Методические указания по теме
“Встроенные функции PHP”

Функции для работы со строками

Функции для работы с символами

Функция	Описание
strlen($str)	Возвращает количество символов в строке $str
chr($ascii)	Принимает в качестве аргумента ASCII-код $ascii символа и возвращает соответствующий этому коду фактический символ
ord($str)	Возвращает ASCII-код символа $str, переданного ей в качестве аргумента
count_chars($str [, $mode])	Подсчитывает количество вхождений каждого из символов с ASCII-кодами в диапазоне 0...255 в строку $str. Необязательный параметр $mode позволяет задать формат результата
str_shuffle($str)	Возвращает копию строки $str, символы которой перемешаны в случайном порядке
strrev($str)	Возвращает копию строки $str, символы которой перевернуты, т. е. начальные символы помещены в конец, а конечные символы в начало
ctype_alnum($str)	Возвращает TRUE, если строка $str содержит только буквы английского алфавита и цифры, в противном случае возвращается FALSE
ctype_alpha($str)	Возвращает TRUE, если строка $str содержит только буквы английского алфавита, в противном случае возвращается FALSE
ctype_xdigit($str)	Возвращает TRUE, если строка $str содержит только цифры и буквы A, B, C, D, E, F в верхнем или нижнем регистрах (т. е. символы, из которых состоят шестнадцатеричные числа). В противном случае возвращается FALSE
ctype_cntrl($str)	Возвращает TRUE, если строка $str содержит только управляющие символы (перевод строки, символ табуляции и т. п.), в противном случае возвращается FALSE
ctype_digit($str)	Возвращает TRUE, если строка $str содержит только цифры, в противном случае возвращается FALSE
ctype_print($str)	Возвращает TRUE, если строка $str содержит только видимые символы, в противном случае возвращается FALSE
ctype_graph($str)	Возвращает TRUE, если строка $str содержит только видимые графические символы, в противном случае возвращает FALSE. Пробел, символ табуляции традиционно относятся к видимым символам в

	таблице ASCII-кодов, однако не входят в группу графических символов — для них функция ctype_graph() вернет FALSE
ctype_punct($str)	Возвращает TRUE, если строка $str содержит только символы пунктуации (видимые символы за исключением символов английского алфавита, цифр и пробельных символов), в противном случае возвращается FALSE
ctype_lower($str)	Возвращает TRUE, если строка $str содержит только буквы английского алфавита в нижнем регистре, в противном случае возвращается FALSE
ctype_upper($str)	Возвращает TRUE, если строка $str содержит только буквы английского алфавита в верхнем регистре, в противном случае возвращается FALSE

Функции для поиска в строке

Задача поиска в строках является очень распространенной при создании Web-приложений. Существуют два подхода к решению проблемы: при помощи строковых функций и регулярных выражений. Регулярные выражения позволяют решать более сложные задачи, и в то же время они менее производительны, т. к. поиск с использованием регулярных выражений требует значительных расчетов. В таблице приводится список строковых функций, осуществляющих поиск в строке.

Функция	Описание
substr($str, $start [, $length])	Возвращает для строки $str подстроку, которая начинается с символа $start (отсчет с нуля) и имеет $length символов. Если количество символов $length не указано, предполагается, что подстрока должна включать все символы до конца строки $str
substr_count($str, $search [, $offset [, $length]])	Возвращает количество вхождений подстроки $search в строку $str. Поиск осуществляется с позиции $offset в подстроке, длиной $lenght (если эти два параметра указаны)
strpos($str, $search [, $offset])	Возвращает позицию первого вхождения подстроки $search в строку $str. Поиск может начинаться с позиции $offset, если указывается третий параметр. Возвращает FALSE, если подстрока $search не найдена
stripos($str, $search [, $offset])	Возвращает позицию первого вхождения подстроки $search с конца строки $str. Необязательный параметр $offset позволяет задать позицию с конца строки, начиная с которой следует осуществлять поиск. Возвращает FALSE, если подстрока $search не

	найдена
strstr($str, $search [, $before])	Ищет первое вхождение подстроки $search в строку $str. По умолчанию функция возвращает подстроку, начиная с первого вхождения $search до конца строки $str, однако если необязательный параметр $before принимает параметр TRUE, возвращается подстрока, предшествующая первому вхождению подстроки $search. Возвращает FALSE, если подстрока $search не найдена
stristr($str, $search [, $before])	Функция stristr() аналогична функции strstr(), за исключением того факта, что в процессе поиска не учитывается регистр
strchr()	Синоним функции strstr()
strrchr($str, $search)	Ищет последнее вхождение подстроки $search в строку $str и возвращает подстроку, начиная с $search и до конца строки $str. Возвращает FALSE, если подстрока $search не найдена
strpbrk($str, $char_list)	Ищет в строке $str символы из строки $char_list и возвращает подстроку, начиная с первого найденного символа и до конца строки. Если ни один символ не найден, возвращается FALSE

Замена в тексте

Функции замены осуществляют преобразование строк, связанные с заменой одних подстрок другими, а также удалением подстрок.

