string **gettype** ( [mixed](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) $var ) — Возвращает тип переменной

bool **settype** ( [mixed](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) &$var , string $type ) Присваивает переменной новый тип

bool **isset** ( [mixed](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) $var [, [mixed](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) $... ] ) — Определяет, была ли установлена переменная значением отличным от **NULL**

bool **empty** ( [mixed](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) $var ) — Проверяет, пуста ли переменная

array **get\_defined\_functions** ( void ) — Возвращает массив всех определённых функций

array **get\_defined\_vars** ( void ) — Возвращает массив всех определенных переменных

array **get\_defined\_constants** ([ bool $categorize = false ] ) — Возвращает ассоциативный массив с именами и значениями всех предопределенных констант

[mixed](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) **print\_r** ( [mixed](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) $expression [, bool $return = false ] ) — Выводит удобочитаемую информацию о переменной echo '<pre>';

void **var\_dump** ( [mixed](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) $expression [, [mixed](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) $... ] ) — Выводит информацию о переменной

bool **define** ( string $name , [mixed](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) $value [, bool $case\_insensitive = false ] ) — Определяет именованную константу define("CONSTANT", "Hello world."); **const** TWO\_HUNDRED = 200; // PHP 5.3

управление настройками php.ini : в файле htaccess -> “php\_flag asp\_tags on”

string **ini\_set** ( string $varname , string $newvalue ) — Установка значения настройки конфигурации

ini\_set('display\_errors', '1');

**error\_reporting**(E\_ALL); // Включаем вывод всех ошибок

**error\_reporting**(0); // Отключаем вывод всех ошибок

**error\_reporting**(E\_ERROR | E\_WARNING); // Включаем определённые уровни ошибок

**error\_reporting**(E\_ALL & ~E\_DEPRECATED);

**iconv**("windows-1251", "UTF-8", $name) - **Преобразование кодировок**

echo <<<HEREDOC

подставятся в строку, а сама строка развернётся

Значения переменных $name и $age

HEREDOC;

**ini\_get**('post\_max\_size'); - получить данные из php.ini

**Буферизация**:

bool **ob\_start** ([ [callable](https://php.net/manual/ru/language.types.callable.php) $output\_callback = NULL [, int $chunk\_size = 0 [, int $flags =PHP\_OUTPUT\_HANDLER\_STDFLAGS ]]] ) — Включение буферизации вывода

string **ob\_get\_contents** ( void ) — Возвращает содержимое буфера вывода

void **ob\_flush** ( void )- Эта функция отправит содержимое буфера вывода (если имеется). Если необходима дальнейшая обработка буфера вывода, то следует вызвать [ob\_get\_contents()](https://php.net/manual/ru/function.ob-get-contents.php) перед **ob\_flush()**, так как содержимое буфера будет удалено после вызова **ob\_flush()**.

bool **ob\_end\_flush** ( void )- Эта функция отправит содержимое буфера вывода (если оно имеется) и отключит эту буферизацию вывода. Если вы хотите использовать содержимое буфера, то вам необходимо вызвать [ob\_get\_contents()](https://php.net/manual/ru/function.ob-get-contents.php)перед **ob\_end\_flush()**, т.к. все содержимое буфера удаляется при вызове **ob\_end\_flush()**.

void **ob\_clean** ( void ) — Очищает (стирает) буфер вывода

bool **ob\_end\_clean** ( void )- Эта функция удаляет содержимое самого верхнего буфера вывода и отключает эту буферизацию. Если вы хотите использовать содержимое буфера, то вам необходимо вызвать [ob\_get\_contents()](https://php.net/manual/ru/function.ob-get-contents.php) перед**ob\_end\_clean()**, так как все содержимое буфера удаляется при вызове **ob\_end\_clean()**.

**Аутентификация:**

1. **На формай**
2. **Базовая «HTTP – аутентификация» на уровне сервера (5 строк в .htaccess), но можно сделать и в PHP**

<?php  
if (!isset($\_SERVER['PHP\_AUTH\_USER'])) {  
 header('WWW-Authenticate: Basic realm="My Realm"');  
 header('HTTP/1.0 401 Unauthorized');  
 echo 'Текст, отправляемый в том случае,  
 если пользователь нажал кнопку Cancel';  
 exit;  
} else {  
 echo "<p>Hello {$\_SERVER['PHP\_AUTH\_USER']}.</p>";  
 echo "<p>Вы ввели пароль {$\_SERVER['PHP\_AUTH\_PW']}.</p>";  
}  
?>

**Математические функции:**

[number](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.number) **abs** ( [mixed](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) $number ) — Модуль числа

[mixed](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) **max** ( array $values )

( [mixed](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) $value1 , [mixed](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) $value2 [, [mixed](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) $... ] ) — Возвращает наибольшее значение

[mixed](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) **min** ( array $values )

( [mixed](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) $value1 , [mixed](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) $value2 [, [mixed](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) $... ] ) — Находит наименьшее значение

int **mt\_rand** ( void )

( int $min , int $max ) — Генерирует случайное число

float **ceil** ( float $value ) — Округляет дробь в большую сторону

float **floor** ( float $value ) — Округляет дробь в меньшую сторону

bool **is\_numeric** ( [mixed](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) $var ) — Проверяет, является ли переменная числом или строкой, содержащей число

int **intval** ( [mixed](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) $var [, int $base = 10 ] ) — Возвращает целое значение переменной

string **money\_format** ( string $format , float $number )- форматирует число **number** как денежную величину

string **bin2hex** ( string $str )- — Преобразует бинарные данные в шестнадцатеричное представление

**Работа с сокетами:**

resource **fsockopen** ( string $hostname [, int $port = -1 [, int &$errno [, string &$errstr [,float $timeout = ini\_get("default\_socket\_timeout") ]]]] ) — Открывает соединение с интернет сокетом или доменным сокетом Unix $sock = fsockopen("localhost",80);

/\* $sock = **fsockopen**("from.php.local",80);

**fputs**($sock, "GET /index.html HTTP/1.0\n\n");

while (!**feof**($sock)){

echo **fgets**($sock);

echo "<br>";}

**fclose**($sock) ;

\*/

**Обработка ошибок**

// Функция перехвата ошибок

function MyErro($no, $msg, $file, $line){

$dt = date('d-m-Y H:i:s');

$str = "[$dt] - $msg in $file:$line\n";

// Логгируем пользовательские ошибки

switch ($no) {

case E\_USER\_ERROR:

case E\_USER\_WARNING:

case E\_USER\_NOTICE:

echo $msg;

}

error\_log("$str", 3, "error.log");

}

// Установка перехватчика ошибок

set\_error\_handler("MyErro");

// Отлавливаем ошибки

if(!navMenu($leftMenu1))

trigger\_error(ERR\_NAVIG, E\_USER\_ERROR);

trigger\_error("Что-то случилось", E\_USER\_ERROR);

**Работа с** [JSON](https://php.net/manual/ru/ref.json.php)**:**

[mixed](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) **json\_decode** ( string $json [, bool $assoc = false [, int $depth = 512 [, int $options = 0]]] **) — Декодирует JSON строку**

string **json\_encode** ( [mixed](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) $value [, int $options = 0 [, int $depth = 512 ]] **) — Возвращает JSON-представление данных**

string **json\_last\_error\_msg** ( void ) **— Returns the error string of the last json\_encode() or json\_decode() call**

int **json\_last\_error** ( void **) — Возвращает последнюю ошибку**

**Работа со строками:**

array **explode** ( string $delimiter , string $string [, int $limit ] ) — Возвращает массив строк, полученных разбиением строки **string** с использованием **delimiter** в качестве разделителя

string **implode** ( string $glue , array $pieces ) — Объединяет элементы массива с помощью строки **glue**.

int **substr\_count** ( string $haystack , string $needle [, int $offset = 0 [, int $length ]] ) — возвращает число вхождений подстроки **needle** в строку **haystack**. Параметр **needle** чувствителен к регистру.

int **printf** ( string $format [, [mixed](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) $args [, [mixed](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) $... ]] ) — Выводит строку, отформатированную в соответствии с аргументом **format**.

string **sprintf** ( string $format [, [mixed](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) $args [, [mixed](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) $... ]] ) — Возвращает строку, созданную с использованием строки формата **format**

|  |  |
| --- | --- |
|  *%* - символ процента. Аргумент не используется.   *b* - целое выводится в виде двоичного числа.   *c* - целое выводится в виде символа с соответствующим кодом ASCII.   *d* - целое и выводится в виде десятичного числа со знаком.   *e* - число в в научной нотации (например, 1.2e+2). Описатель точности указывает на количество знаков после запятой   *E* - аналогично *%e*, но использует заглавную букву (например, 1.2E+2).   *f* - число с плавающей точкой и также выводится в зависимости от локали.   *F* - аргумент трактуется как число с плавающей точкой и также выводится, но без зависимости от локали   *g* - выбирает самую краткую запись из *%e* и *%f*.   *G* - выбирает самую краткую запись из *%E* и *%f*.   *o* - целое выводится в виде восьмеричного числа.   *s* - строка.   *u* - целое выводится в виде десятичного числа без знака.   *x* - целое выводится в виде шестнадцатеричного числа (в нижнем регистре).   *X* - целое выводится в виде шестнадцатеричного числа (в верхнем регистре). | <?php $num = 5; $location = 'дереве'; $format = '%d обезьян сидят на %s'; echo sprintf($format, $num, $location); ?>  Результат: "5 обезьян сидят на дереве"  <?php $format = 'На %s сидят %d обезьян'; echo sprintf($format, $num, $location); ?>  <?php $format = 'На %2$s сидят %1$d обезьян'; echo sprintf($format, $num, $location); ?>  <?php $format = 'На %2$s сидят %1$d обезьян.  Как здорово, когда на %2$s сидят %1$d обезьян.'; echo sprintf($format, $num, $location); ?> |

string **substr** ( string $string , int $start [, int $length ] ) — Возвращает подстроку строки string, начинающейся с start символа по счету и длиной lengthсимволов

string **number\_format** ( float $number [, int $decimals = 0 ] ) — Форматирует число с разделением групп

