[**PHP Filesystem 函数** 3](#_Toc278909875)

[basename() 函数 7](#_Toc278909876)

[chgrp() 函数 8](#_Toc278909877)

[chmod() 函数 9](#_Toc278909878)

[chown() 函数 10](#_Toc278909879)

[clearstatcache() 函数 11](#_Toc278909880)

[copy() 函数 12](#_Toc278909881)

[**dirname() 函数** 13](#_Toc278909882)

[**disk\_free\_space() 函数** 14](#_Toc278909883)

[**disk\_total\_space() 函数** 15](#_Toc278909884)

[**diskfreespace() 函数** 16](#_Toc278909885)

[**fclose() 函数** 17](#_Toc278909886)

[**feof() 函数** 18](#_Toc278909887)

[**fflush() 函数** 19](#_Toc278909888)

[**fgetc() 函数** 20](#_Toc278909889)

[**fgetcsv() 函数** 21](#_Toc278909890)

[fgets() 函数 24](#_Toc278909891)

[fgetss() 函数 26](#_Toc278909892)

[file() 函数 28](#_Toc278909893)

[file\_exists() 函数 29](#_Toc278909894)

[file\_get\_contents() 函数 30](#_Toc278909895)

[file\_put\_contents() 函数 31](#_Toc278909896)

[fileatime() 函数 33](#_Toc278909897)

[filectime() 函数 34](#_Toc278909898)

[filegroup() 函数 35](#_Toc278909899)

[fileinode() 函数 36](#_Toc278909900)

[filemtime() 函数 37](#_Toc278909901)

[fileowner() 函数 38](#_Toc278909902)

[fileperms() 函数 39](#_Toc278909903)

[filesize() 函数 40](#_Toc278909904)

[filetype() 函数 41](#_Toc278909905)

[flock() 函数 42](#_Toc278909906)

[fnmatch() 函数 44](#_Toc278909907)

[fopen() 函数 45](#_Toc278909908)

[fpassthru() 函数 47](#_Toc278909909)

[fputcsv() 函数 49](#_Toc278909910)

[fputs() 函数 51](#_Toc278909911)

[fread() 函数 52](#_Toc278909912)

[fscanf() 函数 53](#_Toc278909913)

[fseek() 函数 54](#_Toc278909914)

[fstat() 函数 55](#_Toc278909915)

[ftell() 函数 58](#_Toc278909916)

[ftruncate() 函数 59](#_Toc278909917)

[fwrite() 函数 60](#_Toc278909918)

[glob() 函数 61](#_Toc278909919)

[is\_dir() 函数 63](#_Toc278909920)

[is\_executable() 函数 64](#_Toc278909921)

[is\_file() 函数 65](#_Toc278909922)

[is\_link() 函数 67](#_Toc278909923)

[is\_readable() 函数 68](#_Toc278909924)

[is\_uploaded\_file() 函数 69](#_Toc278909925)

[is\_writable() 函数 71](#_Toc278909926)

[is\_writeable() 函数 72](#_Toc278909927)

[link() 函数 73](#_Toc278909928)

[linkinfo() 函数 74](#_Toc278909929)

[lstat() 函数 74](#_Toc278909930)

[mkdir() 函数 77](#_Toc278909931)

[move\_uploaded\_file() 函数 78](#_Toc278909932)

[parse\_ini\_file() 函数 79](#_Toc278909933)

[pathinfo() 函数 82](#_Toc278909934)

[pclose() 函数 83](#_Toc278909935)

[popen() 函数 84](#_Toc278909936)

[readfile() 函数 85](#_Toc278909937)

[readlink() 函数 86](#_Toc278909938)

[realpath() 函数 87](#_Toc278909939)

[rename() 函数 88](#_Toc278909940)

[rewind() 函数 89](#_Toc278909941)

[rmdir() 函数 90](#_Toc278909942)

[set\_file\_buffer() 函数 91](#_Toc278909943)

[stat() 函数 92](#_Toc278909944)

[symlink() 函数 94](#_Toc278909945)

[tempnam() 函数 95](#_Toc278909946)

[tmpfile() 函数 96](#_Toc278909947)

[touch() 函数 97](#_Toc278909948)

[umask() 函数 98](#_Toc278909949)

[unlink() 函数 99](#_Toc278909950)

**PHP Filesystem 函数**

* [Previous Page](http://www.w3school.com.cn/php/php_ref_error.asp)
* [Next Page](http://www.w3school.com.cn/php/php_ref_filter.asp)

