Урок №4

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.

Проверка существования файла

bool file_exists (string filename)

Проверка существования файла.

bool is_dir (string filename)

Определяет, является ли имя файла директорией.

bool is_executable (string filename)

Определяет, является ли файл исполняемым.

bool is_file (string filename)

Определяет, является ли файл обычным файлом.

bool is_link (string filename)

Определяет, является ли файл символической ссылкой.

bool is_readable (string filename)

Проверка существования файла, доступного для чтения.

bool is_writable (string filename)

Проверка существования файла, доступного для записи.

Манипулирование файлами

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

Устанавливает время доступа и модицикации файла.

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.

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.

Работа с директориями

resource opendir (string path [, resource context])

Возвращает дескриптор открытой директории path.

void closedir ([resource dir_handle])

Закрытие открытого дескриптора директории.

bool readdir ([resource dir_handle])

Возвращает имя следующего по порядку элемента каталога.

void rewinddir ([resource dir_handle])

Реинициализация дескриптора директории.

Пример

```php
# Открыть известный каталог и начать считывать его содержимое

$dir = '/etc/php5/';

if (is_dir($dir)) {
  if ($dh = opendir($dir)) {
    while ( ($file = readdir($dh)) !== false) {
      echo "файл: $file, тип: " . filetype($dir . $file) . "\n";
    }

    closedir($dh);
  }
}
```

Задача №4.1

Написать функцию, которая будет удалять каталог и всё содержимое в нём, т.е. подкаталоги и файлы.

Осуществить рекурсивный вызов этой функции в подкаталогах.

Исходные данные: path – путь удаляемого каталога.

Урок №4

Ввод и вывод

Открытие файла 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.

```php
$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');
```

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

Функция закрывает файл, на который указывает дескриптор handle.

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

Режимы открытия файлов

'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 - пытается его создать.

'a' - Открывает файл только для записи.

Помещает указатель в конец файла.

Если файл не существует - пытается его создать.

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

Режимы открытия файлов

'x' - Создаёт и открывает файл только для записи.

Помещает указатель в начало файла.

Если файл не существует, пытается его создать.

Если файл уже существует возвращает FALSE и выдаст ошибку уровня E_WARNING.

'c' - Открывает файл только для записи.

Помещает указатель в начало файла.

Если файл не существует, то он создается.

Если файл существует, то он не обрезается.

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

Чтение 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.

Запись 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 длинны size.

int filesize (string filename)

Функция возвращает размер файла.

Примеры

```php
$filename = '/usr/local/something.txt';
$fp = fopen($filename, 'r');
$contents = fread($fp, filesize($filename));
fclose($fp);
```

```php
$fp = @fopen('/tmp/inputfile.txt', 'r');

if ($fp) {
  while (!feof($fp)) {
    $buffer = fgets($fp, 4096);
    echo $buffer;
  }

  fclose($fp);
}
```

```php
$fp = fopen('data.txt', 'w');
fwrite($fp, '1');
fwrite($fp, '23');
fclose($fp);
```

Позиция курсора

int ftell (resource handle)

Возвращает позицию файлового указателя handle - смещение в файловом потоке.

string fseek (resource handle, int offset [, int whence])

Установка позиции курсора файла.

Смещение измеряется в байтах от начала файла путём прибавления параметра offset к позиции, указанной в параметре whence.

SEEK_SET - смещение в offset байт (по-умолчанию).

SEEK_CUR - смещение в текущее положение плюс offset.

SEEK_END - смещение в конец файла плюс offset.

bool rewind (resource handle)

Установка курсора на начало файлового потока.

Эквивалентно fseek(fp, 0).

Для справки

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

Читает содержимое файла и помещает его в массив.

flags: FILE_USE_INCLUDE_PATH, FILE_IGNORE_NEW_LINES,

FILE_SKIP_EMPTY_LINES.

```
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.

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

Пишет строку в файл.

flags: FILE_USE_INCLUDE_PATH, FILE_APPEND, LOCK_EX.

Задача №4

Создать HTML-форму с одним текстовым полем и кнопкой Submit.

Пользователь вводит в форму URL-адрес (гиперссылку).

Написать обработчик, который проверяет наличие такой же ссылки в файле и если не находит совпадений, то записывает ее в конец файла.

Урок №4

Загрузка файлов на сервер методом HTTP POST

HTML-форма

```
<form enctype="multipart/form-data" action="__URL__" 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.

Предотвращает ожидание пользователей при передаче больших файлов.

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

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.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.

```
Array (
    [userfile] => Array (
        [name] => jshtm.zip
        [type] => application/x-zip-compressed
        [tmp_name] => /tmp/phpAE.tmp
        [size] => 21344
        [error] => 0
    )
)
```

- 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.

Коды ошибок

UPLOAD_ERR_OK
Загрузка выполнена успешно.

UPLOAD_ERR_INI_SIZE
Размер загружаемого файла превысил upload_max_filesize.

UPLOAD_ERR_FORM_SIZE
Размер загружаемого файла превысил MAX_FILE_SIZE.

UPLOAD_ERR_PARTIAL
Файл был загружен лишь частично.

UPLOAD_ERR_NO_FILE
Файл не был загружен.

UPLOAD_ERR_NO_TMP_DIR
Отсутствует временная папка.

UPLOAD_ERR_CANT_WRITE
Не удалось записать файл на диск.

UPLOAD_ERR_EXTENSION
PHP-расширение остановило загрузку файлов.

Урок №4

Директивы require и include

Множественное включение

Директивы require и include заменяются интерпретатором в ходе выполнения на содержимое файла, имя которого указывается в качестве параметра этих директив.[1]

```
include <имя_файла>
require <имя_файла>
```

При использовании директивы include, если указанный файл не найден, то выдается предупреждение, а сценарий продолжает свою работу.

При использовании директивы require, если указанный файл не найден, то выполнение сценария завершится критической ошибкой.

Можно использовать файлы из сети, указав вместо имени URL, если это разрешено в настройках php (см. allow_url_include в php.ini).

Подключаемый файл интерпретируется как HTML-файл.

1. Аналогией этих директив является директива препроцессора языка C #include.

Однократное включение

Директивы require_once и include_once не допускают повторного использования файла, если его включение уже происходило ранее.

```
include_once <имя_файла>
require_once <имя_файла>
```

```php
#включение с указанием относительного пути
include 'config.php';
include_once 'config.php';

#включение с указанием абсолютного пути
require '/domains/user/host/config.php';
require_once '/domains/user/host/config.php';

#включение с указанием URL
include 'http://host/file.php';
```

Задача №4.1

MiniFramework.

Необходимо реализовать функцию **getPathOfAlias().**

Эта функция принимает в качестве аргумента псевдоним файлового пути (алиас) и возвращает фактический реальный путь к файлу.

Если алиас не корректный, то функция возвращает false.

MiniFramework.

Написать функцию import(), которая в качестве аргумента будет принимать псевдоним файлового пути, использовать функцию getPathOfAlias() для получения реального пути и выполнять включение файла.

Мы вводим условие, что последний компонент в файловом пути – это имя файла.
Т.е. в алиасе application.models.CUserModel имя файла CUserModel.

В функцию getPathOfALias() мы передаем алиас application.models.
Если функция вернет false, но прекращаем работу и возвращаем false.
Если функция вернет реальный путь, то объединяем его с именем файла и расширением .php. Т.е. application/models/CUserModel.php.

Проверяем существование файла.
Если файл не существует, то возвращаем false.
Если файл существует, то выполняем включение с помощью include.

Конец

http://www.php.net/manual/ru/session.upload-progress.php