[mixed](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) **strpos** ( string $haystack , [mixed](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) $needle [, int $offset = 0 ] ) — Возвращает позицию первого вхождения подстроки $needle в $haystack

[mixed](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) **stripos** ( string $haystack , string $needle [, int $offset = 0 ] ) — Возвращает позицию первого вхождения подстроки без учета регистра

int **strrpos** ( string $haystack , string $needle [, int $offset = 0 ] )- Возвращает позицию последнего вхождения подстроки $needle в $haystack

int **strripos** ( string $haystack , string $needle [, int $offset = 0 ] )- Возвращает позицию последнего вхождения без учета регистра

int **mb\_strlen** ( string $string ) - Возвращает длину строки **string**.

string **strstr** ( string $haystack , [mixed](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) $needle [, bool $before\_needle = false ] )- Возвращает подстроку строки **haystack** начиная с первого вхождения **needle** (и включая его) и до конца строки **haystack**. Если **before\_needle** установлен в **TRUE**, возвращает часть строки **haystack** до первого вхождения **needle**(исключая needle).

string **stristr -** ||-||-|| без учета регистра

**strchr** — Псевдоним [strstr()](https://php.net/manual/ru/function.strstr.php)

int **strcmp** ( string $str1 , string $str2 )- Возвращает отрицательное число, если **str1** меньше, чем **str2**, положительное число,если **str1** больше, чем **str2**, и 0 если строки равны.

int **strncmp** ( string $str1 , string $str2 , int $len ) )- Возвращает отрицательное число, если **str1** меньше, чем **str2**, положительное число, если **str1** больше, чем **str2**, и 0 если строки равны. **Len** Количество символов, используемых в сравнении. учитывает регистр

int **strcasecmp** ( string $str1 , string $str2 ) - Возвращает отрицательное число, если **str1** меньше, чем **str2**, положительное число, если **str1** больше, чем **str2**, и 0 если строки равны. без учета регистра.

int **strncasecmp** ( string $str1 , string $str2 , int $len )- Возвращает отрицательное число, если **str1** меньше, чем **str2**, положительное число, если **str1** больше, чем **str2**, и 0 если строки равны. без учета регистра

int **strnatcmp** ( string $str1 , string $str2 )- возвращает отрицательное число, если **str1**меньше, чем **str2**, положительное число, если **str1** больше, чем **str2**, и 0 если строки равны. сравнение алфавитно-цифровое

int **strnatcasecmp** ( string $str1 , string $str2 )- возвращает отрицательное число, если **str1**меньше, чем **str2**, положительное число, если **str1** больше, чем **str2**, и 0 если строки равны. сравнение алфавитно-цифровое без учета регистра

string **str\_repeat** ( string $input , int $multiplier )- Возвращает строку **input**, повторенную **multiplier** раз.

[mixed](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) **str\_replace** ( [mixed](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) $search , [mixed](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) $replace , [mixed](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) $subject [, int &$count ] )- Эта функция возвращает строку или массив, в котором все вхождения **search** в **subject** заменены на**replace**.

int **strspn** ( string $subject , string $mask [, int $start [, int $length ]] )- Возвращает длину участка от начала строки **subject**, содержащий только символы из **mask**.

int **strcspn** ( string $str1 , string $str2 [, int $start [, int $length ]] )- Возвращает длину участка в начале строки **str1**, который не содержит ни одного символа из строки **str2**.

**Преобразование строк:**

string **strip\_tags** ( string $str [, string $allowable\_tags ] ) — Удаляет HTML, NULL-байты и PHP-теги из строки strip\_tags($text, '<p><a>');

string **strtolower** ( string $string ) — Преобразует строку в нижний регистр

string **strtoupper** ( string $string ) — Преобразует строку в верхний регистр

string **lcfirst** ( string $str ) — Преобразует первый символ строки в нижний регистр

string **ucfirst** ( string $str ) — Преобразует первый символ строки в верхний регистр

string **trim** ( string $str [, string $character\_mask = " \t\n\r\0\x0B" ] ) — Удаляет пробелы из начала и конца строки, rtrim, ltrim

**chop** — Псевдоним [rtrim()](https://php.net/manual/ru/function.rtrim.php)

string **nl2br** ( string $string [, bool $is\_xhtml = true ] )- Возвращает строку **string**, в которой перед каждым переводом строки (\r\n, \n\r, \n и \r) вставлен '<br />' или '<br>'.

string **htmlspecialchars** ( string $string [, int $flags = ENT\_COMPAT | ENT\_HTML401 [, string$encoding = ini\_get("default\_charset") [, bool $double\_encode = true ]]] )- Преобразует специальные символы в HTML-сущности

string **htmlspecialchars\_decode** ( string $string [, int $flags = ENT\_COMPAT | ENT\_HTML401 ] )- Преобразует специальные HTML-сущности обратно в соответствующие символы

string **stripslashes** ( string $str )- — Удаляет экранирование символов

string **strrev** ( string $string )- Возвращает строку **string**, перевернутую задом наперед

string **str\_shuffle** ( string $str ) — Переставляет символы в строке случайным образом

string **str\_rot13** ( string $str )- Выполняет над строкой **str** преобразование ROT13 и возвращает полученную строку.

string **urlencode** ( string $str ) — URL-кодирование строки

string **urldecode** ( string $str ) — Декодирование URL-кодированной строки

string **ucwords** ( string $str ) — Преобразует в верхний регистр первый символ каждого слова в строке

string **wordwrap** ( string $str [, int $width = 75 [, string $break = "\n" [, bool $cut = false ]]] ) — Переносит строку по указанному количеству символов

string **convert\_cyr\_string** ( string $str , string $from , string $to ) — Преобразует строку из одной кириллической кодировки в другую

string **convert\_uudecode** ( string $data ) — Декодирует строку из формата uuencode в обычный вид

string **chr** ( int $ascii )- Возвращает строку из одного символа, код которого задан аргументом **ascii**.

int **ord** ( string $string )- Возвращает ASCII-код первого символа строки **string**.

string **password\_hash** ( string $password , integer $algo [, array $options ] ) — Creates a password hash (5.5 php)

boolean **password\_verify** ( string $password , string $hash ) — Verifies that a password matches a hash (5.5 php)

string **md5** ( string $str [, bool $raw\_output = false ] Возвращает MD5-хэш строки

string **md5\_file** ( string $filename [, bool $raw\_output = false ] )- Возвращает MD5-хэш файла

string **sha1** ( string $str [, bool $raw\_output = false ] ) — Возвращает SHA1-хэш строки

string **sha1\_file** ( string $filename [, bool $raw\_output = false ] ) — Возвращает SHA1-хэш файла

int **crc32** ( string $str ) — Вычисляет полином CRC32 для строки

string **crypt** ( string $str [, string $salt ] ) — Необратимое хэширование строки

[mixed](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) **count\_chars** ( string $string [, int $mode = 0 ] )- Подсчитывает количество вхождений каждого из символов с ASII-кодами в диапазоне (0..255) в строке**string** и возвращает эту информацию в различных форматах.

[mixed](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) **str\_word\_count** ( string $string [, int $format = 0 [, string $charlist ]] )- Подсчитывает количество слов, входящих в строку **string**. Если необязательный аргумент **format** не передан, возвращается целое число, равное количеству слов. В случае, если указан аргумент **format**, возвращается массив, содержимое которого зависит от значения **format**. Ниже описаны допустимые значения аргумента **format** и соответствующие им возвращаемые значения.

int **substr\_count** ( string $haystack , string $needle [, int $offset = 0 [, int $length ]] )- возвращает число вхождений подстроки **needle** в строку **haystack**. Параметр **needle** чувствителен к регистру.

