

Массивы

Инициализация:

```
$addresses = array("spam@cyberpromo.net", "abuse@example.com", "root@example.com" );  
$price = array(  
'прокладка'=> 15.29,  
'диск'=> 75.25,  
'шина' => 50.00  
);
```

```
$addresses[0] = "spam@cyberpromo.net";  
$price['прокладка'] = 15.29;
```

Добавление в конец:

```
$family[] = "Павел";
```

`range(1, 50)`; создает массив последовательных чисел или символов между двумя заданными значениями.

`count($family)`; кол-во элементов

Многомерные массивы

```
$row0 = array( 1, 2, 3 );  
$row1 = array(4, 5, 6);  
$row2 = array(7, 8, 9);  
$multi = array($row0, $row1, $row2);
```

Копирование нескольких значений и вырезка:

```
list ($var1, ... ) = $array;
```

Оставшиеся значения игнорируются. Если в массиве меньше - NULL.

```
$subset = array_slice(array, offset,length);
```

Если массив ассоциативный, то ключи заменяются на цифры.

```
$nums = range(1, 7);  
$rows = array_chunk($nums, 3);
```

```
rows[0] = (1,2,3); rows[1] = (4,5,6); rows[3] = (7);
```

Обработка всего массива

`array_sum(array)` Возвращает сумму значений

`array_merge(arr1, arr2, ...)` Объединение массивов

`array_diff(arr1, arr2)` Разница

`array_filter(arr, callback)` Каждое значение элемента передается в `callback`, в возвращаемом - только элементы, для которых `callback` вернул `true`. Ключи массива сохраняются.

Использование в виде множеств

```
function arrayUnion($a, $b){  
    $union = array_merge($a, $b);  
    $union = array_unique($union);  
    return $union;  
}
```

Стек

При помощи `array_push`, `array_pop`

Разное

`array_keys(array)`; - массив ключей

`array_key_exists(key, array)` - проверка существования ключа в массиве

`in_array(to_find, array [, strict])` - поиск значений

`$removed = array_splice(array, start [, length [, replacement]])`; - удаление/вставка элементов в массив

`extract()` - создание переменных из массива (ключи - имена)

`compact()` - из переменных массив

`array_walk(array, callable)`; - применение к каждому элементу функции, третий параметр - параметр функции (для нескольких - массив)

`array_search(элемент, массив)` - возвращает индекс найденного элемента

`in_array()` - есть или нет в массиве элемент

`array_reverse()` - обратный порядок массива

`array_flip()` - массив, в котором ключи стали значениями и наоборот

	По возрастанию	По убыванию	Пользовательская
Сортирует по значениям, затем обновляет индексы	<code>sort()</code>	<code>rsort()</code>	<code>usort()</code>
По значениям	<code>asort()</code>	<code>arsort()</code>	<code>ausort()</code>
По ключам	<code>ksort()</code>	<code>krsort()</code>	<code>kusort()</code>

Пользовательская сортировка требует предоставления функции, которая принимает два значения и возвращает значение, которое определяет порядок двух переданных значений в отсортированном массиве. Функция должна вернуть 1, если первое значение больше, чем второе, -1, если первое значение меньше, чем второе и 0, если значения равны с точки зрения вашей функции сортировки.

Revision #2

Created 4 July 2024 06:27:23 by Admin

Updated 5 July 2024 08:25:13 by Admin