**Создание массивов в PHP**

Массив - это набор данных, которые объединены под одним именем. Массив состоит из нескольких элементов, которые имеют свой определенный индекс.

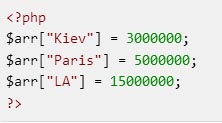
Массивы создаются при помощи [оператора присвоения](http://php720.com/lesson/9), также как и переменная.   
Имена массивов начинаются со знака **$**, после которого следует произвольный идентификатор, далее идут квадратные скобки: **$**arr[0] = "php";

Данная конструкция создает массив и присваивает его элементу с индексом 0 значение "php", после чего мы можем обращаться к этому элементу как к обычной переменной: echo $arr[0]. В результате мы увидим слово *php*.

Также, мы можем добавить еще элементы к массиву:

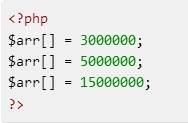


В качестве индекса элементов массива мы можем использовать не только числа:



В качестве значений индексов элементов и самих элементов мы можем использовать одинаковые [типы данных](http://php720.com/lesson/6) **одновременно**!

Также существует сокращенная запись для индексирования:



В этом случае первый элемент (3000000) получит индекс 0! Нужно иметь это ввиду.

**Для создания массива**

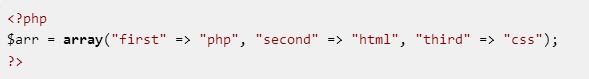
Для создания массива мы можем использовать функцию array:



В этом случае первый элемент получит индекс 0. Если нужно присвоить какой-то другой номер, то можно воспользоваться конструкцией =>:



Теперь элемент под номером 1 это "php", а не "html"! Также, можно создать массив со строковым индексом:



Начиная с версии PHP 5.4, массивы можно создать через квадратные скобки:



# Модификация элементов массива в PHP

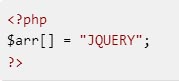
Нередко, нужно изменить значение того, или иного элемента массива, для этого нужно всего лишь обратится к элементу массива по его индексу и присвоить ему новое значение.   
Например, у нас есть массив:



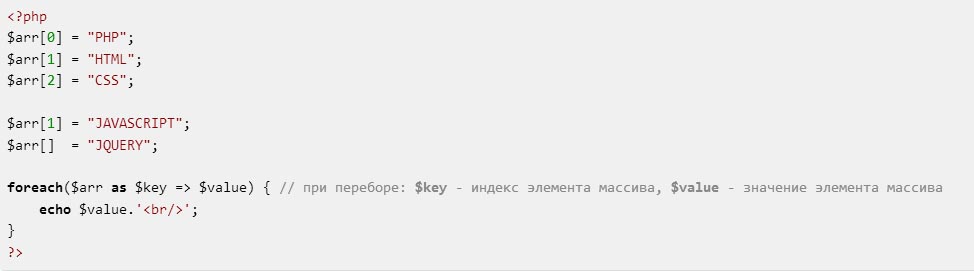
Для того чтобы изменить значение элемента, используем [оператор присвоения](http://php720.com/lesson/9):



Для того чтобы добавить новый элемент в конец массива, использует конструкцию:



Для того чтобы вывести на экран массив, можно использовать [foreach](http://php720.com/lesson/24):



Отображение в браузере:

PHP

JAVASCRIPT

CSS

JQUERY

**Удаление элементов массива в PHP**

Если нам нужно удалить один из элементов массива, то для этого мы должны использовать функцию unset



# Перебор элементов массива в PHP

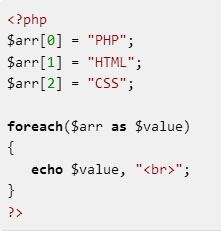
Кроме использования цикла [for](http://php720.com/lesson/21) для вывода всех элементов массива на экран мы можем использовать функцию print\_r, которая выведет все элементы массива вместе с их индексами.



