$arr[] = 'e'; // элемент добавится в ключ 4

var\_dump($arr); // выведет ['a', 'b', 'c', 'd', 'e']

?>

## Явное указание ключей

При необходимости ключи можно указать явным образом:

<?php

$arr = []; // создаем пустой массив

$arr[0] = 'a'; // в ключ 0 добавим элемент 'a'

$arr[1] = 'b'; // в ключ 1 добавим элемент 'b'

$arr[2] = 'c'; // в ключ 2 добавим элемент 'c'

var\_dump($arr); // выведет ['a', 'b', 'c']

?>

Ключи могут быть не только числовыми, но и строковыми:

<?php

$arr = [];

$arr['a'] = 1;

$arr['b'] = 2;

$arr['c'] = 3;

var\_dump($arr); // выведет ['a' => 1, 'b' => 2, 'c' => 3]

?>

**Ключи из переменных в PHP**

Пусть у нас дан вот такой массив:

<?php

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

?>

Выведем на экран элемент с ключом 0:

<?php

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

echo $arr[0]; // выведет 'a'

?>

Давайте теперь ключ выводимого элемента не будем писать непосредственно в квадратных скобках, а запишем его в переменную:

<?php

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

$key = 0; // запишем ключ в переменную

?>

Используем теперь нашу переменную для вывода соответствующего элемента:

<?php

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

$key = 0; // запишем ключ в переменную

echo $arr[$key]; // выведет 'a'

?>

**Задания для самостоятельного выполнения:**

**~~Задача №1 (НЕ ДЕЛАТЬ!)~~**

~~Напишите один большой и работающий ассоциативный массив, который будет обрабатывать все операции, описанные выше.~~

**Задача №2**

Пусть дан такой массив:

<?php

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

?>

Выведите на экран количество элементов в этом массиве.

**Задача №3**

Пусть дан такой массив:

<?php

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

?>

Выведите на экран количество элементов в этом массиве.

**Задача №4**

Пусть дан такой массив:

<?php

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

?>

Выведите на экран последний элемент этого массива.

**Задача №5**

Модифицируйте предыдущую задачу так, чтобы выводился предпоследний элемент массив.

**Задача №6**

Дан следующий массив:

<?php

$arr = ['a' => 1, 'b' => 2, 'c' => 3];

?>

Измените значение каждого из элементов этого массива.

**Задача №7**

Дан следующий массив:

<?php

$arr = ['a' => 1, 'b' => 2, 'c' => 3];

?>

Прибавьте к каждому элементу массива число 3. Выведите на экран измененный массив.

**Задача №8**

Дан следующий код:

<?php

$arr = ['a' => 1, 'b' => 2, 'c' => 3];

$arr['a']++;

$arr['a']++;

$arr['b']--;

$arr['c']--;

$arr['c']--;

var\_dump($arr);

?>

Не запуская код, расскажите, каким будет результат функции var\_dump.

**Задача №9**

Пусть дан пустой массив:

<?php

$arr = [];

?>

Описанным способом заполните этот массив элементами со значениями 1, 2, 3, 4 и 5.

**Задача №10**

Пусть дан такой массив:

<?php

$arr = [1, 2, 3];

?>

Добавьте ему в конец элементы 4 и 5.

**Задача №11**

Пусть дан пустой массив:

<?php

$arr = [];

?>

Описанным способом в ключ 'year' запишите текущий год, в ключ 'month' - текущий месяц, а в ключ 'day' - текущий день.

**Задача №12**

Дан следующий массив:

<?php

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

?>

Дана также переменная $key1 со значением 1 и переменная $key2 со значением 2. Найдите сумму элементов, ключи которых хранятся в наших переменных.

**Задача №13**

Дан следующий массив:

<?php

$arr = ['a' => 1, 'b' => 2, 'c' => 3];

?>

Дана переменная $key, в которой хранится один из ключей нашего массива. Выведите с помощью этой переменной соответствующий элемент массива.

**Задача №14**

Создайте многомерный массив $arr. С его помощью выведите на экран слова 'joomla', 'drupal', 'зеленый', 'красный'.

$arr **=** [

'cms'**=>**['joomla', 'wordpress', 'drupal'],

'colors'**=>**['blue'**=>**'голубой', 'red'**=>**'красный', 'green'**=>**'зеленый']

];

**Задача №15**

Создайте 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. Выведите с помощью этого массива понедельник по-русски и среду по-английски (пусть понедельник - это первый день).

**Задача №16**

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. Выведите словом день недели, соответствующий переменным $lang и $day. То есть: если, к примеру, $lang = 'ru' и $day = 3 - то выведем 'среда'.