Функция	Описание
str_replace($search, $replace, $str [, &$count])	Возвращает копию строки $str, в которой подстрока $search заменена на $replace. В необязательном параметре $count может быть возвращено количество осуществленных замен. Вместо отдельных строк, параметры $search и $replace могут содержать массивы строк
str_ireplace($search, $replace, $str [, &$count])	Функция str_ireplace() аналогична функции str_replace() за исключением того, что поиск осуществляется без учета регистра
substr_replace($str, $replacement, $start [, $length])	Возвращает копию строки $str, в которой подстрока, начинающаяся с символа $start и длиной $length, заменена подстрокой replacement

strtr($str, $from, $to)	Возвращает копию строки $str, в которой каждый символ из строки $from заменен оответствующим символом из строки $to
strtr($str, $replace_pairs)	Возвращает копию строки $str, в которой замены осуществлены в соответствии с массивом $replace_pairs (ключи заменяются значениями массива)
trim($str [, $charlist])	Удаляет пробельные символы из начала и конца строки $str
rtrim($str [,$charlist])	Удаляет пробельные символы из конца строки $str. Необязательный параметр $charlist позволяет уточнить список пробельных символов
chop()	Синоним для функции rtrim()
ltrim($str [, $charlist])	Удаляет пробельные символы из начала строки $str. Необязательный параметр $charlist позволяет уточнить список пробельных символов

Преобразование регистра

Функция	Описание
strtolower($str)	Преобразует строку $str в нижний регистр
strtoupper($str)	Преобразует строку $str в верхний регистр
ucfirst($str)	Преобразует первый символ строки $str в верхний регистр
ucwords($str)	Преобразует каждое слово строки в верхний регистр

Работа с HTML-кодом

Так как PHP проектировался как язык, специально предназначенный для Web, среди строковых функций имеется несколько, специально предназначенных для Web-разработки.

Функция	Описание
nl2br($str)	Добавляет перед переводами строки HTML-тег перевода строки
htmlspecialchars($str [, $quote_style [, $charset [,	Преобразует HTML-теги в строке $str в

$double_encode]]])	представление, делающее их видимыми в браузере. Необязательный параметр $quote_style определяет режим замены кавычек и может принимать три значения: ENT_NOQUOTES — двойные и одиночные кавычки остаются без изменений, ENT_COMPAT — двойные кавычки преобразуются, одиночные остаются без изменений, ENT_QUOTES — одиночные и двойные кавычки преобразуются. Параметр $charset задает кодировку текста. Если четвертый необязательный параметр $double_encode принимает значение FALSE, один раз уже преобразованный HTML-тег не подвергает вторичному преобразованию функции. По умолчанию, этот параметр принимает значение TRUE
htmlspecialchars_decode($str [, $quote_style])	Выполняет обратную функции htmlspecialchars() задачу — преобразует представление HTML-тегов в исходные HTML-теги
htmlentities($str [, $quote_style [, $charset [, $double_encode]]])	Функция аналогична htmlspecialchars(), однако преобразует не минимальное количество символов для представления HTML-кода, а все возможные символы, для которых предусмотрено представление
html_entity_decode($str [, $quote_style [,$charset]])	Выполняет обратную функции htmlentities() задачу — преобразует представление HTML-тегов в исходные HTML-теги
get_html_translation_table ([$table [, $quote_style]])	Возвращает таблицу преобразований, используемую функциями htmlspecialchars() и htmlentities(). Необязательный параметр $table может принимать либо константу HTML_ENTITIES, либо HTML_SPECIALCHARS для htmlspecialchars() и htmlentities(), соответственно. Параметр $quote_style определяет режим замены кавычек
strip_tags($str [, $allowable_tags])	Удаляет из строки $str HTML-теги, кроме тех, которые указываются в параметре $allowable_tags

Экранирование

Группа функций экранирования предназначена для экранирования специальных символов в строках.

