Сделайте таблицу основных функций для работы с массивами:

|  |  |  |
| --- | --- | --- |
| Функция | Назначение | Пример |
| in\_array | Проверка наличия элемента в массиве | <?php $os = array("Mac", "NT", "Irix", "Linux"); if (in\_array("Irix", $os)) {     echo "Нашел Irix"; } if (in\_array("mac", $os)) {     echo "Нашел mac"; } ?> |
| array\_unshift    array\_push | добавление элемента в массив (в начало, в конец) | <?php $queue = array("orange", "banana"); array\_unshift($queue, "apple", "raspberry"); print\_r($queue); ?>  <?php $stack = array("orange", "banana"); array\_push($stack, "apple", "raspberry"); print\_r($stack); ?> |
| array\_pop  array\_shift | удаление элемента (из начала, с конца) | <?php // добавить два элемента в $input array\_push($input, $x, $y); array\_splice($input, count($input), 0, array($x, $y));  // удалить последний элемент из $input array\_pop($input); array\_splice($input, -1);  // удалить первый элемент из $input array\_shift($input); array\_splice($input, 0, 1); // добавить элемент в начало $input array\_unshift($input, $x, $y); array\_splice($input, 0, 0, array($x, $y)); // заменить в $input элемент с индексом $x на значение $y $input[$x] = $y; // для массивов, где ключ равен смещению array\_splice($input, $x, 1, $y);  ?> |
| array\_unique | получение уникальных значений | <?php $input = array("a" => "green", "red", "b" => "green", "blue", "red"); $result = array\_unique($input); print\_r($result); ?> |
| array\_keys | получение ключей ассоциативного массива | <?php $array = array(0 => 100, "color" => "red"); print\_r(array\_keys($array));  $array = array("blue", "red", "green", "blue", "blue"); print\_r(array\_keys($array, "blue"));  $array = array("color" => array("blue", "red", "green"),                "size"  => array("small", "medium", "large")); print\_r(array\_keys($array)); ?> |
| Shuffle  array\_reverse | перемешивание элементов и переворачивание массива | <?php $numbers = range(1, 20); shuffle($numbers); foreach ($numbers as $number) {     echo "$number "; } ?>  <?php $input  = array("php", 4.0, array("green", "red")); $reversed = array\_reverse($input); $preserved = array\_reverse($input, true);  print\_r($input); print\_r($reversed); print\_r($preserved); ?> |
| Sort  rsort  asort  ksort  Arsort  krsort  usort | сортировка (sort, rsort, asort, ksort, arsort, krsort, usort) | <?php $fruits = array("lemon", "orange", "banana", "apple"); sort($fruits); foreach ($fruits as $key => $val) {     echo "fruits[" . $key . "] = " . $val . "\n"; } ?>  <?php $fruits = array("lemon", "orange", "banana", "apple"); rsort($fruits); foreach ($fruits as $key => $val) {     echo "$key = $val\n"; } ?>  <?php $fruits = array("d" => "lemon", "a" => "orange", "b" => "banana", "c" => "apple"); arsort($fruits); foreach ($fruits as $key => $val) {     echo "$key = $val\n"; } ?>  <?php $fruits = array("d"=>"lemon", "a"=>"orange", "b"=>"banana", "c"=>"apple"); ksort($fruits); foreach ($fruits as $key => $val) {     echo "$key = $val\n"; } ?>  <?php $fruits = array("d" => "lemon", "a" => "orange", "b" => "banana", "c" => "apple"); arsort($fruits); foreach ($fruits as $key => $val) {     echo "$key = $val\n"; } ?>  <?php $fruits = array("d"=>"lemon", "a"=>"orange", "b"=>"banana", "c"=>"apple"); krsort($fruits); foreach ($fruits as $key => $val) {     echo "$key = $val\n"; } ?>  <?php function cmp($a, $b) {     if ($a == $b) {         return 0;     }     return ($a < $b) ? -1 : 1; }  $a = array(3, 2, 5, 6, 1);  usort($a, "cmp");  foreach ($a as $key => $value) {     echo "$key: $value\n"; } ?> |
| array\_splice | array\_splice | <?php $input = array("red", "green", "blue", "yellow"); array\_splice($input, 2); var\_dump($input);  $input = array("red", "green", "blue", "yellow"); array\_splice($input, 1, -1); var\_dump($input);  $input = array("red", "green", "blue", "yellow"); array\_splice($input, 1, count($input), "orange"); var\_dump($input);  $input = array("red", "green", "blue", "yellow"); array\_splice($input, -1, 1, array("black", "maroon")); var\_dump($input); ?> |
| array\_slice | array\_slice | <?php $input = array("a", "b", "c", "d", "e");  $output = array\_slice($input, 2);      // возвращает "c", "d" и "e" $output = array\_slice($input, -2, 1);  // возвращает "d" $output = array\_slice($input, 0, 3);   // возвращает "a", "b" и "c"  // обратите внимание на различия в индексах массивов print\_r(array\_slice($input, 2, -1)); print\_r(array\_slice($input, 2, -1, true)); ?> |
| split | разделение строки на элементы массива | <?php list($user, $pass, $uid, $gid, $extra) =     split(":", $passwd\_line, 5); ?> |
| implode | вывод элементов массива в строку | <?php  $array = array('имя', 'почта', 'телефон'); $comma\_separated = implode(",", $array);  echo $comma\_separated; // имя,почта,телефон  // Пустая строка при использовании пустого массива: var\_dump(implode('hello', array())); // string(0) ""  ?> |
|  | объединение элементов массивов, их пересечение |  |