

Основы работы с массивами в PHP

В программировании очень часто возникает задача хранения списка похожих значений, например, всех дней недели или всех месяцев. Можно было бы создавать под каждое значение списка свою переменную, но это очень неудобно и долго - для списка дней недели понадобилось бы 7 переменных, а для месяцев - вообще 12.

А теперь представьте, что вам нужно вывести на экран название третьего месяца. С помощью 12-ти различных переменных это было бы проблематично, так как вам необходимо помнить названия всех переменных, в которые вы записали имена месяцев.

Поэтому для таких вещей был изобретен специальный тип данных. Он называется **массив**.

Массив создается с помощью функции []:

```
<?php  
    $a = []; //создаем массив $a  
?>
```

Пока созданный нами массив не содержит никаких данных. Заполним его названиями дней недели (для краткости пишу две буквы каждого дня):

```
<?php  
    $a = ['пн', 'вт', 'ср', 'чт', 'пт', 'сб', 'вс'];  
?>
```

Каждое значение списка, который мы записали в массив (в нашем случае каждый день недели), называется **элементом массива**.

Элементы разделяются между собой запятой. После этой запятой можно ставить пробелы, а можно и не ставить.

Обратите внимание на то, что названия дней недели представляют собой строки и поэтому **взяты в кавычки**. Кроме строк в массиве можно хранить числа, и их в кавычки мы не берем:

```
<?php  
    //В массиве можно хранить как строки, так и числа:  
    $a = ['пн', 256, 'ср', 34, 38, 'сб', 95];  
?>
```

Посмотреть содержимое массива с помощью echo мы, увы, не сможем. Попробуйте сделать так:

```
<?php  
    //Пробуем вывести с помощью echo содержимое массива $a:  
    $a = ['пн', 'вт', 'ср', 'чт', 'пт', 'сб', 'вс'];  
    echo $a;  
?>
```

Вы увидите на экране слово Array вместо содержимого массива.

Для того, чтобы PHP вывел нам все элементы массива, нужно воспользоваться функцией var_dump:

```
<?php  
    //Пробуем вывести с помощью var_dump содержимое массива $a:  
    $a = ['пн', 'вт', 'ср', 'чт', 'пт', 'сб', 'вс'];  
    var_dump($a);  
?>
```

Вот теперь мы действительно увидим все элементы нашего массива.

Запомните этот момент: для того, чтобы узнать, какие именно элементы хранятся в массиве, мы должны пользоваться функцией var_dump, а не echo.

Как вывести отдельный элемент массива

Итак, массив месяцев мы составили и даже научились смотреть на него функцией var_dump. Однако, полезного пока мало, так как var_dump используется для отладки кода и выводит весь массив целиком.

Предположим, мы хотим вывести на экран среду.

Делается это так: нужно после переменной массива (в нашем случае \$a) написать квадратные скобки [], а в них указать порядковый номер элемента, который мы хотим вывести: \$a[3]. Казалось бы, что порядковый номер среды - это 3, но это не так. Потому что в программировании нумерация начинается с нуля. Поэтому 0 - это понедельник, 1 - это вторник, а 2 - это среда.

Посмотрите и повторите пример:

```
<?php
    //Выведем слово 'ср':
    $a = ['пн', 'вт', 'ср', 'чт', 'пт', 'сб', 'вс'];
    echo $a[2];
?>
```

Ассоциативный массив

Чтобы обратиться к нужному элементу массива, мы писали в квадратных скобках его порядковый номер (нумерация начинается с нуля, если вы уже забыли). Эти порядковые номера называются **ключами массива**. То есть мы получали значение элемента массива по его ключу.

В нашем случае PHP сам определял ключи для элементов - это были их порядковые номера. Иногда это может оказаться неудобным - мы хотим вывести на экран название третьего дня недели (среду), а должны писать в квадратных скобках цифру 2.

Поэтому в PHP можно указать ключи в явном виде – так, как нам нужно. Сделаем так, чтобы понедельник имел ключ 1, а не ноль, как было раньше (и всем остальным дням прибавим единицу):

```
<?php
    //Укажем ключи в явном виде:
    $a = [1=>'пн', 2=>'вт', 3=>'ср', 4=>'чт', 5=>'пт', 6=>'сб', 7=>'вс'];
    echo $a[1]; //выведет 'пн'
?>
```

Синтаксис здесь такой: ключ, затем идет стрелка =>, а потом значение.

Ключи не обязательно должны быть числами, они могут быть и строками. Сделаем массив, в котором ключами будут имена работников, а элементами - их зарплаты:

```
<?php
    //Массив работников:
    $a = ['Коля'=>200, 'Вася'=>300, 'Петя'=>400];
?>

Узнаем зарплату Васи:
<?php
    $a = ['Коля'=>200, 'Вася'=>300, 'Петя'=>400];
    echo $a['Вася']; //выведет 300
?>
```

Массивы, у которых явно указаны ключи, называются **ассоциативными**.