String **serialize** ( [mixed](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) $value ) — Генерирует пригодное для хранения представление переменной (массив в строку)

[mixed](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) **unserialize** ( string $str ) — Создает PHP-значение из хранимого представления (из строки в массив)

**Работа с массивами:**

int **count** ( [mixed](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) $array\_or\_countable [, int $mode = COUNT\_NORMAL ] ) — Подсчитывает количество элементов массива или что-то в объекте

**sizeof** — Псевдоним [count()](https://php.net/manual/ru/function.count.php)

bool **array\_key\_exists** ( [mixed](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) $key , array $array ) — Функция возвращает **TRUE**, если в массиве присутствует указанный ключ **key**

bool **in\_array** ( [mixed](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) $needle , array $haystack [, bool $strict = FALSE ] ) — Ищет в **haystack** значение **needle**. Если **strict** не установлен, то при поиске будет использовано нестрогое сравнение.

array **array\_keys** ( array $array [, [mixed](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) $search\_value [, bool $strict = false ]] ) — возвращает числовые и строковые ключи, содержащиеся в массиве **array**.

array **array\_values** ( array $array ) — возвращает массив со всеми элементами массива **array**

array **array\_merge** ( array $array1 [, array $... ] ) — Сливает элементы одного или большего количества массивов таким образом, что значения одного массива присоединяются к концу предыдущего

array **array\_merge\_recursive** ( array $array1 [, array $... ] ) — Рекурсивное слияние двух или более многомерных массивов

array **array\_diff** ( array $array1 , array $array2 [, array $... ] ) — Сравнивает **array1** с одним или несколькими другими массивами и возвращает значения из **array1**, которые отсутствуют во всех других массивах.

array **array\_diff\_assoc** ( array $array1 , array $array2 [, array $... ] ) Сравнивает **array1** с **array2** и возвращает разницу. В отличие от [array\_diff()](https://php.net/manual/ru/function.array-diff.php), ключи также участвуют в сравнении.

array **array\_diff\_key** ( array $array1 , array $array2 [, array $... ] ) Сравнивает ключи **array1** с ключами **array2** и возвращает разницу. Эта функция схожа с [array\_diff()](https://php.net/manual/ru/function.array-diff.php) за исключением того, что сравниваются ключи, а не значения.

array **array\_intersect** ( array $array1 , array $array2 [, array $... ] ) — возвращает массив, содержащий все значения массива **array1**, которые содержатся во всех аргументах. Обратите внимание, что ключи сохраняются.

int **array\_push** ( array &$array , [mixed](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) $value1 [, [mixed](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) $... ] ) — Добавляет один или несколько элементов в конец массива

[mixed](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) **array\_pop** ( array &$array ) — извлекает и возвращает последнее значение параметра **array**, уменьшая размер **array** на один элемент

int **array\_unshift** ( array &$array , [mixed](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) $value1 [, [mixed](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) $... ] ) — Добавляет один или несколько элементов в начало массива

[mixed](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) **array\_shift** ( array &$array ) — Извлекает первый элемент массива, сокращая массив

bool **sort** ( array &$array [, int $sort\_flags = SORT\_REGULAR ] ) — Сортирует массив по возрастанию значений

bool **rsort** ( array &$array [, int $sort\_flags = SORT\_REGULAR ] ) — Сортирует массив в обратном порядке

bool **ksort** ( array &$array [, int $sort\_flags = SORT\_REGULAR ] ) — Сортирует массив по ключам

bool **krsort** ( array &$array [, int $sort\_flags = SORT\_REGULAR ] ) — Сортирует массив по ключам в обратном порядке

bool **asort** ( array &$array [, int $sort\_flags = SORT\_REGULAR ] ) — Сортирует массив, сохраняя ключи

bool **arsort** ( array &$array [, int $sort\_flags = SORT\_REGULAR ] ) — Сортирует массив в обратном порядке, сохраняя ключи

array **array\_reverse** ( array $array [, bool $preserve\_keys = false ] ) — Возвращает массив с элементами в обратном порядке

bool **shuffle** ( array &$array ) — Перемешивает массив

bool **natsort** ( array &$array ) — Сортирует массив, используя алгоритм "natural order"

bool **natcasesort** ( array &$array ) — Сортирует массив, используя алгоритм "natural order" без учета регистра символов

array **compact** ( [mixed](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) $varname1 [, [mixed](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) $... ] ) — упаковывает переменные со значениями в массив  
**compact** (“host”, “user”, “pass”); - имя переменной становится ключом, а содержимое переменной становится значением

int **extract** ( array &$array [, int $flags = EXTR\_OVERWRITE [, string $prefix = NULL ]] ) — Импортирует переменные из массива в текущую таблицу символов  
**EXTR\_OVERWRITE** Если переменная с таким именем существует, она будет перезаписана.

**EXTR\_SKIP** Если переменная с таким именем существует, ее текущее значение не будет перезаписано.

**EXTR\_PREFIX\_SAME** Если переменная с таким именем существует, к её имени будет добавлен префикс, определённый параметром **prefix**.

**EXTR\_PREFIX\_ALL** Добавить префикс **prefix** ко всем именам переменных.

**EXTR\_PREFIX\_INVALID** Добавить префикс **prefix** только к некорректным/числовым именам переменных.

array **array\_slice** ( array $array , int $offset [, int $length = NULL [, bool $preserve\_keys = false ]] ) — Выбирает срез массива $array от $offset длинной $length

array **array\_fill** ( int $start\_index , int $num , [mixed](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) $value ) — Заполняет массив **num** элементами со значением **value**, начиная с ключа **start\_index**.

array **array\_fill\_keys** ( array $keys , [mixed](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) $value ) — Создает и заполняет массив значением параметра **value**, используя значения массива **keys** в качестве ключей.

array **range** ( [mixed](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) $start , [mixed](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) $end [, [number](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.number) $step = 1 ] ) — Создает массив, содержащий диапазон элементов

[number](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.number) **array\_sum** ( array $array ) — Вычисляет сумму значений массива

[number](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.number) **array\_product** ( array $array ) — Вычислить произведение значений массива

[mixed](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) **array\_rand** ( array $array [, int $num = 1 ] ) — Выбирает одно или несколько случайных значений из массива

array **array\_unique** ( array $array [, int $sort\_flags = SORT\_STRING ] ) — Убирает повторяющиеся значения из массива

array **array\_flip** ( array $array ) — Меняет местами ключи с их значениями в массиве

array **array\_count\_values** ( array $array ) Функция возвращает массив, ключами которого являются значения массива **array**, а значениями - частота повторения значений **array**.

array **array\_replace** ( array $array1 , array $array2 [, array $... ] ) **-** замещает значения массива **array1** значениями с такими же ключами из других переданных массивов

[mixed](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) **array\_search** ( [mixed](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) $needle , array $haystack [, bool $strict = false ] ) — Осуществляет поиск данного значения в массиве и возвращает соответствующий ключ в случае удачи

bool **array\_walk** ( array &$array , [callable](https://php.net/manual/ru/language.types.callable.php) $callback [, [mixed](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) $userdata = NULL ] ) - Применяет пользовательскую функцию **callback** к каждому элементу массива **array**.

bool **array\_walk\_recursive** ( array &$array , [callable](https://php.net/manual/ru/language.types.callable.php) $callback [, [mixed](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) $userdata = NULL ] ) — Рекурсивно применяет пользовательскую функцию к каждому элементу массива

array **array\_map** ( [callable](https://php.net/manual/ru/language.types.callable.php) $callback , array $array1 [, array $... ] ) — Применяет callback-функцию ко всем элементам указанных массивов

[mixed](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) **current** ( array &$array ) — Возвращает текущий элемент массива по внутреннему указателю

**pos** — Псевдоним [current()](https://php.net/manual/ru/function.current.php)

[mixed](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) **next** ( array &$array ) — Передвигает внутренний указатель массива на одну позицию вперёд

[mixed](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) **end** ( array &$array ) — Устанавливает внутренний указатель массива на его последний элемент

[mixed](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) **prev** ( array &$array ) — Передвигает внутренний указатель массива на одну позицию назад

[mixed](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) **reset** ( array &$array ) — Устанавливает внутренний указатель массива на его первый элемент

[mixed](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) **key** ( array &$array ) — Выбирает ключ из массива

**Функции даты и времени:**

array **getdate** ([ int $timestamp = time() ] ) — Возвращает информацию о дате/времени

int **time** ( void ) — Возвращает текущую метку времени Unix

int **mktime** ([ int $hour = date("H") [, int $minute = date("i") [, int $second = date("s") [, int$month = date("n") [, int $day = date("j") [, int $year = date("Y") [, int $is\_dst = -1 ]]]]]]] ) — Возвращает метку времени Unix для заданной даты

string **strftime** ( string $format [, int $timestamp = time() ] ) — Форматирует текущую дату/время с учетом текущих настроек локали

|  |  |  |
| --- | --- | --- |
| %a | Сокращенное название дня недели | От Sun до Sat |
| %A | Полное название дня недели | От Sunday доSaturday |
| %d | Двухзначное представление дня месяца (с ведущими нулями) | От 01 до 31 |
| %j | Порядковый номер в году, 3 цифры с ведущими нулями | От 001 до 366 |
| %u | Порядковый номер дня недели согласно стандарту ISO-8601 | От 1 (понедельник) до 7 (воскресенье) |
| %w | Порядковый номер дня недели | От 0 (воскресенье) до 6 (суббота) |
| %b | Аббревиатура названия месяца, в соответствии с настройками локали | От Jan до Dec |
| %B | Полное название месяца, в соответствии с настройками локали | От January доDecember |
| %m | Двухзначный порядковый номер месяца | От 01 (январь) до 12(декабрь) |
| %y | Двухзначный порядковый номер года | Пример: 09 для 2009, 79 для 1979 |
| %Y | Четырехзначный номер года | Пример: 2038 |
| %H | Двухзначный номер часа в 24-часовом формате | От 00 до 23 |
| %k | Двухзначное представление часа в 24-часовом формате, с пробелом | От 0 до 23 |
| %I | Двухзначный номер часа в 12-часовом формате | От 01 до 12 |
| %l | Час в 12-часовом формате, с пробелом перед одиночной цифрой | От 1 до 12 |
| %M | Двухзначный номер минуты | От 00 до 59 |
| %r | Тоже что и "%I:%M:%S %p" | Пример: 09:34:17 PM для 21:34:17 |
| %R | Тоже что и "%H:%M" | Пример: 00:35 для 12:35 AM, |
| %S | Двухзначный номер секунды | От 00 до 59 |
| %T | Тоже что и "%H:%M:%S" | Пример: 21:34:17 для 09:34:17 PM |
| %X | Предпочитаемое отображение времени в зависимости от локали, без даты | Example: 03:59:16 or 15:59:16 |
| %D | Тоже что и "%m/%d/%y" | 02/05/09 для 5 февраля 2009 |
| %F | Тоже что и "%Y-%m-%d" (обычно используется в метках времени баз данных) | 2009-02-05для 5 февраля 2009 |
| setlocale(LC\_ALL,"ru\_RU.UTF-8"); echo strftime("%D"); | | |

