

Функции для работы с файловой системой

Файловая система

`string basename (string path)`

Возвращает имя файла, выделенного из пути к файлу.

`string dirname (string path)`

Возвращает имя родительского каталога из указанного пути.

`string realpath (string path)`

Возвращает канонизированный абсолютный путь к файлу.

`string getcwd (void)`

Возвращает имя текущего рабочего каталога.

`string chdir (string directory)`

Изменяет текущий каталог на указанный в аргументе `directory`.

Как проверить существует ли файл?

`bool file_exists (string filename)`

Проверка существования файла.

`bool is_readable (string filename)`

Проверка существования файла, доступного для чтения.

`bool is_writable (string filename)`

Проверка существования файла, доступного для записи.

Как проверить существует ли файл определенного типа?

```
bool is_file ( string filename )
```

Определяет, является ли файл обычным файлом.

```
bool is_dir ( string filename )
```

Определяет, является ли имя файла директорией.

```
bool is_executable ( string filename )
```

Определяет, является ли файл исполняемым.

```
bool is_link ( string filename )
```

Определяет, является ли файл символической ссылкой.

Как манипулировать файлами?

`bool touch (string filename [, int time [, int atime]])`

Устанавливает время доступа и модификации файла.

`bool copy (string source, string dest)`

Копирует файл `source` в файл с именем `dest`.

`bool rename (string oldname, string newname)`

Переименовывает файл `oldname` в `newname`.

`bool unlink (string filename)`

Удаляет файл `filename`.

`bool mkdir (string path [, int mode [, bool recursive]])`

Создает новую директорию `path` с атрибутами доступа `mode`.

`bool rmdir (string dirname)`

Функция пытается удалить директорию `dirname`.

Как работать с директориями?

```
resource opendir ( string path [, resource context ] )
```

Возвращает дескриптор открытой директории `path`.

```
void closedir ( [ resource dir_handle ] )
```

Закрытие открытого дескриптора директории.

```
string|bool readdir ( [ resource dir_handle ] )
```

Возвращает имя следующего по порядку элемента каталога.

```
void rewinddir ( [ resource dir_handle ] )
```

Реинициализация дескриптора директории.

Как открыть файл или URL?

```
resource fopen (  
    string filename, string mode  
    [, bool use_include_path [, resource context ]]  
)
```

Функция открывает указанный аргументом `filename` поток ввода-вывода (файл) в режиме `mode` и возвращает соответствующий дескриптор.

```
$fp = fopen('/home/rasmus/file.txt', 'r');
```

```
$fp = fopen('/home/rasmus/file.gif', 'wb');
```

```
$fp = fopen('http://php.net', 'r');
```

```
$fp = fopen('ftp://user:password@example.com/', 'w');
```

```
$fp = fopen('c:\\data\\info.txt', 'r');
```

Как открыть файл для записи?

'w', 'a', 'x', 'c' - Открывает файл только для записи.

Если файл не существует - пытается его создать.

'w', 'x', 'c' - Помещает указатель в начало файла.

'a' - Помещает указатель в конец файла.

'w' - Обрезает файл до нулевой длины.

'a', 'c' - Если файл существует, то он не обрезается.

'x' - Если файл существует возвращает `false` и выдаст ошибку уровня `E_WARNING`.

Как записать в файл?

`int fwrite (resource handle, string str [, int length])`

Бинарно-безопасная запись в файл.

Записывает содержимое `str` в файловый поток `handle`.

Запись прекратится когда `length` байтов будут записаны или будет достигнут конец строки `str`.

`int fputs (resource handle, string str [, int length])`

Синоним `fwrite()`.

`bool ftruncate (resource handle, int size)`

Урезает файл до указанной длины `size`.

`int filesize (string filename)`

Функция возвращает размер файла.

Как открыть файл для чтения?

'r' - Открывает файл только для чтения.

Помещает указатель в начало файла.

'r+' - Открывает файл для чтения и записи.