Хитрость с ключами

Когда мы делали ассоциативный массив дней недели, нам приходилось расставлять все ключи вручную. И все для того, чтобы нумерация началась не с нуля, а с единицы. Это было немного неудобно.

Напомню вам этот массив:

```
<?php
    //Укажем ключи в явном виде:
    $a = [1=>'пн', 2=>'вт', 3=>'ср', 4=>'чт', 5=>'пт', 6=>'сб', 7=>'вс'];
    echo $a[3]; //выведет 'ср'
?>
```

На самом деле нет необходимости расставлять ключи всем элементам - достаточно только первому элементу поставить ключ 1.

Если у второго элемента не будет ключа, PHP поставит его автоматически, причем следующий по порядку.

А следующим номером будет как раз-таки число 2, так как предыдущий элемент имел ключ 1(неважно, что мы сами его поставили, а не PHP автоматически).

Давайте поправим наш массив:

```
<?php
    //Укажем ключи в явном виде:
    $a = [1=>'пн', 'вт', 'ср', 'чт', 'пт', 'сб', 'вс'];
?>
```

```
echo $a[3]; //выведет 'ср'
?>
```

Итак, этот массив с одним нашим ключом фактически будет таким же, как и массив со всеми ключами, но сделать его проще - ключ ставится только первому элементу, а не всем.

Как еще можно создать массив

Объявление массива с помощью команды [] не является единственным способом его создания.

Можно просто присвоить значения элементам массива, не объявляя его через array (PHP нас поймет и сам создаст массив):

```
<?php
    //Можно не писать [], а просто присвоить значения его элементам:
    $a[0] = 1;
    $a[1] = 2;
    $a[2] = 3;
    var_dump($a); //с помощью var_dump убеждаемся в том, что $a - это массив
?>
```

Естественно, ключи могут быть не только числовыми, но и текстовыми:

```
<?php
    //Ключи могут быть строками:
    $a['Коля'] = 100;
    $a['Вася'] = 200;
    $a['Петя'] = 300;
    var_dump($a); //с помощью var_dump убеждаемся в том, что $a - это массив
?>
```

Кроме того, можно сделать так, что PHP сам добавит ключи (начиная с нуля и так далее). Для этого мы оставим квадратные скобки пустыми: \$a[] = 1, а PHP сам добавит ключ. Пример:

```
<?php
    //Мы не пишем ключи в квадратных скобках
    $a[] = 100; //100 будет иметь ключ 0
    $a[] = 200; //200 будет иметь ключ 1
    $a[] = 300; //300 будет иметь ключ 1
    var_dump($a); //массив $a будет иметь вид [0=>100, 1=>200, 2=>300]
?>
```

Все способы создания массива

Итак, повторим все способы создания массива:

```
<?php
    //Можно просто его объявить:
    $a = [];

    //Можно его сразу и заполнить:
    $a = [1, 2, 3];

    //Можно не писать [], а просто присвоить значения его элементам:
    $a[0] = 1;
    $a[1] = 2;
    $a[2] = 3;
```

```
//Можно даже не писать ключи, тогда PHP добавит их автоматически, начиная с нуля:  
$a[] = 1;  
$a[] = 2;  
$a[] = 3;  
?>
```

Многомерный массив

Элементы массива могут быть не только строками и числами, но и массивами. То есть у нас получится массив массивов или многомерный массив.

Давайте сделаем массив студентов \$students, который будет содержать два подмассива: студенты мужского пола и женского:

```
<?php  
//Многомерный массив студентов:  
$a = [  
    'boys' => ['Коля', 'Вася', 'Петя'],  
    'girls' => ['Даша', 'Маша', 'Лена'],  
];  
?>
```

Чтобы вывести какой-либо элемент из многомерного массива следует писать уже не одну пару [], а две: \$a['boys'][0] – так мы выведем 'Коля'.

Задачи на основы работы с массивами в PHP

Примеры решения задач

Задача

Задача. Дан массив с элементами 'Привет', 'мир' и '!'. Необходимо вывести на экран фразу 'Привет, мир!'.

Решение:

```
<?php  
//Выведем фразу 'Привет, мир!':  
$arr = ['Привет', ' ', 'мир', '!'];  
echo $arr[0].$arr[1].$arr[2];  
?>
```

Разберем это решение.

Слово 'Привет,' хранится под номером 0, это значит, что для доступа к нему мы должны написать \$arr[0].

Для доступа к слову 'мир' мы должны написать \$arr[1], а \$arr[2] содержит в себе '!'. Далее с помощью оператора 'точка' мы сложим три наши строки ('Привет, ', 'мир' и '!') в одну строку таким образом: \$arr[0].\$arr[1].\$arr[2], и выведем на экран с помощью echo.