string **date** ( string $format [, int $timestamp = time() ] ) — Форматирует вывод системной даты/времени

|  |  |  |
| --- | --- | --- |
| d | День месяца, 2 цифры с ведущим нулём | от 01 до 31 |
| j | День месяца без ведущего нуля | от 1 до 31 |
| D | Текстовое представление дня недели, 3 символа | от Mon до Sun |
| l | Полное наименование дня недели | от Sunday доSaturday |
| N | Порядковый номер дня недели в соответствии со стандартом ISO-8601 | от 1(понедельник) до 7(воскресенье) |
| S | Английский суффикс порядкового числительного дня месяца, 2 символа | st, nd, rd или th |
| F | Полное наименование месяца, например January или March | от January доDecember |
| m | Порядковый номер месяца с ведущим нулём | от 01 до 12 |
| M | Сокращенное наименование месяца, 3 символа | от Jan до Dec |
| n | Порядковый номер месяца без ведущего нуля | от 1 до 12 |
| Y | Порядковый номер года, 4 цифры | Примеры: 1999,2003 |
| y | Номер года, 2 цифры | Примеры: 99, 03 |
| g | Часы в 12-часовом формате без ведущего нуля | от 1 до 12 |
| G | Часы в 24-часовом формате без ведущего нуля | от 0 до 23 |
| h | Часы в 12-часовом формате с ведущим нулём | от 01 до 12 |
| H | Часы в 24-часовом формате с ведущим нулём | от 00 до 23 |
| i | Минуты с ведущим нулём | от 00 до 59 |
| s | Секунды с ведущим нулём | от 00 до 59 |
| // выведет примерно следующее: Monday 8th of August 2005 03:12:46 PM echo date('l jS \of F Y h:i:s A'); | | |

int **strtotime** ( string $time [, int $now = time() ] ) — Преобразует текстовое представление даты на английском языке в метку времени Unix

**Работа с папками и файлами:**

bool **mkdir** ( string $pathname [, int $mode = 0777 [, bool $recursive = false [, resource $context]]] ) — **Создаёт** директорию

bool **rmdir** ( string $dirname [, resource $context ] ) — **Удаляет** директорию должна быть пустая

string **getcwd** ( void ) — **Получает имя** текущего рабочего каталога

resource **opendir** ( string $path [, resource $context ] ) — **Открывает** дескриптор каталога

void **closedir** ([ resource $dir\_handle ] ) — **Освобождает** дескриптор каталога

string **readdir** ([ resource $dir\_handle ] ) — **Получает** элемент каталога по его дескриптору

bool **is\_file** ( string $filename ) — **Определяет**, является ли файл обычным файлом

bool **is\_dir** ( string $filename ) — **Определяет**, является ли имя файла директорией

array **scandir** ( string $directory [, int $sorting\_order = SCANDIR\_SORT\_ASCENDING [, resource$context ]] ) — **Получает список файлов** и каталогов, расположенных по указанному пути

array **glob** ( string $pattern [, int $flags = 0 ] ) — Находит файловые пути, совпадающие с шаблоном glob("\*.txt")

**require\_once** ‘file.php’;

**include\_once** 'vars.php';

array **file** ( string $filename [, int $flags = 0 [, resource $context ]] ) — Читает содержимое файла и помещает его в массив

string **file\_get\_contents** ( string $filename [, bool $use\_include\_path = false [, resource$context [, int $offset = -1 [, int $maxlen ]]]] ) — Читает содержимое файла в строку

int **file\_put\_contents** ( string $filename , [mixed](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) $data [, int $flags = 0 [, resource $context ]] )

bool **file\_exists** ( string $filename ) — Проверяет наличие указанного файла или каталога

array **parse\_ini\_file** ( string $filename [, bool $process\_sections = false [, int $scanner\_mode = INI\_SCANNER\_NORMAL ]] )- загружает ini-файл, указанный в аргументе **filename**, и возвращает его настройки в виде ассоциативного массива.

resource **fopen** ( string $filename , string $mode [, bool $use\_include\_path = false [, resource$context ]] ) — Открывает файл или URL

|  |  |
| --- | --- |
| **'r'** | Открывает файл только для чтения; помещает указатель в начало файла. |
| 'r+' | Открывает файл для чтения и записи; помещает указатель в начало файла. |
| **'w'** | Открывает файл только для записи; помещает указатель в начало файла и обрезает файл до нулевой длины. Если файл не существует - пробует его создать. |
| **'w+'** | Открывает файл для чтения и записи; помещает указатель в начало файла и обрезает файл до нулевой длины. Если файл не существует - пытается его создать. |
| **'a'** | Открывает файл только для записи; помещает указатель в конец файла. Если файл не существует - пытается его создать. |
| **'a+'** | Открывает файл для чтения и записи; помещает указатель в конец файла. Если файл не существует - пытается его создать. |
| **'x'** | Создаёт и открывает только для записи; помещает указатель в начало файла. Если файл уже существует, вызов **fopen()** закончится неудачей, вернёт **FALSE** и выдаст ошибку уровня**E\_WARNING**. Если файл не существует, попытается его создать. Это эквивалентно указанию флагов O\_EXCL|O\_CREAT для внутреннего системного вызова open(2). |
| **'x+'** | Создаёт и открывает для чтения и записи; иначе имеет то же поведение что и'x'. |
| **'c'** | Открывает файл только для записи. Если файл не существует, то он создается. Если же файл существует, то он не обрезается (в отличии от 'w'), и вызов к этой функции не вызывает ошибку (также как и в случае с 'x'). Указатель на файл будет установлен на начало файла. Это может быть полезно при желании заблокировать файл (смотри [flock()](https://php.net/manual/ru/function.flock.php)) перед изменением, так как использование 'w' может обрезать файл еще до того как была получена блокировка (если вы желаете обрезать файл, можно использовать функцию[ftruncate()](https://php.net/manual/ru/function.ftruncate.php) после запроса на блокировку). |
| **'c+'** | Открывает файл для чтения и записи; иначе имеет то же поведение, что и 'c'. |

bool **fclose** ( resource $handle )- Функция закрывает файл, на который указывает дескриптор **handle**.

bool **flock** ( resource $handle , int $operation [, int &$wouldblock ] ) — блокировка файлов  
- **LOCK\_EX** для блокировки (запись).  
- **LOCK\_UN** для снятия блокировки  
- **LOCK\_SH** для получения разделяемой блокировки

int **fileatime** ( string $filename ) — Возвращает время последнего **доступа** к файлу

int **filemtime** ( string $filename ) — Возвращает время последнего **изменения** файла

bool **is\_readable** ( string $filename ) — Определяет существование файла и доступен ли он для чтения

bool **is\_writable** ( string $filename ) — Определяет, доступен ли файл для записи

int **fpassthru** ( resource $handle )- Читает указанный файловый указатель с текущей позиции до EOF (конца файла) и выводит без echo.

int **readfile** ( string $filename [, bool $use\_include\_path = false [, resource $context ]] )- Читает файл и выводит целиком.

string **fgets** ( resource $handle [, int $length ] )- Читает строку из файлового указателя.

string **fgetss** ( resource $handle [, int $length [, string $allowable\_tags ]] ) - Читает строку из файлового указателя осуществляет отбрасывание любых HTML и PHP-тегов из прочитанной строки

string **fgetc** ( resource $handle )- Считывает символ из переданного указателя на файл.

int **fwrite** ( resource $handle , string $string [, int $length ] )- записывает содержимое **string** в файловый поток **handle**.