Отображение в браузере:

Array ( [0] => PHP [1] => HTML [2] => CSS )

Также в PHP присутствует специальный цикл для обработки массивов - цикл [foreach](http://php720.com/lesson/24)



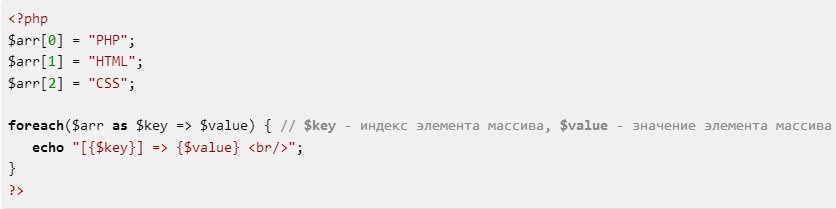
Отображение в браузере:

PHP

HTML

CSS

Для вывода индекса элемента нужно использовать второй вариант синтаксиса цикла foreach



Отображение в браузере:

[0] => PHP

[1] => HTML

[2] => CSS

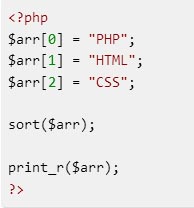
**Функции для работы с массивами в PHP**

Для работы с массивами в PHP предусмотрено очень много функций, полный список которых находиться ниже:

|  |  |
| --- | --- |
| array\_chunk | Разбивает массив на несколько меньших массивов заданного размера |
| array\_combine | Создает массив из двух заданных массивов - массива индексов элементов и массива значений |
| array\_count\_values | Формирует массив, индексами которого являются значения заданного массива, а значениями - число повторений соответствующего значения в заданном массиве |
| array\_diff | Формирует массив из тех элементов первого заданного массива, которые отсутствуют в остальных заданных в качестве аргументов функции массива |
| array\_fill | Заполняет массив заданным значением |
| array\_intersect | Формирует массив из элементов, которые присутствуют во всех заданных массивах |
| array\_key\_exists | Проверяет наличие заданного индекса в массиве |
| array\_keys | Возвращает массив из индексов заданного массива |
| array\_merge | Объединяет несколько массивов в один |
| array\_multisort | Выполняет сортировку многомерного массива или нескольких одномерных массивов |
| array\_pad | Дополняет массив до заданного количества элементов заданным значением |
| array\_pop | Возвращает последний элемент массива, одновременно удаляя элемент из массива |
| array\_product | Возвращает произведение элементов массива |
| array\_push | Добавляет заданные элементы в конец массива |
| array\_rand | Выбирает один или несколько случайно взятых элементов из массива |
| array\_reduce | Осуществляет последовательное применение заданной функции к элементам массива, формируя итоговое значение |
| array\_reverse | Производит обращение массива - первый элемент становится последним, второй - предпоследним и т.д. |
| array\_search | Ищет заданный элемент в массиве и возвращает соответствующим ему индекс |
| array\_shift | Возвращает первый элемент массива, одновременно удаляя его из массива с перенумерацией числовых индексов |
| array\_slice | Вырезает из массива подмассив заданной длины, начиная с указанного элемента |
| array\_splice | Удаляет последовательность элементов массива и заменяет её другой последовательностью |
| array\_sum | Вычисляет сумму всех элементов массива |
| array\_unique | Удаляет дублирующиеся значения из массива |
| array\_unshift | Добавляет один или несколько элементов в начало массива с перенумерацией числовых индексов |
| array\_walk | Вызывает заданную функцию последовательно для каждого элемента массива |
| array | Создает массив из заданных значений или пар индекс - значение |
| arsort | Сортирует массив по убыванию его значений, сохраняя индексы неизменными |
| asort | Сортирует массив по возрастанию его значений, сохраняя индексы неизменными |
| krsort | Сортирует массив по убыванию его индексов |
| ksort | Сортирует массив по возрастанию его индексов |
| sort | Сортирует массив по возрастанию значений его элементов с перенумерацией его индексов |
| usort | Сортирует массив с использование заданной функции сравнения элементов массива |
| rsort | Сортирует массив по убыванию значений его элементов с перенумерацией его индексов |
| natcasesort | Сортирует массив естественным образом без учета регистра массива |
| natsort | Сортирует массив естественным образом с учетом регистра символов |
| count | Возвращает количество элементов в массиве |
| current | Возвращает значение текущего элемента массива |
| each | Возвращает текущие индекс и значение элемента массива и продвигает указатели на следующий элемент |
| in\_array | Проверяет, присутствует ли заданное значение в массиве |
| key | Возвращает индекс текущего элемента массива |
| list | Присваивает значения из массива списку переменных |
| pos | Синоним функции *current* |
| reset | Устанавливает внутренний указатель на первый элемент массива |
| shuffle | Переставляет элементы массива случайным образом |
| sizeof | Синоним функции *count* |