Задача

Задача. Решим немного другую задачу: дан массив с элементами 'Привет', ' ', 'мир' и '!'. Необходимо записать в переменную \$text фразу 'Привет, мир!', а затем вывести на экран содержимое этой переменной.

Решение:

```
<?php  
$arr = ['Привет', ' ', 'мир', '!'];  
$text = $arr[0].$arr[1].$arr[2];
```

```

/*
    В переменной $text теперь лежит строка 'Привет, мир!'.
    Выведем ее на экран:
*/
echo $text;
?>

```

Задача

Задача. Дан массив ['Привет, ', 'мир', '!']. Необходимо записать в первый элемент (то есть элемент с номером ноль) этого массива слово 'Пока, ' (то есть вместо слова 'Привет, ' будет 'Пока, ').

Решение:

```

<?php
$arr = ['Привет, ', 'мир', '!'];
$arr[0] = 'Пока, '; //перезапишем первый элемент массива
var_dump($arr); //посмотрим на массив и убедимся в том, что он изменился
?>

```

Задача . Ассоциативные массивы

Задача. Создайте массив заработных плат \$arr. Выведите на экран зарплату Пети и Коли.

```

<?php
//Этот массив дан:
$arr = ['Коля'=>'1000$', 'Вася'=>'500$', 'Петя'=>'200$'];
?>
Решение: чтобы вывести зарплату Коли следует вывести значение элемента массива с ключом 'Коля'. Сделаем это:
<?php
$arr = ['Коля'=>'1000$', 'Вася'=>'500$', 'Петя'=>'200$'];
echo $arr['Коля']; //выведет 1000$
?>

```

Задача . Способы создания массива

Задача. Создайте массив \$arr с элементами 1, 2, 3, 4, 5 двумя различными способами.

Решение:

Первый способ создать массив - объявить его через []:

```

<?php
$arr = [1, 2, 3, 4, 5];
?>

```

Второй способ создания массива - это поступить таким образом:

```

<?php
$arr[] = 1;
$arr[] = 2;
$arr[] = 3;
$arr[] = 4;
$arr[] = 5;
?>

```

Задача . Многомерный массив

Задача. Дан многомерный массив \$arr:

```
$arr = [

```

```
'ru'=>['голубой', 'красный', 'зеленый'],
'en'=>['blue', 'red', 'green'],
];
```

Выведите с его помощью слово 'голубой'.

Решение: так как массив многомерный (в нашем случае двухмерный), то нам придется написать несколько квадратных скобок подряд. Поясню это по шагам. Давайте сделаем так:

```
<?php
    var_dump($arr['ru']);
?>
```

В этом случае результатом будет массив ['голубой','красный', 'зеленый'], который является частью нашего многомерного массива. Чтобы вывести слово 'голубой', на необходимо дописать еще одну квадратную скобку с ключом, соответствующим этому элементу (у него нет явного ключа - значит его ключ 0):

```
<?php
    echo $arr['ru'][0]; //выведет 'голубой'
?>
```

Выведем теперь слово 'красный':

```
<?php
    echo $arr['ru'][1]; //выведет 'красный'
?>
```

Выведем 'red':

```
<?php
    echo $arr['en'][1]; //выведет 'red'
?>
```

Задачи для решения

Работа с массивами

- Создайте массив \$arr=['a', 'b', 'c']. Выведите значение массива на экран с помощью функции var_dump().
- С помощью массива \$arr из предыдущего номера выведите на экран содержимое первого, второго и третьего элементов.
- Создайте массив \$arr=['a', 'b', 'c', 'd'] и с его помощью выведите на экран строку 'a+b, c+d'.
- Создайте массив \$arr с элементами 2, 5, 3, 9. Умножьте первый элемент массива на второй, а третий элемент на четвертый. Результаты сложите, присвойте переменной \$result. Выведите на экран значение этой переменной.
- Заполните массив \$arr числами от 1 до 5. Не объявляйте массив, а просто заполните его присваиванием \$arr[] = новое значение.

Ассоциативные массивы

- Создайте массив \$arr. Выведите на экран элемент с ключом 'c'.

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

- Создайте массив \$arr. Найдите сумму элементов этого массива.

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

- Создайте массив заработных плат \$arr. Выведите на экран зарплату Пети и Коли.

```
$arr = ['Коля'=>'1000$', 'Вася'=>'500$', 'Петя'=>'200$'];
```

- Создайте ассоциативный массив дней недели. Ключами в нем должны служить номера дней от начала недели (понедельник - должен иметь ключ 1, вторник - 2 и т.д.). Выведите на экран текущий день недели.

- Пусть теперь номер дня недели хранится в переменной \$day, например там лежит число 3. Выведите день недели, соответствующий значению переменной \$day.

Многомерные массивы

- 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' ]  
];
```

12. 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).

13. 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'.