Функция	Описание
addslashes($str)	Возвращает копию строки $str, в которой перед каждым специальным символом добавлен обратный слеш (\). Экранируются одиночная кавычка ('), двойная кавычка ("), обратный слеш (\) и NUL (символ \0)
stripslashes($str)	Функция stripslashes() является обратной для addslashes(), предназначена для удаления обратных слешей
addcslashes($str, $charlist)	Возвращает копию строки $str, в которой перед каждым символом из перечисленных в $charlist добавлен обратный слеш (\)
stripcslashes($str)	Функция stripcslashes() удаляет символ обратного слеша и является противоположной функции addcslashes()
quoted_printable_decode($str)	Декодирует строку $str, закодированную методом quoted printable
quotemeta($str)	Возвращает копию строки $str, в которой перед каждым символом . \\ + * ? [^] ($) помещается обратный слеш (\). Функция удобна для экранирования данных в регулярных выражениях

Локальные настройки (локаль)

В разных странах в силу сложившихся традиций имеются различия в представлении чисел, денежных сумм, времени и даты. Правила их представления для каждой страны называют локальными настройками или сокращенно локалью. Каждая операционная система, будь то Windows или одна из UNIX-подобных ОС, поддерживает локальные настройки, которые позволяют изменять представление данных. Такой возможностью обладает и PHP. Ниже представлены функции, ответственные за работу с локалью.

Функция	Описание
Setlocale($category, $locale [, ...])	Устанавливает локальные настройки или возвращает текущую локальную настройку. Функция возвращает значение только что установленной локали или FALSE, если операционная система не поддерживает работу с локальными настройками
localeconv()	Возвращает ассоциативный массив с информацией о число-вых и денежных форматах текущей локали. Ключи массива и их значения описываются в таблице ниже
nl_langinfo($item)	Предоставляет доступ к отдельным элементам локали. Если аргумент $item имеет недопустимое значение, возвращает FALSE.

Локальные настройки управляются шестью переменными окружения, список которых представлен ниже в таблице.

Переменная окружения	Описание
LC_COLLATE	Определяет правила сравнения и, следовательно, сортировки строк для местного алфавита
LC_CTYPE	Определяет правила преобразования одиночных символов для местного алфавита. Позволяет правильно распознавать вид символа: цифра, буква, знак, верхний и нижний регистр
LC_MONETARY	Определяет правила национального представления денежных величи
LC_NUMERIC	Определяет правила национального представления чисел с плавающей точкой
LC_TIME	Определяет правила национального представления даты и времени. Задает именование дней недели, месяцев, формат даты
LC_MESSAGES	Определяет формат информационных, диагностических и интерактивных сообщений операционной системы
LC_ALL	Особая переменная окружения, позволяющая устанавливать значения для всех вышеперечисленных переменных

Функция localeconv() возвращает ассоциативный массив с информацией о числовых и денежных форматах. Структура результирующего массива описывается таблице ниже.

Ключ	Описание
decimal_point	Символ десятичной точки
thousands_sep	Разделитель групп (тысячи)
grouping	Массив, содержащий количество цифр в группах для числовых данных
int_curr_symbol	Международное обозначение валюты, например, RUR
currency_symbol	Национальное обозначение валюты, например, руб
mon_decimal_point	Символ десятичной точки в денежном формате
mon_thousands_sep	Разделитель групп в денежном формате
mon_grouping	Массив, содержащий количества цифр в группах для денежных данных
positive_sign	Знак для положительных чисел

negative_sign	Знак для отрицательных чисел
int_frac_digits	Количество разрядов после точки (международное)
frac_digits	Количество разрядов после точки (национальное)
p_cs_precedes	TRUE, если currency_symbol записывается перед положительным значением, иначе FALSE
p_sep_by_space	TRUE, если currency_symbol отделяется от положительного значения пробелом, иначе FALSE
n_cs_precedes	TRUE, если currency_symbol записывается перед отрицательным значением, иначе FALSE
n_sep_by_space	TRUE, если currency_symbol отделяется от отрицательного значения пробелом, иначе FALSE
p_sign_posn	Для положительных чисел: 0 — число и обозначение валюты заключаются в скобки; 1 — знак записывается перед числом и обозначением валюты; 2 — знак записывается после числа и обозначения валюты; 3 — знак записывается перед обозначением валюты; 4 — знак записывается после обозначения валюты
n_sign_posn	Для отрицательных чисел: 0 — число и обозначение валюты заключаются в скобки; 1 — знак записывается перед числом и обозначением валюты; 2 — знак записывается после числа и обозначения валюты; 3 — знак записывается перед обозначением валюты; 4 — знак записывается после обозначения валюты

Форматный вывод

Функции данной группы осуществляют вывод информации в окно браузера и ее форматирование.