**fputs** — Псевдоним [fwrite()](https://php.net/manual/ru/function.fwrite.php)

int **fseek** ( resource $handle , int $offset [, int $whence = SEEK\_SET ] )- Устанавливает смещение в файле, на который ссылается **handle**.

int **ftell** ( resource $handle )- Возвращает позицию файлового указателя **handle**.

bool **rewind** ( resource $handle ) — Сбрасывает курсор у файлового указателя

bool **feof** ( resource $handle ) — Проверяет, достигнут ли конец файла

string **fread** ( resource $handle , int $length )- читает до **length** байт из файлового указателя **handle**. Чтение останавливается как только было достигнуто одно из следующих условий:

* было прочитано **length** байт
* достигнут EOF (конец файла)
* стал доступен пакет или произошел [таймаут сокета](https://php.net/manual/ru/function.socket-set-timeout.php) (для сетевых потоков)
* если читаемый поток является буферизованным и не представляет собой обычный файл, то за один раз максимум читается количество байт, равное размеру одной порции данных (обычно это 8192), однако, в зависимости от ранее буферизованных данных размер возвращаемых данных может быть больше размера одной порции данных.

array **file** ( string $filename [, int $flags = 0 [, resource $context ]] ) — Читает содержимое файла и помещает его в массив

int **filesize** ( string $filename )- Возвращает размер указанного файла в байтах или **FALSE** в случае возникновения ошибки.

bool **copy** ( string $source , string $dest [, resource $context ] )- Копирует файл **source** в файл с именем **dest**.

bool **rename** ( string $oldname , string $newname [, resource $context ] )- Пытается переименовать **oldname** в **newname**, перенося файл между директориями, если необходимо. Если**newname** существует, то он будет перезаписан

bool **unlink** ( string $filename [, resource $context ] )- Удаляет файл **filename**.

Директивы php.ini:

* file\_uploads – запрещает или разрешает загрузку на сервер
* upload\_max\_filesize – максимальный размер загружаемого файла (окола 2 Mb)
* post\_max\_size – общая длинна файла из формы
* upload\_tmp\_dir – временная директория ля хранения файлов
* max\_input\_time – время для загрузки
* max\_file\_uploads – сколько файлов можна загружать

**<!-- Тип кодирования данных, enctype, ДОЛЖЕН БЫТЬ указан ИМЕННО так -->**

**<form enctype="multipart/form-data" action="\_\_URL\_\_" method="POST">**

**<!-- Поле MAX\_FILE\_SIZE должно быть указано до поля загрузки файла -->**

**<input type="hidden" name="MAX\_FILE\_SIZE" value="30000" />**

**<!-- Название элемента input определяет имя в массиве $\_FILES -->**

**Отправить этот файл: <input name="userfile" type="file" />**

**<input type="submit" value="Send File" />**

**</form>**

**<form action="upload.php" method="post" enctype="multipart/form-data">**

**<input type="file" name="filename"><br>**

**<input type="submit" value="Загрузить"><br>**

**</form>**

# Суперглобальный массив $\_FILES

Для каждого загруженного файла массив $\_FILES содержит массив, со следующими данными:

* **$\_FILES['userfile']['name']** - исходное имя файла, такое, каким его видел пользователь, выбирая файл;
* **$\_FILES['userfile']['type']** - mime/type файла, к примеру, может быть image/gif; это поле полезно сохранить, если Вы хотите предоставлять интерфейс для скачивания загруженных файлов;
* **$\_FILES['userfile']['size']** - размер загруженного файла;
* **$\_FILES['userfile']['tmp\_name']** - полный путь к временному файлу на диске;
* **$\_FILES['userfile']['error']** - Начиная с версии 4.2.0, содержит код ошибки, который равен 0, если операция прошла успешно.

## php загрузка файла коды ошибок

**UPLOAD\_ERR\_OK**

Значение: 0; Ошибок не возникало, файл был успешно загружен на сервер.

**UPLOAD\_ERR\_INI\_SIZE**

Значение: 1; Размер принятого файла превысил максимально допустимый размер, который задан директивой *upload\_max\_filesize* конфигурационного файла [**php.ini**](http://htmlweb.ru/prepare/?phpini).

**UPLOAD\_ERR\_FORM\_SIZE**

Значение: 2; Размер загружаемого файла превысил значение MAX\_FILE\_SIZE, указанное в HTML-форме.

**UPLOAD\_ERR\_PARTIAL**

Значение: 3; Загружаемый файл был получен только частично.

**UPLOAD\_ERR\_NO\_FILE**

Значение: 4; Файл не был загружен.

**UPLOAD\_ERR\_NO\_TMP\_DIR**

Значение: 6; отсутствует временная папка.

**UPLOAD\_ERR\_CANT\_WRITE**

Value: 7; Failed to write file to disk. Introduced in PHP 5.1.0.

**UPLOAD\_ERR\_EXTENSION**

Value: 8; A PHP extension stopped the file upload. PHP does not provide a way to ascertain which extension caused the file upload to stop; examining the list of loaded extensions with phpinfo() may help. Introduced in PHP 5.2.0.

bool **move\_uploaded\_file** ( string $filename , string $destination ) Эта функция проверяет, является ли файл **filename** загруженным на сервер (переданным по протоколу HTTP POST). Если файл действительно загружен на сервер, он будет перемещён в место, указанное в аргументе **destination**.

**if ( $\_FILES["userfile"]["error"] != UPLOAD\_ERR\_OK ){  
switch($\_FILES["userfile"]["error"]){**

**case UPLOAD\_ERR\_INI\_SIZE:**

**echo "Превышен максимально допустимый размер"; break;**

**case UPLOAD\_ERR\_FORM\_SIZE:**

**echo "Превышено значение MAX\_FILE\_SIZE"; break;**

**case UPLOAD\_ERR\_PARTIAL:**

**echo "Файл загружен частично"; break;**

**case UPLOAD\_ERR\_NO\_FILE:**

**echo "Файл не был загружен"; break;**

**case UPLOAD\_ERR\_NO\_TMP\_DIR:**

**echo "Отсутствует временная папка"; break;**

**case UPLOAD\_ERR\_CANT\_WRITE:**

**echo "Не удалось записать файл не диск";**

**}**

**}else{**

**echo "Размер загруженного файла: " . $\_FILES["userfile"]["size"];**

**echo "Тип загруженного файла: " . $\_FILES["userfile"]["type"];**

**move\_uploaded\_file($\_FILES**["**userfile**"]["**tmp\_name**"], "**upload**/" . **$\_FILES**["**userfile**"]["**name**"]);}

Работа с почтой:

// Настройка php.ini

SMTP – адрес почтового сервера, кому передается письмо

smtp\_port – почтовый порт

sendmail\_from – кто будет писаться во from по умолчанию

bool **mail** ( string $to , string $subject , string $message [, string $additional\_headers [,string $additional\_parameters ]] ) - Отправляет электронную почту.

**to -**Получатель, или получатели письма.

**Subject -** Тема отправляемого письма.

**Message -** Отправляемое сообщение.

mail('caffeinated@example.com', 'My Subject', "Line 1\r\nLine 2\r\nLine 3");

|  |  |
| --- | --- |
| **\_\_LINE\_\_** | Текущий номер строки в файле. |
| **\_\_FILE\_\_** | Полный путь и имя текущего файла с развернутыми симлинками. Если используется внутри подключаемого файла, то возвращается имя данного файла. |
| **\_\_DIR\_\_** | Директория файла. Если используется внутри подключаемого файла, то возвращается директория этого файла. Это эквивалентно вызову *dirname(\_\_FILE\_\_)*. Возвращаемое имя директории не оканчивается на слеш, за исключением корневой директории. |
| **\_\_FUNCTION\_\_** | Имя функции. |
| **\_\_CLASS\_\_** | Имя класса. Это имя содержит название пространства имен, в котором класс был объявлен (например, *Foo\Bar*). Обратите внимание, что начиная с PHP5.4 \_\_CLASS\_\_ также работает в трейтах. При использовании в методах трейтов \_\_CLASS\_\_ является именем класса, в котором эти методы используется. |
| **\_\_TRAIT\_\_** | Имя трейта. Это имя содержит название пространства имен, в котором трейт был объявлен (например, *Foo\Bar*). |
| **\_\_METHOD\_\_** | Имя метода класса. |
| **\_\_NAMESPACE\_\_** | Имя текущего пространства имен. |
| **PHP\_VERSION** | Возвращает строку с номером версии текущего PHP-анализатора или расширения |
| **PHP\_OS** | Для вывода названия операционной системы |

Суперглобальные переменные — Суперглобальные переменные - это встроенные переменные, которые всегда доступны во всех областях видимости

[***$GLOBALS***](http://php.net/manual/ru/reserved.variables.globals.php)— Ссылки на все переменные глобальной области видимости

[***$\_SERVER***](http://php.net/manual/ru/reserved.variables.server.php) -- $HTTP\_SERVER\_VARS [удалено] — Информация о сервере и среде исполнения

[***$\_GET***](http://php.net/manual/ru/reserved.variables.get.php) -- $HTTP\_GET\_VARS [deprecated] — GET-переменные HTTP

[***$\_POST***](http://php.net/manual/ru/reserved.variables.post.php) -- $HTTP\_POST\_VARS [deprecated] — HTTP POST variables

[***$\_FILES***](http://php.net/manual/ru/reserved.variables.files.php) -- $HTTP\_POST\_FILES [устаревшее] — Переменные файлов, загруженных по HTTP

[***$\_COOKIE***](http://php.net/manual/ru/reserved.variables.cookies.php) -- $HTTP\_COOKIE\_VARS [устаревшее] — HTTP Куки

[***$\_SESSION***](http://php.net/manual/ru/reserved.variables.session.php) -- $HTTP\_SESSION\_VARS [устаревшее] — Переменные сессии

