

#8 – Массивы данных. Одномерные и многомерные

За урок мы научимся работать с массивами данных в языке PHP. При помощи массивов можно хранить большие объемы информации в одном единственном месте. Мы изучим одномерные и многомерные массивы данных.

Массивы позволяют хранить большой объем информации в одном месте. В языке PHP можно найти несколько основных типов массивов.

Одномерный массив

Одномерный массив состоит из нескольких элементов, объединенных под одним именем. Чтобы создать массив необходимо создать переменную и присвоить ей значение `array()`. Внутри вы можете через запятую записывать различные значения, которые и будут элементами массива.

В массивах отсчет начинается с 0, поэтому первый элемент по индексу будет равен 0, второй - 1 и так далее.

Примеры создания массива:

```
$some = array(); // Создание пустого массива
$some[0] = '1'; // Добавление 1 элемента
$some[1] = 2; // Добавление 2 элемента
echo $some[0]; // Вывод первого элемента
$array = array(1, 5, 2, "Some", true); // Создание массива со значениями сразу же
```

Работать с элементами массива можно точно как с переменными. Мы можем их выводить или же устанавливать для них новые значения.

Для массивов существует несколько дополнительных функций, которые ещё будут изучены в ходе курса. Сейчас был изучен лишь метод `count()`, который позволяет получить длину всего массива.

Пример метода:

```
$elements = array(23, 6, 0, true, "Первый");
// Выведет значение 5, так как в массиве 5 элементов
echo count($elements);
```

Ассоциативные массивы

Ассоциативный массив отличается от обычного лишь индексами. Здесь вместо чисел можно подставить, так называемый ключ, который будет

отвечать за определённый элемент в массиве. Название для ключей может быть каким-угодно.

Пример массива:

```
$list = array ("first" => 1, "second" => "2");  
// Вместо числовых индексов здесь идут слова  
echo $list["second"];
```

Многомерные массивы

Многомерный массив - это массив, в котором каждый элемент является другим массивом. На практике очень редко используются массивы с более чем третьим уровнем вложенности. То есть массивы, в которых все элементы являются другими массивами и в котором все элементы также другие массивы встречаются очень и очень редко на практике.

Мы не будем изучать подобные массивы, так как принцип их построения точно такой же как при работе с двумерными массивами.

Для создания двумерных массивов необходимо использовать двойные квадратные скобки после типа данных. Выглядит такой массив как матрица, а записывается следующим образом:

```
$x = array(array(0, 34, 2), array(3, 4, 5));  
echo $x[0][1]; // Выведет 34  
  
// Можно их сразу не присваивать  
$symbols = array(array(), array());  
$symbols[0][1] = 'A';
```

Мы видим, что элементы первого массива являются другими массивами. Чтобы выбрать какой-либо объект используйте ту же структуру что и для одномерных массивов, вот только теперь указывайте индекс как первого массива, так и второго:

```
$x[0][1] = 1; // Вместо 34 теперь будет 1
```

Исходный код

```

<?php
    // Одномерные массивы
    $nums = array(4, 5, 7, 2, 0, -23, 6);
    $nums[1] = 45;
    echo $nums[1]. '<br>';

    $arr = [4, true, 6, "8", 0.4, 'c', 24, 16];
    $arr[0] = "false";
    echo $arr[0]. '<br>';

    // Ассоциативные массивы
    $list = ["age" => 50, "name" => "Alex", "hobby" => "Football"];
    $list["name"] = "Bob";
    echo $list["name"]. '<br>';

    // Многомерные массивы
    $matrix = [
        [4, 6.4, 8],
        [3, 2],
        [1, 5, 8, "9"]
    ];

    $matrix[0][1] = 4;
    echo $matrix[0][1];
?>

```

Самостоятельная работа

Поиск одинаковых чисел в массивах

Дано два массива чисел:

```
$array_1 = array (1, 2, 3, 5, 7, 9, 10);
```

```
$array_2 = array (1, 4, 3, 5, 8, 9, 0);
```

Необходимо вывести на экран все числа, которые одновременно присутствуют в обоих массивах.

Сумма чисел

Есть массив:

```
$elements = array(3, -2, 9, -34, 23, -12, 1);
```

Найдите сумму только положительных элементов массива.

Максимальный элемент

Создайте массив из 7 чисел. При помощи цикла for найдите максимальный элемент в массиве.

Минимальный элемент

Создайте двумерный массив и найдите минимальный элемент среди всех чисел в массиве. Массив ниже:

```
[[ 0, 34, -2 ], [ 3, -4, 5 ]]
```

Вывод массива

Выведите двумерный массив на экран при помощи циклов for. Оформите вывод данных в виде матрицы! Вот сам массив:

```
$numbers = array(  
    array(2, 45, 91),  
    array(90, 47, 82),  
    array(10, 5, 32),  
    array(8, 78, 30),  
    array(25, 1, 27),  
    array(38, 82, 8)  
);
```