Функция	Описание
echo($arg1 [, ...])	Выводит аргумент $arg1 и все последующие в окно браузера
print($arg)	Выводит аргумент $arg в окно браузера
printf($format [, $args [, ...]])	Выводит аргументы $args в окно браузера, отформатированные в соответствии со строкой $format
fprintf($handle, $format [, $args [, ...]])	Выводит аргументы $args в открытый файл $handle, отформатированные в соответствии со строкой $format
sprintf($format [, $args [, ...]])	Возвращает строку, состоящую из аргументов $args, отформатированных в соответствии со строкой $format

vprintf($format, $args)	Выводит элементы массива $args в окно браузера, отформатированные в соответствии со строкой $format
vfprintf($handle, $format, $args)	Выводит элементы $args в открытый файл $handle, отформатированные в соответствии со строкой $format
vsprintf($format, $args)	Возвращает строку, состоящую из элементов массива $args, отформатированных в соответствии со строкой $format
number_format($number [, $decimals [, $dec_point, $thousands_sep]])	Возвращает отформатированное число $number с $decimals знаками после запятой. При этом в качестве десятичной точки будет использован символ $dec_point, а тысячи будет разделять символ $thousands_sep
money_format($format, $number)	Возвращает отформатированную денежную сумму $number в соответствии со строкой форматирования $format и текущей локалью

Хранение данных

Строки очень часто используются для хранения разнообразных данных. Поэтому PHP имеет целый арсенал функций для упаковки данных в строки и извлечению их.

Функция	Описание
bin2hex($str)	Осуществляет побайтовое преобразование символов строки $str в шестнадцатеричный формат
pack($format[, $args [, ...]])	Упаковывает параметры $args в двоичную строку. Формат параметров и их количество определяется первым параметром $format
unpack($format, $data)	Осуществляет обратное функции pack() преобразование
serialize($value)	Осуществляет упаковку массива или объекта в строку $value
unserialize($str)	Осуществляет распаковку массива или объекта из строки $str, созданной функцией serialize()

Работа с путями к файлам и каталогам

Функции данной группы позволяют облегчить работу с путями к файлам и каталогам.

Функция	Описание
pathinfo($path [, $options])	Принимает путь к файлу $path и возвращает ассоциативный

	массив, в элементах которого сохраняются каталог, в котором расположен файл, имя файла, его расширение и имя файла без расширения. Если указан необязательный параметр $options, функция возвращает строковое значение. Параметр $options может принимать следующие константы: • PATHINFO_DIRNAME — путь к файлу (каталог); • PATHINFO_BASENAME — имя файла; • PATHINFO_EXTENSION — расширение файла; • PATHINFO_FILENAME — имя файла без расширения
realpath($path)	Возвращает абсолютный путь (путь от начала диска) для файла $path
basename($path [, $suffix])	Извлекает из пути $path имя файла. Необязательный параметр $suffix позволяет исключить расширение из результирующей строки
dirname($path)	Извлекает из пути $path каталог, в котором расположен файл

Объединение и разбиение строк

Функции данной группы осуществляют объединение подстрок в единую строку, а также разбиение строки на отдельные подстроки.

Функция	Описание
str_repeat($str, $number)	Создает новую строку, состоящую из $number повторов строки $str
str_pad($str, $length [, $pad [, $pad_type]])	Увеличивает размер строки $str до $length символов. По умолчанию новые символы заполняются пробелами, однако символзаполнитель можно задать в необязательном параметре $pad. Необязательный параметр $pad_type определяет, с какой стороны добавляются новые символы, и может принимать следующие значения: • STR_PAD_RIGHT — новые символы добавляются в конец строки (по умолчанию); • STR_PAD_LEFT — новые символы добавляются в начало строки; • STR_PAD_BOTH — новые символы добавляются как в конец, так и в начало;
chunk_split($str [, $chunklen [, $end]])	Возвращает копию строки $str, в которой через каждые $chunklen (по умолчанию 76) символов вставляется подстрока $end (по умолчанию \r\n
str_split($str [, $split])	Преобразует строку $str в массив. По умолчанию в качестве элементов массива выступают символы строки.

	Необязательный параметр $split позволяет задать произвольную длину подстрок в символах, образующих элементы массива
strtok($str, $token)	Разбивает строку $str на подстроки, используя в качестве разделителя подстрок последовательность $token
str_word_count($str [, $format [, $charlist]])	Если указывается единственный параметр $str, возвращается количество слов в строке. Если указывается необязательный параметр $format, отдельные слова строки возвращаются в виде массива. Параметр $format может принимать следующие значения: 1 — возвращается массив, содержащий все слова, входящие в строку $str; 2 — возвращается массив, индексами которого являются позиции в строке $str, а значениями — соответствующие слова Необязательный параметр $charlist позволяет задать дополнительные символы, из которых будет состоять слово
explode($delimiter, $str [, $limit])	Функция возвращает массив из строк, каждая из которых соответствует фрагменту исходной строки $str, находящемуся между разделителями, определяемыми аргументом $delimiter. Необязательный параметр $limit определяет максимальное количество элементов в массиве
implode($delimiter, $arr)	Объединяет элементы массива строк $arr в единую строку, разделяя элементы подстрокой, заданной в параметре $delimiter
join()	Синоним для функции implode()
wordwrap($str [, $width [, $break [, $cut]]])	Вставляет в строке $str символы $break (по умолчанию \n) через каждые $width символов (по умолчанию 75). При этом символ $break не разрывает слово, однако если необязательный параметр $cut принимает значение TRUE, допускается разрыв слова
parse_str($str [, &$arr])	Разбивает строку $str, имеющую формат строки с GET-запросом, и преобразует отдельные GET-параметры в переменные или в элементы массива $arr (если указан второй параметр
sscanf($str, $format [, ...])	Разбирает строку $str в соответствии со строкой форматирования $format. Если передано только два аргумента, функция возвращает массив, в противном случае результаты помещаются в последующие параметры

