

```
<?
МАССИВЫ

Предопределенные константы

Перечисленные ниже константы всегда доступны как часть ядра PHP.

CASE_LOWER (integer)
CASE_LOWER используется с array_change_key_case() для конвертации ключей массива в нижний регистр. Это действие по умолчанию для array_change_key_case().
CASE_UPPER (integer)
CASE_UPPER используется с array_change_key_case() для конвертации ключей массива в верхний регистр.

Флаги, изменяющие порядок сортировки:

SORT_ASC (integer)
SORT_ASC используется с array_multisort() для сортировки в порядке возрастания.
SORT_DESC (integer)
SORT_DESC используется с array_multisort() для сортировки в порядке убывания.

Флаги сортировки, используемые различными функциями:

SORT_REGULAR (integer)
SORT_REGULAR используется для обычного сравнения элементов массива.
SORT_NUMERIC (integer)
SORT_NUMERIC используется для сравнения элементов, как цифр.
SORT_STRING (integer)
SORT_STRING используется для сравнения элементов как строк.
SORT_LOCALE_STRING (integer)
SORT_LOCALE_STRING используется для сравнения элементов как строк на базе текущей локали. Добавлено в PHP 4.4.0 и 5.0.2.
SORT_NATURAL (integer)
SORT_NATURAL используется для сравнения элементов как строк, используя естественное упорядочивание, такое как natsort(). Добавлено в PHP 5.4.0.
SORT_FLAG_CASE (integer)
SORT_FLAG_CASE может быть объединена (побитовое ИЛИ) с SORT_STRING или SORT_NATURAL для регистронезависимой сортировки строк. Добавлено в PHP 5.4.0.

Опции фильтрации:

ARRAY_FILTER_USE_KEY (integer)
ARRAY_FILTER_USE_KEY используется в array_filter() для передачи каждого ключа в виде первого аргумента в заданную функцию. Добавлено в PHP 5.6.0.
ARRAY_FILTER_USE_BOTH (integer)
ARRAY_FILTER_USE_BOTH используется в array_filter() для передачи и значения и ключа в заданную функцию. Добавлено в PHP 5.6.0.

COUNT_NORMAL (integer)
COUNT_RECURSIVE (integer)
EXTR_OVERWRITE (integer)
EXTR_SKIP (integer)
EXTR_PREFIX_SAME (integer)
EXTR_PREFIX_ALL (integer)
EXTR_PREFIX_INVALID (integer)
EXTR_PREFIX_IF_EXISTS (integer)
EXTR_IF_EXISTS (integer)
EXTR_REFS (integer)

Сортировка массивов

В PHP есть несколько функций для сортировки массивов, на этой странице дается их общее описание.

Основные различия между функциями:
```

В одних функциях массивы **array** сортируются по ключам элементов, в других по значениям: `$array['ключ'] = 'значение'`;  
 В каких-то функциях связь между ключами и значениями после сортировки сохраняется, в каких-то нет. Это может приводить к тому, что ключи будут сбрасываться в числовые значения (0, 1, 2, ...).  
 Различия в порядке сортировки: алфавитный, возрастающий, убывающий, числовой, натуральный, случайный или определенный пользователем.  
 Примечание: Все функции сортировки модифицируют переданный массив, а не возвращают отсортированную копию.  
 Если функции определяют два элемента как равные, порядок сортировки в этом случае не определен (нестабильная сортировка)

Свойства функций сортировки	Имя функции	Сортирует по значению	Сохраняет ассоциативные да, числовые нет	Связь ключ - значение	Порядок сортировки	Пожокие функции
<b>array_multisort()</b>						
<b>asort()</b>	значению	да	настройки сортировки	<b>array_walk()</b>		
<b>arsort()</b>	значению	да	по возрастанию	<b>arsort()</b>		
<b>krsort()</b>	ключу	да	по убыванию	<b>asort()</b>		
<b>ksort()</b>	ключу	да	по убыванию	<b>ksort()</b>		
<b>natscasesort()</b>	ключу	да	натуральный	<b>asort()</b>		
<b>natsort()</b>	значению	да	чувствителен к регистру	<b>natsort()</b>		
<b>rsort()</b>	значению	нет	натуральный	<b>natscasesort()</b>		
<b>shuffle()</b>	значению	нет	по убыванию	<b>sort()</b>		
<b>sort()</b>	значению	нет	случайный	<b>array_rand()</b>		
<b>uasort()</b>	значению	да	по возрастанию	<b>uasort()</b>		
<b>uksort()</b>	ключу	да	определяется польз.	<b>uksort()</b>		
<b>usort()</b>	значению	нет	определяется польз.	<b>usort()</b>		

