PHP 5 File System Functions

The following section contains a list of useful PHP file system functions.

PHP File System Functions

The following file system functions are the part of the PHP core so you can use these functions within your script without any further installation.

Function	Description
basename()	Returns the filename component of a path
chgrp()	Changes the file group
chmod()	Changes the file mode
chown()	Changes the file owner
clearstatcache()	Clears the file status cache
copy()	Copies a file
delete()	See `unlink()` or `unset()`
dirname()	Returns the path of the parent directory
disk_free_space()	Returns available space on filesystem or disk partition
disk_total_space()	Returns the total size of a filesystem or disk partition
diskfreespace()	Returns available space on filesystem or disk partition. Alias of `disk_free_space()`
fclose()	Closes an open file pointer
feof()	Tests for end-of-file on a file pointer
fflush()	Flushes the buffered output to a file
fgetc()	Returns a character from file pointer
fgetcsv()	Gets line from file pointer and parse for CSV fields
fgets()	Read a specific number of bytes from a file
fgetss()	Reads a specific number of bytes from a file and strip HTML tags and PHP code
file()	Reads entire file into an array
file_exists()	Checks whether a file or directory exists
file_get_contents()	Reads entire file into a string

Function	Description
file_put_contents()	Write a string to a file
fileatime()	Returns the last access time of a file
filectime()	Returns the last change time of a file
filegroup()	Returns the group ID of a file
fileinode()	Returns the inode number of the file
filemtime()	Returns the last modification time of a file
fileowner()	Returns the user ID of the owner of the file
fileperms()	Returns permissions for the file
filesize()	Returns the file size
filetype()	Returns the file type
flock()	Locks or releases a file
fnmatch()	Matches a filename or string against a specified pattern
fopen()	Opens a file or URL
fpassthru()	Output all remaining data on a file pointer
fputcsv()	Format line as CSV and write to file pointer
fputs()	Alias of `fwrite()`
fread()	Reads a specific number of bytes from a file
fscanf()	Parses input from a file according to a specified format
fseek()	Seeks on a file pointer
fstat()	Returns information about a file using an open file pointer
ftell()	Returns the current position of the file read/write pointer
ftruncate()	Truncates a file to a given length
fwrite()	Writes the contents of string to the file pointer
glob()	Returns an array of filenames/directories matching a specified pattern
is_dir()	Checks whether the file is a directory
is_executable()	Checks whether the file is executable
is_file()	Checks whether the file is a regular file
is_link()	Checks whether the filename is a symbolic link
is_readable()	Checks whether a file exists and is readable
is_uploaded_file()	Checks whether the file was uploaded via HTTP POST

Function	Description
`is_writable()`	Checks whether the filename is writable
`is_writeable()`	Alias of `is_writable()`
`lchgrp()`	Changes group ownership of symlink
`lchown()`	Changes user ownership of symlink
`link()`	Create a hard link
`linkinfo()`	Returns information about a link
`lstat()`	Returns information about a file or symbolic link
`mkdir()`	Creates a directory
`move_uploaded_file()`	Moves an uploaded file to a new location
`parse_ini_file()`	Parse a configuration file
`parse_ini_string()`	Parse a configuration string
`pathinfo()`	Returns information about a file path
`pclose()`	Closes process file pointer
`popen()`	Opens process file pointer
`readfile()`	Reads a file and writes it to the output buffer
`readlink()`	Returns the target of a symbolic link
`realpath()`	Returns canonicalized absolute pathname
`realpath_cache_get()`	Returns realpath cache entries
`realpath_cache_size()`	Returns realpath cache size
`rename()`	Renames a file or directory
`rewind()`	Rewind the position of a file pointer
`rmdir()`	Removes an empty directory
`set_file_buffer()`	Sets the buffer size of a file
`stat()`	Returns information about a file
`symlink()`	Creates a symbolic link
`tempnam()`	Create temporary file with unique file name
`tmpfile()`	Creates a unique temporary file
`touch()`	Sets access and modification time of file
`umask()`	Changes the current umask
`unlink()`	Deletes a file