Функции даты и времени

Исторически сложилось, что время и дата имеют собственную достаточно запутанную систему счисления, которая значительно отличается от десятичной, используемой в математических и компьютерных вычислениях. Чтобы совместить эти две системы счисления, в компьютерных вычислениях время хранят в UNIXSTAMP-формате, при необходимости форматируя дату для отображения в традиционном формате.

В UNIXSTAMP-формате время хранится в секундах, прошедших с 0:00 1 января 1970 года. Время в таком формате позволяет легко сравнивать даты друг с другом, а также прибавлять и вычитать интервалы времени. Функции, формирующие дату в этом формате, представлены.

Формирование даты и времени

Функция	Описание
time()	Возвращает текущую дату и время в UNIXSTAMP-формате
microtime([$get_as_float])	Возвращает текущую дату и время в UNIXSTAMP-формате с микросекундами. Если параметр $get_as_float принимает значение TRUE, функция возвращает результат в виде числа с плавающей точкой
mktime([$hour [, $minute [, $second [, $month [, $day [, $year [, $is_dst]]]]]]])	Возвращает дату и время в UNIXSTAMP-формате с годом $year, месяцем $month, днем месяца $day, часом $hour, минутами $minute, секундами $second. Если параметр $id_dst принимает значение 1, считается, что установлено летнее время, если 0 — зимнее, если –1 или параметр не указан вовсе, PHP пытается автоматически определить его
gmmktime([$hour [, $minute [, $second [, $month [, $day [, $year [, $is_dst]]]]]]])	Функция gmmktime() аналогична mktime(), однако возвращается не местное время, а время по Гринвичу
gettimeofday([$return_float])	Возвращает текущую дату и время либо в виде ассоциативного массива, либо, если параметр $return_float принимает значение TRUE, в виде числа с плавающей точкой
strtotime($time [, $timestamp])	Разбирает строку с датой и временем и возвращает их в формате UNIXSTAMP

Форматирование даты и времени

Функция	Описание
date($format [, $timestamp])	Форматирует текущую дату в соответствии со строкой, переданной в качестве первого параметра $format. Может отформатировать произвольную дату, если она передана в качестве второго необязательного параметра $timestamp
gmdate($format [, $timestamp])	Функция gmdate() аналогична функции date(), однако возвращает не текущее локальное время, а время по Гринвичу
idate($format [, $timestamp])	Форматирует текущую дату в соответствии со строкой, переданной в качестве первого параметра $format. Может отформатировать произвольную дату, если она передана в качестве второго необязательного параметра $timestamp. В отличие от функции date() возвращает лишь целые значения
checkdate($month, $day, $year)	Проверяет, является ли дата с годом $year, месяцем $month и днем $day корректной, и возвращает TRUE, если это так, и FALSE в противном случае
getdate([$timestamp])	Возвращает текущую дату и время в виде ассоциативного массива. Через необязательный параметр $timestamp можно передать произвольную временную метку
strftime($format [, $timestamp])	Форматирует текущие дату и время согласно локальным настройкам. Параметр $format позволяет задать строку форматирования. Необязательный параметр $timestamp позволяет передать на обработку функции произвольную временную метку
gmstrftime($format [, $timestamp])	Функция gmstrftime() аналогична функции strftime(), однако возвращает не текущее локальное время, а время по Гринвичу
strptime($date, $format)	Разбирает строку, сгенерированную функцией strftime(), и возвращает дату и время в виде ассоциативного массива
date_parse($date)	Разбирает строку, сгенерированную функцией strftime(), и возвращает дату и время в виде ассоциативного массива
localtime([$timestamp [, $is_associative]])	Возвращает текущую дату и время в виде индексного массива (или ассоциативного массива, если параметр $is_associative принимает значение TRUE). Массив может быть возвращен не только для текущего времени, но и для произвольной временной метки в формате UNIXSTAMP,