**PHP Filesystem 简介**

Filesystem 函数允许您访问和操作文件系统。

**安装**

Filesystem 函数是 PHP 核心的组成部分。无需安装即可使用这些函数。

**Runtime 配置**

文件系统函数的行为受到 php.ini 中设置的影响。

文件系统配置选项：

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **默认** | **描述** | **可改变** |
| allow\_url\_fopen | "1" | 本选项激活了 URL 形式的 fopen 封装协议使得可以访问 URL 对象例如文件。默认的封装协议提供用 ftp 和 http 协议来访问远程文件，一些扩展库例如 zlib 可能会注册更多的封装协议。  （PHP 4.0.4 版以后可用。） | PHP\_INI\_SYSTEM |
| user\_agent | NULL | 定义 PHP 发送的 User-Agent。  （PHP 4.3.0 版以后可用。） | PHP\_INI\_ALL |
| default\_socket\_timeout | "60" | 基于 socket 的流的默认超时时间(秒)。  （PHP 4.3.0 版以后可用。） | PHP\_INI\_ALL |
| from | "" | 定义匿名 ftp 的密码（您的 email 地址）。 | PHP\_INI\_ALL |
| auto\_detect\_line\_endings | "0" | 当设为 On 时，PHP 将检查通过 fgets() 和 file() 取得的数据中的行结束符号是符合 Unix，MS-DOS，还是 Macintosh 的习惯。  这使得 PHP 可以和 Macintosh 系统交互操作，但是默认值是 Off，因为在检测第一行的 EOL 习惯时会有很小的性能损失，而且在 Unix 系统下使用回车符号作为项目分隔符的人们会遭遇向下不兼容的行为。  （PHP 4.3.0 版以后可用。） | PHP\_INI\_ALL |