# Сортировка массивов в PHP

Очень часто нужно отсортировать массив по индексу его элементов, по алфавиту его элементов, по возрастанию, по убыванию и т. д. Первая функция - sort, которая сортирует массив по возрастанию значений его элементов, при этом изменяя индекс после сортировки:



Отображение в браузере:

Array ( [0] => CSS [1] => HTML [2] => PHP )

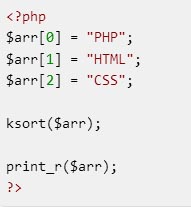
Вторая функция - rsort, которая сортирует массив по убыванию значений его элементов, при этом не изменяя индекс после сортировки:



Отображение в браузере:

Array ( [0] => PHP [1] => HTML [2] => CSS )

Третья функция - ksort, которая сортирует массив по ключам, сохраняя отношения между ключами и значениями:



Отображение в браузере:

Array ( [0] => PHP [1] => HTML [2] => CSS )

Следующая функция - krsort, которая сортирует массив по убыванию индексов его элементов:



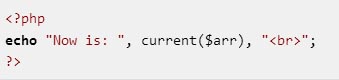
Отображение в браузере:

Array ( [2] => CSS [1] => HTML [0] => PHP )

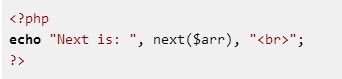
# Навигация по массивам в PHP

Навигация по массива даёт возможность узнать текущий, следующий, предыдущий, последний элемента массива.

Для определения текущего элемента массива используют функцию current:



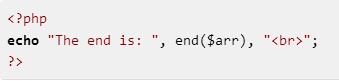
Для определения следующего элемента массива используют функцию next:



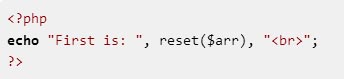
Для определения предыдущего элемента массива используют функцию prev:



Для определения последнего элемента массива используют функцию end:



Для определения первого (возврата указателя) элемента массива используют функцию reset:



Пример навигации по массивам:



Отображение в браузере:

Now is: PHP

Next is: HTML

Previously is: PHP

The end is: CSS

First is: PHP

**Преобразование строк в массивы и наоборот в PHP**

PHP умеет преобразовывать данные из строки в массив и наоборот, для этого в PHP есть функция implode и explode.

implode - формирует строку из массива.

explode - формирует массив из строки.

Использование функции implode:



Отображение в браузере:

PHP, HTML, CSS

Использование функции explode:



Отображение в браузере:

Array ( [0] => PHP [1] => HTML [2] => CSS )

# Слияние и разделение массивов в PHP

Если вам нужно добавить в массив несколько элементов другого массива, то вам поможет функция array\_slice:



Отображение в браузере:

Array ( [one] => PHP [two] => HTML )

Первый параметр функции array\_slice - это название того массива, с которого будут браться элементы; второй - начальный номер элемента (с какого элемента начинать брать элементы); третий - количество элементов выборки.