	которая может быть передана через параметр $timestamp

Математические функции

Поиск максимума и минимума

Функция	Описание
max($value1, [$value2 [, $value3...]])	Возвращает максимальное значение среди параметров
min($value1, [$value2 [, $value3...]])	Возвращает минимальное значение среди параметров

Генерация случайных чисел

Функция	Описание
rand([$min, $max])	Генерирует случайное число в интервале от $min до $max. Если параметры $min и $max отсутствуют, возвращается число, лежащее в пределах от 0 до RAND_MAX (32 768)
mt_rand([$min, $max])	Более быстрая версия функции rand(), обладающая более равномерным распределением случайных значений (что важно с точки зрения криптографии)
lcg_value()	Генерирует и возвращает случайное дробное число, лежащее в диапазоне от 0 до 1

Преобразование значений между разными системами счисления

Функция	Описание
base_convert($number, $frombase, $tobase)	Переводит число $number из системы счисления по основанию $frombase в систему по основанию $tobase. $frombase и $tobase принимают значения только от 2 до 36 включительно. В строке $number цифры обозначают сами себя, символ a соответствует числу 11, b — числу 12 и т. д. Символ z обозначает число 36
decbin($number)	Преобразует десятичное число $number в двоичное
decoct($number)	Преобразует десятичное число $number в восьмеричное

dechex($number)	Преобразует десятичное число $number в шестнадцатеричное
bindec($binary)	Преобразует двоичное число $binary в десятичное
hexdec($hex)	Преобразует шестнадцатеричное число $hex в десятичное
octdec($octal)	Преобразует восьмеричное число $octal в десятичное
deg2rad($number)	Преобразует градусы $number в радианы
rad2deg($number)	Преобразует радианы $number в градусы

Округление чисел

Функция	Описание
abs($number)	Возвращает модуль числа *$number*
round($value [, $precision])	Округляет число $value, если указан необязательный параметр $precision — округление осуществляется до $precision знака после запятой
ceil($value)	Округляет число $value до ближайшего целого числа, причем результат всегда больше $value
floor($value	Округляет число $value до ближайшего целого числа, причем результат всегда меньше $value
fmod($x, $y)	Возвращает остаток деления двух чисел $x и $y

Логарифмические и степенные функции

Функция	Описание
pow($base, $exp)	Возвращает число $base в степени $exp
exp($number)	Возвращает экспоненту в степени $number
expm1($number)	Возвращает экспоненту в степени $number минус единица
sqrt($number)	Возвращает корень квадратный числа $number
log($number [, $base])	Возвращает логарифм числа $number по основанию $base, если параметр $base не указан — возвращается натуральный логарифм

log10($number)	Возвращает десятичный логарифм числа $number
log1p($number)	Возвращает натуральный логарифм от суммы $number плюс единица

Работа с файлами

Создание файлов

PHP предоставляет разработчикам большое количество функций для работы с файлами. Первыми рассмотрим функции, позволяющие создать файлы.

Функция	Описание
fopen($filename, $mode [, $use_include_path[, $context]])	Открывает файл с именем $filename в режиме $mode. Некоторые режимы позволяют создать файл, некоторые требуют, чтобы файл уже существовал. Необязательный параметр $use_include_path позволяет задать путь для поиска файла. В случае успеха функция возвращает дескриптор открытого файла, в случае неудачи — FALSE. Параметр $context позволяет задать HTTP-заголовки, отправляемые сервером при обращении к сетевому файлу.
fclose($handle)	Закрывает файл, открытый ранее при помощи функции fopen(). В качестве единственного параметра $handler функция принимает дескриптор открытого файла. Возвращает TRUE при успешном закрытии файла и FALSE в противном случае
touch($filename [, $time [, $atime]])	Функция touch() предназначена для изменения времени последней модификации файла $time и времени последнего доступа $atime. Если файл не существует — он создается, поэтому функция часто используется для создания файлов
tempnam($dir, $prefix)	Создает в каталоге $dir файл с уникальным именем, при этом в имени файла используется префикс $prefix. В случае успеха возвращает имя нового файла, в случае неудачи — FALSE
tmpfile()	Создает временный файл с уникальным именем и возвращает его дескриптор. Файл автоматически удаляется после его закрытия. Если создать файл не удалось, функция возвращает FALSE

Манипуляция файлами

Помимо операции создания файлов очень часто возникают задачи, связанные с их перемещением, переименованием и удалением. Функции, ответственные за эти операции, представлены в таблице.

