

## ■ファイル操作関連の主な関数

作成: 2011.12.5/2015.9.16修正 yoshi

関数	処理機能	備考(利用方法他)
<b>【ディレクトリ関連】</b>		
<b>mkdir</b>	ディレクトリを作成	mkdir(\$folder,0755);
<b>rmdir</b>	ディレクトリを削除	rmdir('examples');
<b>getcwd</b>	現在のワーキングディレクトリを取得(絶対パスでget可能)	echo getcwd();
<b>chdir</b>	現在のディレクトリを変更	chdir('public_html');
<b>opendir</b>	ディレクトリのオープン	opendir(\$dir);
<b>readdir</b>	ディレクトリの内容を読み込む	while (\$file = readdir(\$dh)) { ... }
<b>closedir</b>	ディレクトリのクローズ	closedir(\$dh);
<b>scandir</b>	ディレクトリの内容を読み取る	\$files = scandir(\$dir);
<b>chroot</b>	ルートディレクトリを変更する	chroot("/path/to/your/chroot/");
<b>【ファイル情報取得関連】</b>		
<b>realpath</b>	絶対パス名を取得(成功した場合は正規化した絶対パス名を返す)	echo realpath('/windows/system32');
<b>dirname</b>	パスからディレクトリ名を切り出す	dirname('c:/x'); // 'c:\' を返します
<b>basename</b>	パスの最後にある名前の部分を返す	\$filename=basename("/etc/sudoers.gif");
<b>stat</b>	ファイル情報をまとめて取得(mode,dev,size,atime,mtime他)	\$stat = stat('C:\php\php.exe'); echo '更新日時: ' . \$stat['mtime'];
<b>file*</b>	ファイル情報を個別に取得→*は、(node,perms,owner,size,mtime,type他)	filetype('/etc/passwd'); //ファイルタイプ
<b>file_exists</b>	ファイルが存在するかどうかのチェック	if( file_exists(\$filename) ){ ... }
<b>touch</b>	ファイルの最終アクセス時刻および最終更新日をセット (存在しないファイルの時は、空ファイル出力)	touch(\$filename,\$time); \$timeを省略すると現在の時刻
<b>is_dir</b>	ファイルがディレクトリかどうかを調べる(ファイルが存在して、かつそれがディレクトリであれば TRUE、それ以外の場合は FALSE を返します。)	//一つ上のディレクトリ(true) var_dump(is_dir('.'));
<b>is_file</b>	通常ファイルかどうかを調べる(ファイルが存在し、かつそれが通常のファイルである場合に TRUE、それ以外の場合に FALSE を返します。)	var_dump(is_file('a_file.txt')); //true
<b>【ファイル関連】</b>		
<b>unlink</b>	ファイルを削除	unlink(\$filename);
<b>glob</b>	shellパターンにマッチするファイルを検索して、配列で返す	foreach (glob("*.txt") as \$filename) { ... }
<b>copy</b>	ファイルのコピー	copy(\$org_filename,\$copy_filename);
<b>rename</b>	ファイル名の変更	rename(\$before_filename,\$after_filename);
<b>readfile</b>	ファイルをすべて読みとり、標準出力に出力	readfile(\$file);
<b>fopen</b>	ファイルのオープン(バイナリーモードでの書き込みは、「b」を付加)	\$fh = fopen(\$filename,"r");
<b>fclose</b>	ファイルのクローズ	fclose(\$fh);
<b>flock</b>	開かれているファイルをロック制御(LOCK_EX,LOCK_UN,LOCK_SH) flock(ファイルハンドル,ロック方法);	flock(\$fh,LOCK_EX);
<b>rewind</b>	ファイルポインタの位置を先頭に戻す	rewind(\$fh);
<b>fgets</b>	開かれているファイルの1行読み取り(文字列として、1行の改行文字が含まれる)	\$data = fgets(\$fh);
<b>fputs</b>	ファイルの書き込み(fwriteと同じ働き)	fputs(\$fh,\$data);
<b>fread</b>	バイナリーで画像ファイル読み取り(fopenモードで「b」の指定)	\$data = fread(\$fh, filesize(\$filename));
<b>fwrite</b>	開かれているファイルに書き込み(文字列として)	if (fwrite(\$fh, \$data) === FALSE) { ... }
<b>fgetcsv</b>	CSVファイルを読み取り(配列データとして読み込み)	while ((\$data = fgetcsv(\$fh, 1000, ",")) !== FALSE) { ... }
<b>fputcsv</b>	CSVファイルを書き込み(配列データとして書き込み)デリミタの付加可能(tab:¥tなど)	fputcsv(\$fh, \$fields,"t");
<b>feof</b>	ファイルの終端(EOF)かをチェック	while( !feof(\$fh) ){ ... }
<b>file</b>	ファイル全体を読み込んで配列に格納する	\$lines = file('http://exm.jp/');
<b>file_get_contents</b>	ファイルの内容を全て文字列に読み込む	\$page = file_get_contents('http://exm.jp/');
<b>file_put_contents</b>	文字列をファイルに書き込む(バイナリデータに対応) fopen()、fwrite()、fclose() を続けてコールしてデータをファイルに書き込むのと等価	file_put_contents(\$filename,\$item);
<b>【その他付加情報】</b>		
<b>@</b>	@は、エラー制御演算子で、式の前につけるとその式でのエラーメッセージを無視する	\$fh = @fopen(\$filename,"r");
<b>r</b>	[ファイルモード] 読み込みのみ	
<b>r+</b>	[ファイルモード] 読み込みして、書き込み	
<b>w</b>	[ファイルモード] 書き込みのみ(ファイルがない場合は、作成)	
<b>w+</b>	[ファイルモード] 読み込み/書き込みのみ(ファイルがない場合は、作成)	
<b>a</b>	[ファイルモード] 追加書き込みのみ	
<b>a+</b>	[ファイルモード] 読み込み/追加書き込みのみ(ファイルがない場合は、作成)	
<b>x</b>	[ファイルモード] 書き込みのみ(ファイルの先頭のファイルポインタから)ファイルが既に存在する場合は、エラーとなる	
<b>x+</b>	[ファイルモード] 読み込み/書き込みのみで、他はxモードの振る舞い	
<b>c</b>	[ファイルモード] 書き込みのみで、ファイルが存在しない場合には新規作成(PHP5.2.6以降)	
<b>c+</b>	[ファイルモード] 読み込み/書き込みのみで、他はcモードの振る舞い(PHP5.2.6以降)	
<b>b</b>	ファイルをオープンするとき、バイナリーモードで処理する指定で、入出力モードと一緒に指	\$fh = fopen(\$filename,"wb");

LOCK_SH	[ロック定数] 共有ロック … 読み込みは可能で、解放されるまで書き込み不可	
LOCK_EX	[ロック定数] 排他ロック … 解放されるまで、読み込み/書き込み不可	
LOCK_UN	[ロック定数] ロック解除 … ロック状態の解放	
LOCK_NB	[ロック定数] ノンブロック … ロック中ブロックモードとしない	