**Unix / Windows 兼容性**

当在 Unix 平台上规定路径时，正斜杠 (/) 用作目录分隔符。而在 Windows 平台上，正斜杠 (/) 和反斜杠 (\) 均可使用。

**PHP Filesystem 函数**

PHP：指示支持该函数的最早的 PHP 版本。

|  |  |  |
| --- | --- | --- |
| **函数** | **描述** | **PHP** |
| [basename()](http://www.w3school.com.cn/php/func_filesystem_basename.asp) | 返回路径中的文件名部分。 | 3 |
| [chgrp()](http://www.w3school.com.cn/php/func_filesystem_chgrp.asp) | 改变文件组。 | 3 |
| [chmod()](http://www.w3school.com.cn/php/func_filesystem_chmod.asp) | 改变文件模式。 | 3 |
| [chown()](http://www.w3school.com.cn/php/func_filesystem_chown.asp) | 改变文件所有者。 | 3 |
| [clearstatcache()](http://www.w3school.com.cn/php/func_filesystem_clearstatcache.asp) | 清除文件状态缓存。 | 3 |
| [copy()](http://www.w3school.com.cn/php/func_filesystem_copy.asp) | 复制文件。 | 3 |
| delete() | 参见 [unlink()](http://www.w3school.com.cn/php/func_filesystem_unlink.asp) 或 unset()。 |  |
| [dirname()](http://www.w3school.com.cn/php/func_filesystem_dirname.asp) | 返回路径中的目录名称部分。 | 3 |
| [disk\_free\_space()](http://www.w3school.com.cn/php/func_filesystem_disk_free_space.asp) | 返回目录的可用空间。 | 4 |
| [disk\_total\_space()](http://www.w3school.com.cn/php/func_filesystem_disk_total_space.asp) | 返回一个目录的磁盘总容量。 | 4 |
| [diskfreespace()](http://www.w3school.com.cn/php/func_filesystem_diskfreespace.asp) | disk\_free\_space() 的别名。 | 3 |
| [fclose()](http://www.w3school.com.cn/php/func_filesystem_fclose.asp) | 关闭打开的文件。 | 3 |
| [feof()](http://www.w3school.com.cn/php/func_filesystem_feof.asp) | 测试文件指针是否到了文件结束的位置。 | 3 |
| [fflush()](http://www.w3school.com.cn/php/func_filesystem_fflush.asp) | 向打开的文件输出缓冲内容。 | 4 |
| [fgetc()](http://www.w3school.com.cn/php/func_filesystem_fgetc.asp) | 从打开的文件中返回字符。 | 3 |
| [fgetcsv()](http://www.w3school.com.cn/php/func_filesystem_fgetcsv.asp) | 从打开的文件中解析一行，校验 CSV 字段。 | 3 |
| [fgets()](http://www.w3school.com.cn/php/func_filesystem_fgets.asp) | 从打开的文件中返回一行。 | 3 |
| [fgetss()](http://www.w3school.com.cn/php/func_filesystem_fgetss.asp) | 从打开的文件中读取一行并过滤掉 HTML 和 PHP 标记。 | 3 |
| [file()](http://www.w3school.com.cn/php/func_filesystem_file.asp) | 把文件读入一个数组中。 | 3 |
| [file\_exists()](http://www.w3school.com.cn/php/func_filesystem_file_exists.asp) | 检查文件或目录是否存在。 | 3 |
| [file\_get\_contents()](http://www.w3school.com.cn/php/func_filesystem_file_get_contents.asp) | 将文件读入字符串。 | 4 |
| [file\_put\_contents](http://www.w3school.com.cn/php/func_filesystem_file_put_contents.asp) | 将字符串写入文件。 | 5 |
| [fileatime()](http://www.w3school.com.cn/php/func_filesystem_fileatime.asp) | 返回文件的上次访问时间。 | 3 |
| [filectime()](http://www.w3school.com.cn/php/func_filesystem_filectime.asp) | 返回文件的上次改变时间。 | 3 |
| [filegroup()](http://www.w3school.com.cn/php/func_filesystem_filegroup.asp) | 返回文件的组 ID。 | 3 |
| [fileinode()](http://www.w3school.com.cn/php/func_filesystem_fileinode.asp) | 返回文件的 inode 编号。 | 3 |
| [filemtime()](http://www.w3school.com.cn/php/func_filesystem_filemtime.asp) | 返回文件的上次修改时间。 | 3 |
| [fileowner()](http://www.w3school.com.cn/php/func_filesystem_fileowner.asp) | 文件的 user ID （所有者）。 | 3 |
| [fileperms()](http://www.w3school.com.cn/php/func_filesystem_fileperms.asp) | 返回文件的权限。 | 3 |
| [filesize()](http://www.w3school.com.cn/php/func_filesystem_filesize.asp) | 返回文件大小。 | 3 |
| [filetype()](http://www.w3school.com.cn/php/func_filesystem_filetype.asp) | 返回文件类型。 | 3 |
| [flock()](http://www.w3school.com.cn/php/func_filesystem_flock.asp) | 锁定或释放文件。 | 3 |
| [fnmatch()](http://www.w3school.com.cn/php/func_filesystem_fnmatch.asp) | 根据指定的模式来匹配文件名或字符串。 | 4 |
| [fopen()](http://www.w3school.com.cn/php/func_filesystem_fopen.asp) | 打开一个文件或 URL。 | 3 |
| [fpassthru()](http://www.w3school.com.cn/php/func_filesystem_fpassthru.asp) | 从打开的文件中读数据，直到 EOF，并向输出缓冲写结果。 | 3 |
| [fputcsv()](http://www.w3school.com.cn/php/func_filesystem_fputcsv.asp) | 将行格式化为 CSV 并写入一个打开的文件中。 | 5 |