Функция	Описание
copy($source, $destination)	Копирует файл с именем $source в файл с именем $destination. В случае успешного копирования функция возвращает TRUE, в противном случае возвращает FALSE
unlink($filename)	Удаляет файл с именем $filename. В случае успешного удаления функция возвращает TRUE, в противном случае возвращает FALSE
rename($oldname, $newname)	Переименовывает файл с именем $oldname, назначая ему новое имя $newname. В случае успешного переименования функция возвращает TRUE, в противном случае возвращает FALSE
is_uploaded_file($filename)	Возвращает TRUE, если файл был загружен на сервер, и FALSE в противном случае. В качестве аргумента функция принимает элемент массива $_FILES['filename']['tmp_name'], содержащий загруженный файл во временном каталоге
move_uploaded_file ($filename, $destination)	Перемещает файл из временного каталога в каталог назначения. В качестве первого аргумента $filename зачастую используется элемент $_FILES['filename']['tmp_name'], второй аргумент может быть произвольным. Если необходимо сохранить исходное имя файла, можно воспользоваться элементом $_FILES['filename']['name']. В отличие от функции copy() функция move_uploaded_file() оперирует лишь загруженными файлами

Чтение и запись файлов

Данные функции раздела позволяют оперировать содержимым файла: читать, записывать и перезаписывать.

Функция	Описание
fopen($filename, $mode [, $use_include_path [, $context]])	Открывает файл с именем $filename в режиме $mode. Некоторые режимы позволяют создать файл, некоторые требуют, чтобы файл уже существовал. Необязательный параметр $use_include_path позволяет задать путь для поиска файла. В случае успеха функция возвращает дескриптор открытого файла, в случае неудачи — FALSE. Параметр $context позволяет задать HTTP-заголовки, отправляемые сервером при обращении к сетевому файлу

fclose($handle)	Закрывает файл, открытый ранее при помощи функции fopen(). В качестве единственного параметра $handle функция принимает дескриптор открытого файла. Возвращает TRUE при успешном закрытии файла и FALSE в противном случае
feof($handle)	Возвращает TRUE, если файловый указатель $handler указывает на конец файла, и FALSE в противном случае
fgetc($handle)	Считывает и возвращает в качестве результата из открытого файла $handle один символ
fgets($handle [, $length])	Считывает и возвращает в качестве результата из открытого файла $handle строку. Чтение заканчивается по достижении строки длины $handle – 1 (по умолчанию $handle принимает значение 1000), по достижении символа перевода строки или символа конца файла
fgetss($handle [, $length [, $allowable_tags]])	Аналогична функции fgets(), однако из прочитанной строки удаляются все HTML-теги. Необязательный параметр $allowable_tags может содержать допустимые HTML-теги, которые не должны отбрасываться
fread($handle, $length)	Считывает и возвращает в качестве результата из открытого файла $handle строку длиной $length символов. В отличие от функции fgets() переводы строк не прекращают чтение
fscanf($handle, $format [, ...])	Аналогична функции форматного разбора строки sscanf(), однако чтение и разбор строки по форматному правилу $format осуществляются не из строки, а из открытого файла $handler. Если в функцию переданы только два первых параметра, обработанные значения будут возвращены в виде массива. В противном случае, если были переданы необязательные аргументы, функция вернет количество присвоенных значений
fpassthru($handle)	Читает содержимое открытого файла $handle от текущей позиции до конца файла и выводит в окно браузера
file_get_contents($filename [, $flags [, $context [, $offset [, $maxlen]]]])	Читает файл с именем $filename в режиме $flags и возвращает его содержимое в виде строки. Необязательный параметр $context позволяет задать HTTP-заголовки, отправляемые сервером при обращении к сетевому файлу. Если это не требуется, вместо него можно передать NULL. Параметр $offset задает позицию, начиная с которой выполняется чтение, а $maxlen — количество символов, которые

	следует прочитать
file($filename [, $flags [, $context]])	Читает файл с именем $filename и возвращает его содержимое в виде массива, каждый элемент которого соответствует отдельной строке. Необязательный параметр $flags позволяет задать режим чтения файла. Параметр $context позволяет задать HTTP-заголовки, отправляемые сервером при обращении к сетевому файлу
readfile($filename [, $use_include_path [, $con- text]])	Читает файл с именем $filename и выводит его содержимое в окно браузера. Необязательный параметр $use_include_path позволяет задать путь для поиска файла. Параметр $context позволяет задать HTTP-заголовки, отправляемые сервером при обращении к сетевому файлу
fwrite($handle, $str [, $length])	Записывает в открытый файл $handle содержимое строки $str. Необязательный параметр $length позволяет задать количество байтов, предназначенных для записи (по достижении этого количества запись останавливается). В случае успеха функция возвращает количество записанных байтов, в случае неудачи — FALSE
fputs()	Синоним для функции fwrite()
file_put_contents($filename, $data [, $flags [, $context]])	Записывает в файл с именем $filename данные из параметра $data (либо строка, либо одномерный массив). Необязательный параметр $flags определяет режим записи файла. Параметр $context позволяет задать HTTP-заголовки, отправляемые сервером при обращении к сетевому файлу
fflush($handle)	Сбрасывает информацию, предназначенную для записи в файл из оперативной памяти на жесткий диск. Использование этой функции предотвращает потерю записанной в файл информации при нештатном завершении скрипта
flock($handle, $operation [, $wouldblock])	Устанавливает для открытого файла с дескриптором $handle режим блокировки $operation. Необязательный параметр $wouldblock позволяет выяснить успешность операции
ftruncate($handle, $size)	Уменьшает размер файла с дескриптором $handle до $size байтов. Возвращает TRUE в случае успеха и FALSE в противном случае