[***$\_REQUEST***](http://php.net/manual/ru/reserved.variables.request.php)— Переменные HTTP-запроса (по умолчанию содержит данные переменных [$\_GET](http://php.net/manual/ru/reserved.variables.get.php), [$\_POST](http://php.net/manual/ru/reserved.variables.post.php) и[$\_COOKIE](http://php.net/manual/ru/reserved.variables.cookies.php))

[***$\_ENV***](http://php.net/manual/ru/reserved.variables.environment.php)-- $HTTP\_ENV\_VARS [устаревшее] — Переменные окружения

string **phpversion** ([ string $extension ] ) — Выдает текущую версию PHP

string **getenv** ( string $varname ) — Получение значения переменной окружения. Список всех переменных окружения можно посмотреть с помощью функции [phpinfo()](http://php.net/manual/ru/function.phpinfo.php)

**PHP\_SELF** Имя файла в настоящее время выполняющегося сценария, относительно корня документа. $\_SERVER ['PHP\_SELF']

**argv** Массив параметров, переданных сценарию. При использовании метода GET, он будет содержать строку запроса.

**rgc** Содержит номер параметров командной строки, переданных сценарию (если они переданы из командной строки).

**GATEWAY\_INTERFACE** Содержит версию CGI, которую использует веб-сервер (Например, CGI/1.1).

**SERVER\_NAME** Имя веб-сервера, под которым текущий скрипт выполняется.

**SERVER\_SOFTWARE** Идентификационная строка веб-сервера, которая передается через заголовки при ответах на запросы сервера.

**SERVER\_PROTOCOL** Имя и версия используемого протокола HTTP (Например, HTTP/1.1).

**REQUEST\_METHOD** Используемый метод запроса веб-страницы ( 'GET', 'HEAD', 'POST', 'PUT' ).

**REQUEST\_TIME** Время начала запроса веб-страницы. Доступна, начиная с PHP 5.1.0

**QUERY\_STRING** Строка запроса веб-страницы, если она существует, через которую был получен доступ к странице.

**DOCUMENT\_ROOT** Корневая директория, под которой выполняется текущий скрипт. Определяется конфигурацией веб-сервера.

**HTTP\_ACCEPT** Содержание заголовка Accept, если он есть.

**HTTP\_ACCEPT\_CHARSET** Содержание заголовка Accept-Charset, если он есть. Пример: 'iso-8859-1,\*,utf-8'.

**HTTP\_ACCEPT\_ENCODING** Содержание заголовка Accept-Encoding, если он есть. Пример: 'gzip'.

**HTTP\_ACCEPT\_LANGUAGE** Содержание заголовка Accept-Language, если он есть. Пример: 'en'.

**HTTP\_CONNECTION** Содержание заголовка Connection, если он есть. Пример: 'Keep-Alive'.

**HTTP\_HOST** Содержание заголовка Host, если он есть.

**HTTP\_REFERER** Адрес страницы, которая ссылается на текущий документ посредством пользовательского клиента

**HTTP\_USER\_AGENT** Содержание заголовка 'User-Agent'. Для получения подобной информации вы также можете использовать

ункцию [**get\_browser()**](http://www.php.su/functions/?get_browser)

**REMOTE\_ADDR** IP-адрес удаленного пользователя, который обращается к текущей странице.

**REMOTE\_HOST** Имя хоста удаленного пользователя, который обращается к текущей странице

**REMOTE\_PORT** Порт, используемый для соединения с веб-сервером

**SCRIPT\_FILENAME** Абсолютный путь для текущего скрипта.

**SERVER\_ADMIN** Значение, задаваемое SERVER\_ADMIN (для Apache).

**SERVER\_PORT** Текущий порт веб-сервера, используемы для предечи данных по протоколу HTTP. По-умолчанию '80'.

**SERVER\_SINGATURE** Строка, содержащая версию веб-сервера или имя виртуального хоста.

**PATH\_TRANSLATED** Базовый путь к текущему сценарию.

**SCRIPT\_NAME** Содержит путь и имя файла текущего исполняемого скрипта.

**REQUEST\_URI** URI для текущей страницы, например, '/index.html'.

**PHP\_AUTH\_DIGEST** Если PHP работает как модуль Apache, переменная используется в аутентификации по протоколу HTTP, для проверки подлинности.

**PHP\_AUTH\_USER** Если PHP работает как модуль Apache, либо как ISAPI IIS, данная переменная содержит имя авторизируемого пользователя при аутентификации по протоколу HTTP.

**PHP\_AUTH\_PW** Если PHP работает как модуль Apache, либо как ISAPI IIS, данная переменная содержит пароль авторизируемого пользователя при аутентификации по протоколу HTTP.

**AUTH\_TYPE** Если PHP работает как модуль Apache, либо как ISAPI IIS, данная переменная содержит тип используемой аутентификации по протоколу HTTP.

**HTTP** ([англ.](https://ru.wikipedia.org/wiki/Английский_язык) *HyperText Transfer Protocol* — «протокол передачи [гипертекста](https://ru.wikipedia.org/wiki/Гипертекст)») — [протокол](https://ru.wikipedia.org/wiki/Сетевой_протокол) [прикладного уровня](https://ru.wikipedia.org/wiki/Протоколы_прикладного_уровня) передачи данных.

**Методы: GET HEAD POST PUT PATCH DELETE TRACE CONNECT**

### Коды состояния

**[1xx](https://ru.wikipedia.org/wiki/Список_кодов_состояния_HTTP" \l "1xx) Informational («Информационный»)**

**[2xx](https://ru.wikipedia.org/wiki/Список_кодов_состояния_HTTP" \l "2xx) Success («Успех»)**

**[3xx](https://ru.wikipedia.org/wiki/Список_кодов_состояния_HTTP" \l "3xx) Redirection («Перенаправление») 302-временное, 301-постоянное**

**[4xx](https://ru.wikipedia.org/wiki/Список_кодов_состояния_HTTP" \l "4xx) Client Error («Ошибка клиента») 402-доступ к сайту платный 400- не понял**

**[5xx](https://ru.wikipedia.org/wiki/Список_кодов_состояния_HTTP" \l "5xx) Server Error («Ошибка сервера»)**

**[GET](https://ru.wikipedia.org/wiki/HTTP" \l "GET) Запрос клиента:**

**[GET](https://ru.wikipedia.org/wiki/HTTP" \l "GET) /wiki/*страница* HTTP/1.1**

**Host**: ru.wikipedia.org

**User-Agent**: Mozilla/5.0 (X11; U; Linux i686; ru; rv:1.9b5) Gecko/2008050509 Firefox/3.0b5

**Accept**: text/html

**Connection**: close

*(пустая строка)*

**Ответ сервера:**

**HTTP/1.1 200 OK**

Date: Wed, 11 Feb 2009 11:20:59 GMT

**Server**: Apache

**X-Powered-By:** PHP/5.2.4-2ubuntu5wm1

**Last-Modified**: Wed, 11 Feb 2009 11:20:59 GMT

**Content-Language**: ru

**Content-Type**: text/html; charset=utf-8

**Set-Cookie: name=value**

**Content-Length**: 1234

**Connection**: close

*(пустая строка)*

*(далее следует запрошенная страница в* [*HTML*](https://ru.wikipedia.org/wiki/HTML)*)*

**[POST](https://ru.wikipedia.org/wiki/HTTP" \l "GET) Запрос клиента:**

POST /send-message.html HTTP/1.1

Host: mail.example.com

Referer: <http://mail.example.com/send-message.html>

User-Agent: BrowserForDummies/4.67b

Content-Type: multipart/form-data; boundary="Asrf456BGe4h"

Content-Length: *(суммарный объём, включая дочерние заголовки)*

Connection: keep-alive

Keep-Alive: 300

*(пустая строка)*

*(отсутствующая преамбула)*

--Asrf456BGe4h

Content-Disposition: form-data; name="DestAddress"

*(пустая строка)*

[brutal-vasya@example.com](mailto:brutal-vasya@example.com)

--Asrf456BGe4h

Content-Disposition: form-data; name="MessageTitle"

*(пустая строка)*

Я негодую

--Asrf456BGe4h

Content-Disposition: form-data; name="MessageText"

*(пустая строка)*

Привет, Василий! Твой ручной лев, которого ты оставил

у меня на прошлой неделе, разодрал весь мой диван.

Пожалуйста, забери его скорее!

Во вложении две фотки с последствиями.