| [fputs()](http://www.w3school.com.cn/php/func_filesystem_fputs.asp) | fwrite() 的别名。 | 3 |
| [fread()](http://www.w3school.com.cn/php/func_filesystem_fread.asp) | 读取打开的文件。 | 3 |
| [fscanf()](http://www.w3school.com.cn/php/func_filesystem_fscanf.asp) | 根据指定的格式对输入进行解析。 | 4 |
| [fseek()](http://www.w3school.com.cn/php/func_filesystem_fseek.asp) | 在打开的文件中定位。 | 3 |
| [fstat()](http://www.w3school.com.cn/php/func_filesystem_fstat.asp) | 返回关于一个打开的文件的信息。 | 4 |
| [ftell()](http://www.w3school.com.cn/php/func_filesystem_ftell.asp) | 返回文件指针的读/写位置 | 3 |
| [ftruncate()](http://www.w3school.com.cn/php/func_filesystem_ftruncate.asp) | 将文件截断到指定的长度。 | 4 |
| [fwrite()](http://www.w3school.com.cn/php/func_filesystem_fwrite.asp) | 写入文件。 | 3 |
| [glob()](http://www.w3school.com.cn/php/func_filesystem_glob.asp) | 返回一个包含匹配指定模式的文件名/目录的数组。 | 4 |
| [is\_dir()](http://www.w3school.com.cn/php/func_filesystem_is_dir.asp) | 判断指定的文件名是否是一个目录。 | 3 |
| [is\_executable()](http://www.w3school.com.cn/php/func_filesystem_is_executable.asp) | 判断文件是否可执行。 | 3 |
| [is\_file()](http://www.w3school.com.cn/php/func_filesystem_is_file.asp) | 判断指定文件是否为常规的文件。 | 3 |
| [is\_link()](http://www.w3school.com.cn/php/func_filesystem_is_link.asp) | 判断指定的文件是否是连接。 | 3 |
| [is\_readable()](http://www.w3school.com.cn/php/func_filesystem_is_readable.asp) | 判断文件是否可读。 | 3 |
| [is\_uploaded\_file()](http://www.w3school.com.cn/php/func_filesystem_is_uploaded_file.asp) | 判断文件是否是通过 HTTP POST 上传的。 | 3 |
| [is\_writable()](http://www.w3school.com.cn/php/func_filesystem_is_writable.asp) | 判断文件是否可写。 | 4 |
| [is\_writeable()](http://www.w3school.com.cn/php/func_filesystem_is_writeable.asp) | is\_writable() 的别名。 | 3 |
| [link()](http://www.w3school.com.cn/php/func_filesystem_link.asp) | 创建一个硬连接。 | 3 |
| [linkinfo()](http://www.w3school.com.cn/php/func_filesystem_linkinfo.asp) | 返回有关一个硬连接的信息。 | 3 |
| [lstat()](http://www.w3school.com.cn/php/func_filesystem_lstat.asp) | 返回关于文件或符号连接的信息。 | 3 |
| [mkdir()](http://www.w3school.com.cn/php/func_filesystem_mkdir.asp) | 创建目录。 | 3 |
| [move\_uploaded\_file()](http://www.w3school.com.cn/php/func_filesystem_move_uploaded_file.asp) | 将上传的文件移动到新位置。 | 4 |
| [parse\_ini\_file()](http://www.w3school.com.cn/php/func_filesystem_parse_ini_file.asp) | 解析一个配置文件。 | 4 |
| [pathinfo()](http://www.w3school.com.cn/php/func_filesystem_pathinfo.asp) | 返回关于文件路径的信息。 | 4 |
| [pclose()](http://www.w3school.com.cn/php/func_filesystem_pclose.asp) | 关闭有 popen() 打开的进程。 | 3 |
| [popen()](http://www.w3school.com.cn/php/func_filesystem_popen.asp) | 打开一个进程。 | 3 |
| [readfile()](http://www.w3school.com.cn/php/func_filesystem_readfile.asp) | 读取一个文件，并输出到输出缓冲。 | 3 |
| [readlink()](http://www.w3school.com.cn/php/func_filesystem_readlink.asp) | 返回符号连接的目标。 | 3 |
| [realpath()](http://www.w3school.com.cn/php/func_filesystem_realpath.asp) | 返回绝对路径名。 | 4 |
| [rename()](http://www.w3school.com.cn/php/func_filesystem_rename.asp) | 重名名文件或目录。 | 3 |
| [rewind()](http://www.w3school.com.cn/php/func_filesystem_rewind.asp) | 倒回文件指针的位置。 | 3 |
| [rmdir()](http://www.w3school.com.cn/php/func_filesystem_rmdir.asp) | 删除空的目录。 | 3 |
| [set\_file\_buffer()](http://www.w3school.com.cn/php/func_filesystem_set_file_buffer.asp) | 设置已打开文件的缓冲大小。 | 3 |
| [stat()](http://www.w3school.com.cn/php/func_filesystem_stat.asp) | 返回关于文件的信息。 | 3 |
| [symlink()](http://www.w3school.com.cn/php/func_filesystem_symlink.asp) | 创建符号连接。 | 3 |
| [tempnam()](http://www.w3school.com.cn/php/func_filesystem_tempnam.asp) | 创建唯一的临时文件。 | 3 |
| [tmpfile()](http://www.w3school.com.cn/php/func_filesystem_tmpfile.asp) | 建立临时文件。 | 3 |
| [touch()](http://www.w3school.com.cn/php/func_filesystem_touch.asp) | 设置文件的访问和修改时间。 | 3 |
| [umask()](http://www.w3school.com.cn/php/func_filesystem_umask.asp) | 改变文件的文件权限。 | 3 |
| [unlink()](http://www.w3school.com.cn/php/func_filesystem_unlink.asp) | 删除文件。 | 3 |

