PHP: работа с массивами

Краткий обзор

Массивы в PHP

- Упорядоченное отображение, которое устанавливает соответствие между значением и ключом
- Значением массива может быть другой массив, что позволяет создавать сложные структуры
- Ключами массива могут быть либо целые числа, либо строки
- В одном массиве могут быть ключи разных типов
- Преобразование типов:
  - Строки, содержащие целое число преобразуются к типу integer
  - Числа типа float преобразуются к типу integer отбрасыванием дробной части
  - Тип bool преобразуется к целому
  - Тип null преобразуется к пустой строке “”
  - Массивы, объекты и ресурсы не могут использоваться в качестве ключей

Задание массива

- С помощью языковой конструкции array()
  $array_var = array(1=> 1, 2=> $b, “3” => “3)
- С помощью прямого присвоения значения:
  $array_var[“key”]=5.7
- Индексирование массива без ключа:
  $array_var = array(5, 6, “cat”, ‘rat’);
  $array_var[]=“first string”; $array_var[]=“next string”;
- Указание части ключей:
  $array_var = array(5, 6, 8=>“cat”, ‘rat’);
  $array_var[8]=“first string”; $array_var[]=“next string”;

Подсчет элементов в массиве: count()

- *int* **count** (*array* $arr [, *int* $mode = COUNT_NORMAL]) – подсчитывает число элементов в массиве
- Список параметров:
  - arr – массив в котором подсчитываются элементы
  - mode - если необязательный параметр mode установлен в COUNT_RECURSIVE (или 1), функция будет рекурсивно подсчитывать количество элементов массива с учетом вложенных
- Возвращает количество элементов в arr; если arr не является массивом (или объектом), будет возвращена 1, но если arr - NULL, то будет возвращён 0.

Выбор ключей массива **array_keys()**

- *array* **array_keys** (*array* $input [, *mixed* $search_value = NULL [, *bool* $strict = false]]) - возвращает числовые и строковые ключи, содержащиеся в массиве input.
- Если указан необязательный параметр search_value, функция возвращает только ключи, совпадающие с этим параметром. В обратном случае, функция возвращает все ключи массива input.
- Список параметров ¶
  - input - массив, содержащий возвращаемые ключи;
  - search_value - если указано, будут возвращены только ключи, содержащие данное значение;
  - strict - определяет использование строгой проверки на равенство (===) при поиске
- Возвращает массив с ключами из массива input, индексированный числовыми индексами

Выбор значений массива **array_values()**

- *array* **array_values** (*array* $input) - возвращает массив со всеми элементами массива input, индексированный числовыми индексами
- Параметр - input - массив, содержащий возвращаемые значения;
- Возвращает массив со всеми значениями элементов массива input, индексированный числовыми индексами.

Проверка наличия искомого значения in_array()

- *bool* **in_array** (*mixed* $val , *array* $target [, *bool* $strict = FALSE]) - ищет в target значение val; если strict не установлен, то при поиске будет использовано нестрогое сравнение
- Список параметров ¶
  - val -искомое значение
  - target – массив поиска
  - strict – флаг, требующий проверки соответствия типов параметра val и соответствующего значения массива target
- Если val - строка, сравнение производится с учетом регистра
- Возвращает TRUE, если val был найден в массиве, и FALSE в обратном случае

Поиск значения в массиве: **array_search()**

- *mixed* **array_search** (*mixed* $val , *array* $target [, *bool* $strict = false]) - ищет в target значение val
  - val - искомое значение
  - target – массив поиска
  - strict – флаг, требующий проверки соответствия типов параметра val и соответствующего значения массива target
- Возвращает ключ для val, если он был найден в массиве, иначе FALSE.
- Если val присутствует в target более одного раза, будет возвращён первый найденный ключ. Для того, чтобы возвратить ключи для всех найденных значений, используйте функцию array_keys() с необязательным параметром search_value.

Сортировка массива по значениям **sort()**

- *bool* **sort** (*array* &$target [, *int* $sort_flags = SORT_REGULAR])
  - сортирует массив, после завершения работы функции элементы массива будут расположены в порядке возрастания.
    - target – обрабатываемый массив
    - sort_flags – определяет изменения поведения сортировки, используя флаги сортировки, например,
      - SORT_REGULAR - обычное сравнение элементов (без изменения типов),
      - SORT_NUMERIC - числовое сравнение элементов,
      - SORT_STRING - строковое сравнение элементов и др.
- Возвращает TRUE в случае успешного завершения или FALSE в случае возникновения ошибки.
- Старые индексы заменяются численными, соответствующими месту элемента в отсортированном массиве

Варианты сортировки с **asort(), rsort(), arsort()**

- **asort()** – сохраняет индексы массива сортировки
- **rsort()** – сортировка в обратном порядке
- **arsort()** – сортировка в обратном порядке с сохранением индексов
- Действуют флаги сортировки

Сортировка массива по ключам
ksort(), krsort()

- bool ksort (array &$target [, int $sort_flags = SORT_REGULAR])
  - сортирует массив по ключам, сохраняя отношения между ключами и значениями
    - target – обрабатываемый массив
    - sort_flags – флаги сортировки
- Возвращает TRUE в случае успешного завершения или FALSE в случае возникновения ошибки
- krsort() сортирует в порядке убывания

Работа с указателями: **reset(), end(), next(), prev(), current()**

- mixed reset (array &$array) - перемещает внутренний указатель array к его первому элементу и возвращает значение первого элемента массива или FALSE если массив пуст.
- mixed end (array &$array) - устанавливает внутренний указатель array на последний элемент и возвращает его значение или FALSE если массив пуст.
- mixed next (array &$array) - передвигает его внутренний указатель на одну позицию вперёд, после чего возвращает значение текущего элемента массива или FALSE, если достигнут конец массива.
- mixed prev (array &$array) - передвигает внутренний указатель массива на одну позицию назад, после чего возвращает значение текущего элемента массива или FALSE, если больше элементов нет
- mixed current (array &$array) – возвращает значение текущего элемента массива, не перемещая внутренний указатель

Работа с указателями: **each(), key()**

- array each (array &$array) - возвращает текущую пару ключ/значение из массива и смещает его указатель; после выполнения each(), указатель массива перемещается к следующему его элементу, пока не будет достигнут конец массива.
- Возвращает массив из четырех элементов, с ключами 0, 1, key и value. Элементы 0 и key содержат имя ключа элемента массива, а 1 и value содержат его данные
- mixed key (array &$array) - возвращает индекс текущего элемента массива
- возвращает ключ того элемента массива, на который в данный момент указывает внутренний указатель массива, не сдвигая указателя или NULL, если внутренний указатель вне границ массива или массив пуст.

Хранение массивов: **serialize () и unserialize()**

- string serialize (mixed $value) - генерирует пригодное для хранения представление переменной
- value - значение, которое необходимо сериализовать. serialize() обрабатывает все типы, кроме resource.
- Возвращает строку, содержащую потоковое представление переменной value, которая может быть сохранена где угодно.
- mixed unserialize (string $str) - принимает одну сериализованную переменную и конвертирует ее обратно в переменную PHP.
- str – строка, полученная ранее с помощью функции serialize()
- Возвращает преобразованное значение, которое принимает один из типов boolean, integer, float, string, array или object.
- Примечание: это полезно для хранения или передачи данных между PHP скриптами без потери их типа и структуры.