--Asrf456BGe4h

Content-Disposition: form-data; name="AttachedFile1"; filename="horror-photo-1.jpg"

Content-Type: image/jpeg

*(пустая строка)*

*(двоичное содержимое первой фотографии)*

--Asrf456BGe4h

Content-Disposition: form-data; name="AttachedFile2"; filename="horror-photo-2.jpg"

Content-Type: image/jpeg

*(пустая строка)*

*(двоичное содержимое второй фотографии)*

--Asrf456BGe4h—

*(отсутствующий эпилог)*

void **header** ( string $string [, bool $replace = true [, int $http\_response\_code ]] ) — Отправка HTTP заголовка

|  |  |  |
| --- | --- | --- |
| Location | [URI](https://ru.wikipedia.org/wiki/URI) по которому клиенту следует перейти или URI созданного ресурса. | (“Location: [http://example.com/about.html](http://example.com/about.html" \l "contacts)”, true, 301) |
| Accept | Список допустимых форматов ресурса. | Accept: text/plain |
| Accept-Сharset | Перечень поддерживаемых кодировок для предоставления пользователю. | Accept-Charset: utf-8 |
| Content-Type | Формат и способ представления сущности. | Content-Type: text/html;charset=utf-8 |
| Content-Length | Размер содержимого сущности в [октетах](https://ru.wikipedia.org/wiki/Октет_(информатика)). | Content-Length: 1348 |
| Host | [Доменное имя](https://ru.wikipedia.org/wiki/Доменное_имя) и порт [хоста](https://ru.wikipedia.org/wiki/Хост) запрашиваемого ресурса. | Host: ru.wikipedia.org |
| Pragma | Особенные опции выполнения операции. |  |
| [Referer](https://ru.wikipedia.org/wiki/Список_заголовков_HTTP" \l "Referer) | URI ресурса, после которого клиент сделал текущий запрос. | header("Refresh: 3; url= http://from.php.local/test2.php"); |
| [User-Agent](https://ru.wikipedia.org/wiki/Список_заголовков_HTTP" \l "User-Agent) | Список названий и версий клиента и его компонентов с комментариями. | User-Agent: Mozilla/5.0 (X11; Linux i686; rv:2.0.1) Gecko/20100101 Firefox/4.0.1 |
| Refresh | Обновление страницы или переход на другую страницу через указанный отрезок времени | “Refresh:3; url=http://www.google.ru” |
| Content-Disposition | можете использовать для предоставления рекомендуемого файла и форсировать отображение браузером диалога save | Content-Disposition: attachment; filename=\”myfile.txt\” |
| Cache-Control | Основные директивы для правления [кэшированием](https://ru.wikipedia.org/wiki/Кэш). | Cache-Control: no-cache Cache-Control: no-store Cache-Control: max-age=3600 |
| Expires | [Дата](https://ru.wikipedia.org/wiki/Заголовки_HTTP" \l ".D0.94.D0.B0.D1.82.D0.B0_.D0.B8_.D0.B2.D1.80.D0.B5.D0.BC.D1.8F) предполагаемого истечения срока актуальности сущности. | Expires: Tue, 31 Jan 2012 15:02:53 GMT  Если есть max-ageб то Expires-игнорится |
| **Set-cookie** |  | header("Set-Cookie: {$name}={$value}; EXPIRES{$date};"); |

Ка́пча (от **CAPTCHA** — [англ.](https://ru.wikipedia.org/wiki/Английский_язык) ***C****ompletely* ***A****utomated* ***P****ublic* ***T****uring test to tell* ***C****omputers and* ***H****umans* ***A****part* — полностью автоматизированный публичный [тест Тьюринга](https://ru.wikipedia.org/wiki/Тест_Тьюринга) для различения компьютеров и людей) — компьютерный тест, используемый для того, чтобы определить, кем является пользователь системы: человеком или компьютером.

**Работа с COOKIE**

**Ку́ки** (от [англ.](https://ru.wikipedia.org/wiki/Английский_язык) *cookie* — печенье) — небольшой фрагмент данных, отправленный [веб-сервером](https://ru.wikipedia.org/wiki/Веб-сервер) и хранимый на [компьютере](https://ru.wikipedia.org/wiki/Компьютер)пользователя.

* имя Cookie – name
* значение переменной – value
* срок жизни (по умолчанию до закрытия браузера) в секундах – expire
* путь Cookie (можно установить для страниц из определенного каталога)
* домен Cookie (можно установить для нескольких доменов)
* защита (если true -> соединение по защищенному каналу SSL)
* HTTPONLY

**Браузер** ←Set-Cookie:name=value **сервер || браузер** Cookie: name=value→ **сервер**

bool **setcookie** ( string $name [, string $value [, int $expire = 0 [, string $path [, string$domain [, bool $secure = false [, bool $httponly = false ]]]]]] ) — Посылает cookie **setcookie**("**name**", "**John**", **time**()**+3600**, "/**docs**/"); (два постедних параметра - безопасность)  
 **setcookie** ( “name”) or **setcookie** ( “name”, “”) - удалить cookie

**// Для сохранения целостности**

$str = **base64\_encode**( **serialize**($user) );

**setcookie**("user", $str);

$user = **unserialize**( **base64\_decode**($\_COOKIE["user"]) );

**Session identifier (SID)** идентификатор сесии

bool **session\_start** ( void ) — Start new or resume existing session

bool **session\_register** ( [mixed](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) $name [, [mixed](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) $... ] ) – регистрация переменной в сессии **session\_register** (“a”,”username”, “password”);

bool **session\_destroy** ( void )- удалить данные из файла сессии но не из cookie

bool **session\_is\_registered** ( string $name ) — Определяет, зарегистрирована ли глобальная переменная в сессии

bool **session\_unregister** ( string $name ) — удалить из сессии переменную

void **session\_unset** ( void )- удалить все переменные из сессии

string **session\_name** ([ string $name ] ) – возвращает имя сессии

string **session\_id** ([ string $id ] ) – узнать SID

bool **session\_regenerate\_id** ([ bool $delete\_old\_session = false ] ) – каждый раз сессия с новым ID

string **serialize** ( [mixed](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) $value ) — Генерирует пригодное для хранения представление переменной (массив в строку)

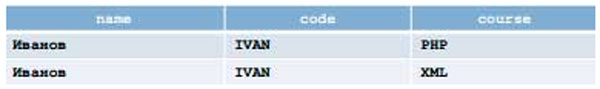
[mixed](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) **unserialize** ( string $str ) — Создает PHP-значение из хранимого представления (из строки в массив)

|  |  |
| --- | --- |
| **session.auto\_start** boolean [‘0’] – автостарт сессий | // session.auto\_start = 0 |
| **session**.**name** string [‘PHPSESSID’] - | ini\_set('session.name', 'PHPSESSID'); |
| **session.save\_path** string [‘’] – где сохранять (по умолчанию во врем. папке системы) | ini\_set('session.save\_path', ''); |
| **session.gc\_maxlifetime** integer [‘1440’] – задать срок жизни сессии | ini\_set('session.gc\_maxlifetime', '1440'); |
| **session.cookie\_lifetime** integer [‘0’] – задать срок жизни сессии | ini\_set('session.cookie\_lifetime', '0'); |
| **session.use\_cookies** boolean [‘1’] | ini\_set('session.cookie\_httponly', ''); |
| **session.use\_only\_cookies** Boolean [‘1’ c PHP 5.3.0] | ini\_set('session.cookie\_path', '/') |
| session.use\_trans\_sid boolean [‘0’] – если откл. Cookie и true php доклеивает данные как “GET” | |

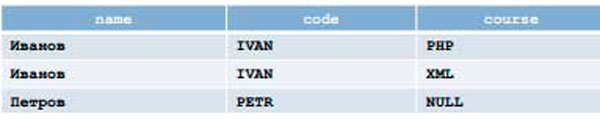
**Объединение таблиц:**



**SELECT** t.name, t.code, l.course  
**FROM** teachers t // псевдоним таблицы teachers – t  
**INNER JOI**N lessons l **ON** t.id = l.tid //псевдоним таблицы lessons -l



**SELECT** t.name, t.code, l.course  
**FROM** teachers t // псевдоним таблицы  
**LEFT** **OUTER** **JOIN** lessons l **ON** t.id = l.tid //левое внешнее объединение



**SELECT DISTINCT** teachers.name  
**FROM** teachers **INNER** **JOIN**  
(lessons **INNER** **JOIN** courses  
**ON** lessons.course = courses.id)  
**ON** teachers.id = lessons.teacher  
**WHERE** courses.title **LIKE** 'Web%'  
**ORDER** **BY** teachers.name

**INSERT – вставка (сроки задаются в апострофах)**

**INSERT INTO** courses  
**VALUES** (Null, ‘PHP’, ‘…’, 40)

**INSERT INTO** courses (title, length)  
**VALUES** (‘PHP’, 40)

**DELETE – удаление записей**

**DELETE FROM** lessons  
**WHERE** lessons.date = ‘2005-06-11’ // без where удалит все записи в таблице

**UPDATE – изменение имеющейся записи**

**UPDATE** teachers  
**SET** zarplata = zarplata \* 2,  
 premia = premia \* 10  
**WHERE** name **LIKE** ‘Иванов%’ // без where изменятся все указаные записи  
 **OR** name **LIKE** ‘Петров%’ **OR** name **LIKE** ‘Сидоров%’

**UPDATE** teachers  
**SET** zarplata = zarplata \* 2,  
 premia = premia \* 10  
**WHERE** name **IN  
 (**‘Иванов’, ‘Петров’**,** ‘Сидоров’)

**Подготовленные запросы  
$sql = "INSERT INTO users(name, email, age) VALUES(?, ?, ?)";  
// Уважаемый сервер, вот запрос - разбери его  
$stmt = mysqli\_prepare($link, $sql);  
// Уважаемый сервер, вот параметры для запроса  
mysqli\_stmt\_bind\_param($stmt, "ssi", $name, $email, $age);  
// А теперь, исполни подготовленный запрос с переданными параметрами  
mysqli\_stmt\_execute($stmt);  
mysqli\_stmt\_close($stmt);**