Права доступа к файлу

Функция	Описание
chmod($filename, $mode)	Устанавливает права доступа $mode для файла $filename. Возвращает TRUE в случае успеха и FALSE — в случае неудачи
chgrp($filename, $group)	Устанавливает группу владельца $group (имя группы или уникальный идентификатор GID) для файла $filename. Возвращает TRUE в случае успеха и FALSE — в случае неудачи
chown($filename, $user)	Устанавливает владельца $user (имя владельца или уникальный идентификатор UID) для файла $filename. Возвращает TRUE в случае успеха и FALSE — в случае неудачи
lchgrp($link, $group)	Осуществляет попытку установить группу владельца $group (имя группы или уникальный идентификатор GID) для символической ссылки $link. Возвращает TRUE в случае успеха и FALSE — в случае неудачи
lchown($link, $user)	Осуществляет попытку установить владельца $user (имя владельца или уникальный идентификатор GID) для символической ссылки $link. Возвращает TRUE в случае успеха и FALSE — в случае неудачи
fileperms($filename)	Возвращает права доступа файла $filename. В случае неудачи возвращает FALSE
filegroup($filename)	Возвращает уникальный идентификатор группы владельца GID. В случае неудачи возвращает FALSE
fileowner($filename)	Возвращает уникальный идентификатор владельца UID файла $filename. В случае неудачи возвращает FALSE
umask([$mask])	Устанавливает права доступа по умолчанию для вновь создаваемых файлов и каталогов
is_readable($filename)	Возвращает TRUE, если файл $filename доступен для чтения, в противном случае возвращает FALSE
is_writable($filename)	Возвращает TRUE, если файл $filename доступен для записи, в противном случае возвращает FALSE
is_writeable()	Синоним функции is_writable()
is_executable($filename)	Возвращает TRUE, если файл $filename доступен для выполнения, в противном случае возвращает FALSE

Работа с каталогами

Функция	Описание

mkdir($pathname [, $mode [, $recursive [, $context]]])	Создает каталог $pathname с правами доступа $mode. Если необязательный параметр $recursive принимает значение TRUE, все несуществующие каталоги в пути $pathname создаются. Параметр $context позволяет задать заголовки, отправляемые сервером при обращении к сетевому файлу. При успешном создании каталога функция возвращает TRUE, в противном случае — FALSE
rmdir($dirname [, $context])	Удаляет пустой каталог с именем $dirname. При успешном удалении каталога функция возвращает TRUE, в противном случае — FALSE
getcwd()	Возвращает полный путь к текущему каталогу
chdir($directory)	Изменяет текущий каталог на новый, указанный в параметре $directory. При успешной смене текущего каталога функция возвращает TRUE, в противном случае — FALSE
chroot($directory)	Изменяет корневой каталог текущего процесса на $directory. При успешной смене корневого каталога возвращает TRUE, в противном случае — FALSE
opendir($path [, $context])	"Открывает" каталог — функция возвращает дескриптор, позволяющий читать содержимое каталога $path
readdir($dir_handle)	Читает одну позицию из каталога и передвигает дескриптор на одну позицию вперед. Последовательный вызов функции позволяет прочитать содержимое каталога файл за файлом. Функция принимает в качестве единственного параметра $dir_handle дескриптор, полученный ранее при помощи функции opendir(). Если достигнут конец каталога, функция возвращает FALSE
closedir($dir_handle)	Закрывает дескриптор $dir_handle, открытый ранее при помощи функции closedir()
rewinddir($dir_handle)	Помещает дескриптор $dir_handle в начало каталога
scandir($directory [, $sorting_order [, $context]])	Возвращает массив с содержимым каталога $directory или FALSE, если $directory не является каталогом. Содержимое каталога сортируется в алфавитном порядке, если необязательный параметр $sorting_order принимает значение TRUE, элементы результирующего массива сортируются в обратном алфавитном порядке
glob($pattern [, $flags])	Возвращает массив с именами файлов и подкаталогов, путь к которым удовлетворяет шаблону $pattern. Необязательный параметр $flags позволяет задать режим интерпретации шаблона