Помещает указатель в начало файла.

'w+', 'a+', 'x+', 'c+' - эти режимы имеют аналогичное поведение, как 'w', 'a', 'x', 'c', но добавляют возможность чтения данных из файла.

Как читать из файла?

`string fgetc (resource handle)`

Считывает символ из переданного указателя на файл.

`string fgets (resource handle [, int length])`

Читает строку из файла.

Чтение заканчивается по достижении `length - 1`.

`string fread (resource handle [, int length])`

Бинарно-безопасное чтение файла.

Читает до `length` байт из файлового указателя.

`bool feof (resource handle)`

Функция возвращает `TRUE`, если указатель файла указывает на EOF или произошла ошибка.

Как закрыть файл?

```
bool fclose ( resource handle )
```

Функция закрывает файл, на который указывает дескриптор `handle`.

```
$fp = fopen('/home/rasmus/file.txt', 'r');
```

```
fclose($fp);
```

Как прочитать файл в массив?

```
bool file ( string filename [, int flags [, resource context ]])
```

Читает содержимое файла и помещает его в массив.

Аргумент **flags** это битовая маска, состоящая из констант:

- **FILE_USE_INCLUDE_PATH**
- **FILE_IGNORE_NEW_LINES**
- **FILE_SKIP_EMPTY_LINES.**

Например: **FILE_IGNORE_NEW_LINES | FILE_SKIP_EMPTY_LINES**

Как прочитать файл в строку?

```
string file_get_contents (  
    string filename [, bool use_include_path  
    [, resource context [, int offset [, int maxlen ]]]]  
)
```

Читает содержимое файла в строку,
начиная с указанного смещения `offset` и до `maxlen` байт.

Как записать строку в файл?

```
string file_put_contents (  
    string filename, mixed data [, int flags [, resource context]]  
)
```

Пишет строку в файл.

Аргумент **flags** это битовая маска, состоящая из констант:

- **FILE_USE_INCLUDE_PATH**
- **FILE_APPEND**
- **LOCK_EX.**

Загрузка файлов на сервер методом HTTP POST

HTML-форма

```
<form enctype="multipart/form-data" action="#" method="post">  
  <input type="hidden" name="MAX_FILE_SIZE" value="30000">  
  Отправить этот файл: <input name="userfile" type="file">  
  <input type="submit" value="Send File">  
</form>
```

Обязателен атрибут `enctype="multipart/form-data"`, в противном случае загрузка файлов на сервер выполняться не будет.

`MAX_FILE_SIZE` максимально допустимый размер загружаемого файла в байтах.

Не заставляет пользователя ждать ответа от сервера, если размер выбранного файла больше допустимого.

Как переместить загруженный файл?

```
bool move_uploaded_file ( string filename, string destination )
```

Проверяет является ли файл загруженным на сервер методом POST.

Если это так, перемещает его в указанное место.

PHP способен получать загруженные файлы из любого браузера, совместимого со стандартом RFC-1867.

php.ini

`file_uploads` **boolean** или **integer**

Разрешить или запретить загрузку файлов.

`upload_max_filesize` **integer**

Максимальный размер загружаемого файла.

`post_max_size` **integer**

Максимально допустимый размер данных, отправляемых методом POST.

`max_file_uploads` **integer**

Максимальное количество одновременно загружаемых файлов.

`max_input_time` **integer**

Максимальное время в секундах, за которое скрипт должен разобрать все входные данные, переданные запросами вроде POST или GET.

`upload_tmp_dir` **string**

Временная директория для хранения файлов во время загрузки.

Суперглобальный массив `$_FILES`

`$_FILES` содержит всю информацию о загруженных файлах.

- **`name`** - оригинальное имя файла на компьютере клиента;
- **`type`** - mime-тип файла;
- **`tmp_name`** - временное имя, с которым принятый файл был сохранен;
- **`size`** - размер в байтах принятого файла;
- **`error`** - код ошибки, которая может возникнуть при загрузке.