**Конфигурационный файл my.ini**

**port = 3306**

**datadir = "c:\\users\\public\\openserver\\userdata\\MySQL-5.5"**

**default-storage-engine = InnoDB**

**init-connect = "SET NAMES utf8"**

**interactive\_timeout = 30**

**wait\_timeout = 60**

**connect\_timeout = 5**

**PHP.ini**

**mysqli\_connect** ([ string $host = ini\_get("mysqli.default\_host") [, string $username = ini\_get("mysqli.default\_user") [, string $passwd = ini\_get ("mysqli.default\_pw") [, string$dbname = "" [, int $port = ini\_get("mysqli.default\_port") [, string $socket = ini\_get("mysqli.default\_socket") ]]]]]] ) — Устанавливает новое соединение с сервером MySQL $link= **mysqli\_connect**('local2.host', 'root', '', 'web');

bool **mysqli\_close** ( [mysqli](https://php.net/manual/ru/class.mysqli.php) $link ) — Закрывает ранее открытое соединение с базой данных

int **mysqli\_connect\_errno** ( void ) — Возвращает код ошибки последней попытки соединения

string **mysqli\_connect\_error** ( void ) — Возвращает описание последней ошибки подключения

int **mysqli\_ errno** ( void ) — Возвращает код ошибки or die(mysqli\_errno($link));

string **mysqli\_error** ( void ) — Возвращает описание ошибки or die(mysqli\_error($link));

bool **mysqli\_select\_db** ( [mysqli](https://php.net/manual/ru/class.mysqli.php) $link , string $dbname ) — Устанавливает базу данных для выполняемых запросов

[mixed](https://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) **mysqli\_query** ( [mysqli](https://php.net/manual/ru/class.mysqli.php) $link , string $query [, int $resultmode = MYSQLI\_STORE\_RESULT ] ) — Выполняет запрос к базе данных $link=**mysqli\_query** ($link , ’ **SELECT** \* **FROM** teachers’); **mysqli\_query** ($link , “ **SET** NAMES ‘cp1251’”);

mixed **mysqli\_fetch\_array** ( mysqli\_result result [, int resulttype] ) — Обрабатывает ряд результата запроса, возвращая ассоциативный массив, численный массив или оба. Каждый вызов работает с одной строкой.

|  |  |  |  |
| --- | --- | --- | --- |
| **mysqli\_fetch\_row** ( $link ) | **mysqli\_fetch\_array** ($link, **MYSQLI\_NUM**) | индексированный массив | |
| **mysqli\_fetch\_assoc** ( $link ) | **mysqli\_fetch\_array** ($link, **MYSQLI\_ASSOC**) | ассоциативный массив | |
| **mysqli\_fetch\_all($result, MYSQLI\_ASSOC)** | | | массив массивов |

mixed **mysqli\_fetch\_all** ( mysqli\_result result [, int resulttype] ) — Обрабатывает ряд результата запроса, возвращая ассоциативный массив в каждом элементе которого лежит массив. Для избежания цикла.

int **mysqli\_num\_rows** ( [mysqli\_result](http://php.net/manual/ru/class.mysqli-result.php) $result ) — Получает число рядов в результирующей выборке

int **mysqli\_num\_fields** ( [mysqli\_result](http://php.net/manual/ru/class.mysqli-result.php) $result ) — Получает число полей в результирующей выборке

int **mysqli\_affected\_rows** ( [mysqli](http://php.net/manual/ru/class.mysqli.php) $link ) — Получает число строк, затронутых предыдущей операцией рядов

int **mysqli\_field\_count** ( [mysqli](http://php.net/manual/ru/class.mysqli.php) $link ) — Возвращает число столбцов, затронутых последним запросом

[mixed](http://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) **mysqli\_insert\_id** ( [mysqli](http://php.net/manual/ru/class.mysqli.php) $link ) — Возвращает автоматически генерируемый ID, используя последний запрос

string **mysql\_real\_escape\_string** ( string $unescaped\_string [, resource $link\_identifier = NULL] ) — Экранирует специальные символы в строках для использования в выражениях SQL

**$mysqli->real\_escape\_string()** - с её помощью необходимо обрабатывать все данные перед помещением их в запрос.

**$mysqli->affected\_rows** – количество затронутых строк предыдущим запросом не на выборку (Update, Insert, Delete).

**$mysqli->insert\_id** – автоматически сгенерированный ID для последнего запроса вставки (Insert).

**$mysqli->field\_count** – количество столбцов в результате запроса выборки (Select).

**$link = mysqli\_connect**($host,$user,$password,$database**) or die**("Ошибка " . mysqli\_error($link));

**$query = "SELECT \* FROM** oc\_customer";

**$result = mysqli\_query**($link,$query**) or die**("Ошибка " . mysqli\_error($link));

**if**(**$result**){

**echo**($link->field\_count);

}

**mysqli\_close**($link);

cURL

resource curl\_init ([ string $url = NULL ] ) - Инициализирует сеанс cURL

bool curl\_setopt ( resource $ch , int $option , [mixed](http://php.net/manual/ru/language.pseudo-types.php" \l "language.types.mixed) $value ) - Устанавливает параметр для сеанса CURL

|  |  |
| --- | --- |
| **CURLOPT\_RETURNTRANSFERTRUE** | для возврата результата передачи в качестве строки из[curl\_exec()](http://php.net/manual/ru/function.curl-exec.php) вместо прямого вывода в браузер. |
| **CURLOPT\_HEADER** | **TRUE** для включения заголовков в вывод. |
| **CURLOPT\_FOLLOWLOCATION** | **TRUE** для следования любому заголовку *"Location: "*, отправленному сервером в своем ответе (учтите, что это происходит рекурсивно, PHP будет следовать за всеми посылаемыми заголовками *"Location: "*, за исключением случая, когда установлена константа **CURLOPT\_MAXREDIRS**). |
| **CURLOPT\_SSL\_VERIFYHOST** | Используйте *1* для проверки существования общего имени в сертификате SSL. Используйте *2* для проверки существования общего имени и также его совпадения с указанным хостом. *0* чтобы не проверять имена. В боевом окружении значение этого параметра должно быть *2* (установлено по умолчанию). |
| **CURLOPT\_SSL\_VERIFYPEER** | **FALSE** для остановки cURL от проверки сертификата узла сети. Альтернативные сверяемые сертификаты могут быть указаны с помощью параметра **CURLOPT\_CAINFO** или директории с сертификатами, указываемой параметром **CURLOPT\_CAPATH**. |
| **CURLOPT\_NOBODY** | **TRUE** для исключения тела ответа из вывода. Метод запроса устанавливается в HEAD. Смена этого параметра в **FALSE** не меняет его обратно в GET. |
| **CURLOPT\_COOKIE** | Содержимое заголовка *"Cookie: "*, используемого в HTTP-запросе. Обратите внимание, что несколько cookies разделяются точкой с запятой с последующим пробелом (например, "*fruit=apple; colour=red*") |
| **CURLOPT\_COOKIEJAR** | Имя файла, в котором будут сохранены все внутренние cookies текущей передачи после закрытия дескриптора, например, после вызова curl\_close |
| **CURLOPT\_COOKIEFILE** | Имя файла, содержащего cookies. Данный файл должен быть в формате Netscape или просто заголовками HTTP, записанными в файл. Если в качестве имени файла передана пустая строка, то cookies сохраняться не будут, но их обработка все еще будет включена. |
| **CURLOPT\_COOKIESESSION** | **TRUE** для указания текущему сеансу начать новую "сессию" cookies. Это заставит libcurl проигнорировать все "сессионные" cookies, которые она должна была бы загрузить, полученные из предыдущей сессии. По умолчанию, libcurl всегда сохраняет и загружает все cookies, вне зависимости от того, являются ли они "сессионными" или нет. "Сессионные" cookies - это cookies без срока истечения, которые должны существовать только для текущей "сессии". |
| **CURLOPT\_USERAGENT** | Содержимое заголовка *"User-Agent: "*, посылаемого в HTTP-запросе. |
| **CURLOPT\_POSTFIELDS** | Все данные, передаваемые в HTTP POST-запросе. Для передачи файла, укажите перед именем файла *@*, а также используйте полный путь к файлу. Тип файла также может быть указан с помощью формата '*;type=mimetype*', следующим за именем файла. Этот параметр может быть передан как в качестве url-закодированной строки, наподобие '*para1=val1&para2=val2&...*', так и в виде массива, ключами которого будут имена полей, а значениями - их содержимое. Если **value** является массивом, заголовок *Content-Type* будет установлен в значение *multipart/form-data*. Начиная с версии PHP 5.2.0, при передаче файлов с префиксом *@*, **value** должен быть массивом. С версии PHP 5.5.0, префикс *@* устарел и файлы можно отправлять с помощью **CURLFile**. Префикс *@* можно отключить, чтобы можно было передавать значения, начинающиеся с *@*, задав опцию **CURLOPT\_SAFE\_UPLOAD** в значение **TRUE**. |
| **CURLOPT\_HTTPHEADER** | Массив устанавливаемых HTTP-заголовков, в формате array('Content-type: text/plain', 'Content-length: 100') |

curl\_exec ( resource $ch ) — Выполняет запрос cURL

curl\_close ( resource $ch ) — Завершает сеанс cURL

http://php.net/manual/ru/function.curl-setopt.php