#### ФУНКЦИИ ДЛЯ РАБОТЫ С МАССИВАМИ

```
array_change_key_case // Меняет регистр всех ключей в массиве
array_chunk           // Разбивает массив на части
array_column          // Return the values from a single column in the input array
array_combine          // Создает новый массив, используя один массив в качестве ключей,
                      // а другой в качестве соответствующих значений
array_count_values     // Подсчитывает количество всех значений массива
array_diff_assoc       // Вычисляет расхождение массивов с дополнительной проверкой индекса
array_diff_key         // Вычисляет расхождение массивов, сравнивая ключи
array_diff_uassoc       // Вычисляет расхождение массивов с дополнительной проверкой индекса,
                      // осуществляемой при помощи callback-функции
array_diff_ukey         // Вычисляет расхождение массивов, используя callback-функцию для сравнения ключей
array_diff             // Вычислить расхождение массивов
array_fill_keys        // Создает массив и заполняет его значениями, с определенными ключами
array_fill            // Заполняет массив значениями
array_filter           // Фильтрует элементы массива с помощью callback-функции
array_flip            // Меняет местами ключи с их значениями в массиве
array_intersect_assoc   // Вычисляет схождение массивов с дополнительной проверкой индекса
array_intersect_key     // Вычислить пересечение массивов, сравнивая ключи
array_intersect_uassoc  // Вычисляет схождение массивов с дополнительной проверкой индекса,
                      // осуществляемой при помощи callback-функции
array_intersect_ukey    // Вычисляет схождение массивов, используя callback-функцию для сравнения ключей
array_intersect        // Вычисляет схождение массивов
array_key_exists        // Проверяет, присутствует ли в массиве указанный ключ или индекс
array_keys             // Возвращает все или некоторое подмножество ключей массива
array_map              // Применяет callback-функцию ко всем элементам указанных массивов
array_merge_recursive  // Рекурсивное слияние двух или более массивов
array_merge            // Сливает один или большее количество массивов
array_multisort        // Сортирует несколько массивов или многомерные массивы
array_pad             // Дополнить размер массива определенным значением до заданной величины
array_pop             // Извлекает последний элемент массива
array_product          // Вычислить произведение значений массива
array_push            // Добавляет один или несколько элементов в конец массива
array_rand            // Выбирает одно или несколько случайных значений из массива
array_reduce           // Итеративно уменьшает массив к единственному значению, используя callback-функцию
array_replace_recursive // Рекурсивно заменяет элементы первого массива элементами переданных массивов
array_replace         // Замена элементов массива элементами других переданных массивов
array_reverse         // Возвращает массив с элементами в обратном порядке
array_search          // Осуществляет поиск данного значения в массиве и возвращает
                      // соответствующий ключ в случае удачи
array_shift           // Извлекает первый элемент массива
array_slice           // Выбирает срез массива
array_splice          // Удаляет часть массива и заменяет её чем-нибудь ещё
array_sum             // Вычисляет сумму значений массива
array_uintersect        // Вычисляет расхождение в массивах с дополнительной проверкой индексов,
                      // используя для сравнения значений callback-функцию
array_uintersect_uassoc // Вычисляет расхождение в массивах с дополнительной проверкой индексов,
                      // используя для сравнения значений callback-функцию
array_uintersect       // Вычисляет пересечение массивов с дополнительной проверкой индексов,
                      // используя для сравнения значений callback-функцию
array_uintersect_uassoc // Вычисляет пересечение массивов с дополнительной проверкой индекса,
                      // используя для сравнения индексов и значений callback-функцию
array_unintersect      // Вычисляет пересечение массивов, используя для сравнения значений callback-функцию
array_unique          // Удаляет повторяющиеся значения из массива
array_unshift         // Добавляет один или несколько элементов в начало массива
array_values          // Выбирает все значения массива
array_walk_recursive  // Рекурсивно применяет пользовательскую функцию к каждому элементу массива
array_walk            // Применит заданную пользователем функцию к каждому элементу массива
array                // Создает массив
arsort                // Сортирует массив в обратном порядке, сохраняя ключи
```

```
asort          // Сортирует массив, сохраняя ключи
compact       // Создает массив, содержащий названия переменных и их значения
count         // Подсчитывает количество элементов массива или что-то в объекте
current       // Возвращает текущий элемент массива
each          // Возвращает текущую пару ключ/значение из массива и смещает его указатель
end           // Устанавливает внутренний указатель массива на его последний элемент
extract       // Импортирует переменные из массива в текущую таблицу символов
in_array      // Провернет, присутствует ли в массиве значение
key_exists    // Псевдоним array_key_exists
key           // Выбирает ключ из массива
krsort        // Сортирует массив по ключам в обратном порядке
ksort         // Сортирует массив по ключам
list          // Присваивает переменным из списка значения подобно массиву
natcasesort   // Сортирует массив, используя алгоритм "natural order" без учета регистра символов
natsort       // Сортирует массив, используя алгоритм "natural order"
next          // Передвигает внутренний указатель массива на одну позицию вперед
pos           // Псевдоним current
prev          // Передвигает внутренний указатель массива на одну позицию назад
range         // Создает массив, содержащий диапазон элементов
reset        // Устанавливает внутренний указатель массива на его первый элемент
rsort         // Сортирует массив в обратном порядке
shuffle       // Перемешивает массив
sizeof        // Псевдоним count
sort          // Сортирует массив
uasort        // Сортирует массив, используя пользовательскую функцию для сравнения
              // элементов с сохранением ключей
uksort        // Сортирует массив по ключам, используя пользовательскую функцию для сравнения ключей
usort         // Сортирует массив по значениям используя пользовательскую функцию для сравнения элементов
```