Для соединения массивов в один нужно использовать функцию array\_merge:



Отображение в браузере:

Array ( [0] => PHP [1] => HTML [2] => CSS [3] => PHOTOSHOP [4] => PAINT.NET [5] => DREAMWEAVER )

# Сравнение массивов в PHP

Если нужно проверить какой элемент совпадает в том, или ином массиве, или наоборот НЕ совпадает, то можно использовать функцию array\_diff:



Отображение в браузере:

Array ( [2] => HTML [3] => CSS )

Если нужно, чтобы сравнивался еще и строковый индекс массивов, то тут уже нужно использовать функцию array\_diff\_assoc:



Отображение в браузере:

Array ( [one] => PHP [three] => CSS )

Если вам нужно найти общие элементы массивов, то вам поможет функция array\_intersect:



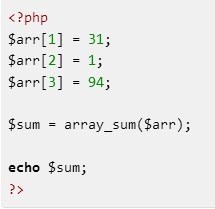
Отображение в браузере:

Array ( [1] => PHP )

Если при нахождении общих элементов массивов нужно учесть и индекс, то, по аналогии array\_diff\_assoc, используйте функцию array\_intersect\_assoc.

# Обработка данных в массивах PHP

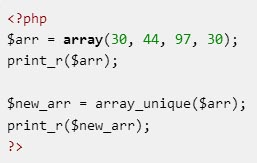
PHP имеет множество функций для обработки данных в массивах. Например, если нужно вычислить сумму всех элементов массив, то можно воспользоваться функцией array\_sum:



Отображение в браузере:

126

Если, требуется избавиться от элементов с повторяющимся значением, то стоит использовать функцию array\_unique:



Отображение в браузере:

Array ( [0] => 30 [1] => 44 [2] => 97 [3] => 30 )

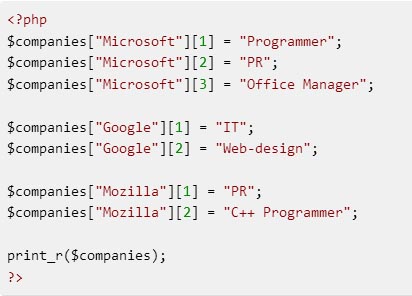
Array ( [0] => 30 [1] => 44 [2] => 97 )

# Многомерные массивы в PHP

Кроме одномерных в PHP есть еще и многомерные массивы. Это такие массивы, какие имеют под одним индексом два или более значений. Например, есть массив companies, который имеет информацию об предложениях по работе, от разных компаний:



А если компания имеет несколько предложений, то тут уже нужно использовать многомерные массивы:



Отображение в браузере:

Array (

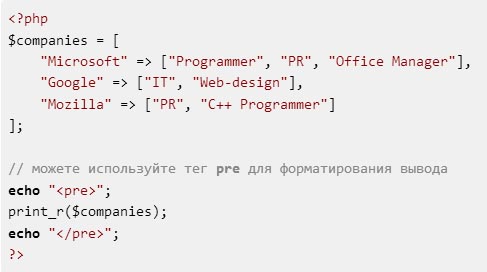
[Microsoft] => Array ( [1] => Programmer [2] => PR [3] => Office Maneger )

[Google] => Array ( [1] => IT [2] => Web-design )

[Mozilla] => Array ( [1] => PR [2] => C++ Programmer )

)

Также, эту конструкцию можно было записать так:



Отображение в браузере:

Array

(

[Microsoft] => Array

(

[0] => Programmer

[1] => PR

[2] => Office Manager

)

[Google] => Array

(

[0] => IT

[1] => Web-design

)

[Mozilla] => Array

(

[0] => PR

[1] => C++ Programmer

)

)

# Использование циклов в многомерных массивах в PHP

Если нам нужно перебрать (вывести на экран) элементы обычного массива, то мы можем использовать [циклы](http://php720.com/lesson/21). Мы также можем использовать циклы и для многомерных массивов, только нужно немножко подправить код. Рассмотрим пример:



[Переменная](http://php720.com/lesson/3) $i отвечает за счет в первых квадратных скобках массива, а переменная $j за счет во вторых квадратных скобках массива. Второй [цикл FOR](http://php720.com/lesson/21) работает, пока в массиве не закончатся элементы определенного индекса.