**PHP Filesystem 常量**

PHP：指示支持该常量的最早的 PHP 版本。

|  |  |  |
| --- | --- | --- |
| **常量** | **描述** | **PHP** |
| GLOB\_BRACE |  |  |
| GLOB\_ONLYDIR |  |  |
| GLOB\_MARK |  |  |
| GLOB\_NOSORT |  |  |
| GLOB\_NOCHECK |  |  |
| GLOB\_NOESCAPE |  |  |
| PATHINFO\_DIRNAME |  |  |
| PATHINFO\_BASENAME |  |  |
| PATHINFO\_EXTENSION |  |  |
| FILE\_USE\_INCLUDE\_PATH |  |  |
| FILE\_APPEND |  |  |
| FILE\_IGNORE\_NEW\_LINES |  |  |
| FILE\_SKIP\_EMPTY\_LINES |  |  |

# basename() 函数

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

## 定义和用法

basename() 函数返回路径中的文件名部分。

### 语法

basename(path,suffix)

|  |  |
| --- | --- |
| **参数** | **描述** |
| path | 必需。规定要检查的路径。 |
| suffix | 可选。规定文件扩展名。如果文件有 suffix，则不会输出这个扩展名。 |

## 例子

<?php

$path = "/testweb/home.php";

//显示带有文件扩展名的文件名

echo basename($path);

//显示不带有文件扩展名的文件名

echo basename($path,".php");

?>

输出：

home.php

home

# chgrp() 函数

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

## 定义和用法

chgrp() 函数改变文件所属的组。

如果成功则返回 TRUE，否则返回 FALSE。

### 语法

chgrp(file,group)

|  |  |
| --- | --- |
| **参数** | **描述** |
| file | 必需。规定要检查的文件。 |
| group | 可选。规定新的组。可以是组名或组的 ID。 |

### 说明

尝试将文件 *file* 所属的组改成 *group* （通过组名或组 ID 指定）。

只有超级用户可以任意修改文件的组，其它用户可能只能将文件的组改成该用户自己所在的组。

## 例子

<?php

chgrp("test.txt","admin")

?>

[PHP Filesystem 函](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

# chmod() 函数

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

## 定义和用法

chmod() 函数改变文件模式。

如果成功则返回 TRUE，否则返回 FALSE。

### 语法

chmod(file,mode)

|  |  |
| --- | --- |
| **参数** | **描述** |
| file | 必需。规定要检查的文件。 |
| mode | 可选。规定新的权限。  mode 参数由 4 个数字组成：   * 第一个数字永远是 0 * 第二个数字规定所有者的权限 * 第二个数字规定所有者所属的用户组的权限 * 第四个数字规定其他所有人的权限   可能的值（如需设置多个权限，请对下面的数字进行总计）：   * 1 - 执行权限 * 2 - 写权限 * 4 - 读权限 |

## 例子

<?php

// 所有者可读写，其他人没有任何权限

chmod("test.txt",0600);

// 所有者可读写，其他人可读

chmod("test.txt",0644);

// 所有者有所有权限，其他所有人可读和执行

chmod("test.txt",0755);

// 所有者有所有权限，所有者所在的组可读

chmod("test.txt",0740);

?>

[PHP Filesystem 函](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

# chown() 函数

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

## 定义和用法

chown() 函数改变指定文件的所有者。

如果成功则返回 TRUE，否则返回 FALSE。

### 语法

chown(file,owner)

|  |  |
| --- | --- |
| **参数** | **描述** |
| file | 必需。规定要检查的文件。 |
| owner | 规定新的所有者。可以是用户名或用户的 ID。 |

### 说明

尝试将文件 *file* 的所有者改成用户 *owner* （由用户名或用户 ID 指定）。只有超级用户可以改变文件的所有者。

## 例子

<?php

chown("test.txt","charles")

?>

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

# clearstatcache() 函数

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

## 定义和用法

clearstatcache() 函数清除文件状态缓存。

clearstatcache() 函数会缓存某些函数的返回信息，以便提供更高的性能。但是有时候，比如在一个脚本中多次检查同一个文件，而该文件在此脚本执行期间有被删除或修改的危险时，你需要清除文件状态缓存，以便获得正确的结果。要做到这一点，就需要使用 clearstatcache() 函数。

会进行缓存的函数，即受 clearstatcache() 函数影响的函数：

* stat()
* lstat()
* file\_exists()
* is\_writable()
* is\_readable()
* is\_executable()
* is\_file()
* is\_dir()
* is\_link()
* filectime()
* fileatime()
* filemtime()
* fileinode()
* filegroup()
* fileowner()
* filesize()
* filetype()
* fileperms()

### 语法

clearstatcache()

## 例子

<?php

//检查文件大小

echo filesize("test.txt");

$file = fopen("test.txt", "a+");

// 截取文件

ftruncate($file,100);

fclose($file);

//清除缓存并再次检查文件大小

clearstatcache();

echo filesize("test.txt");

?>

输出：

792

100

# copy() 函数

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

## 定义和用法

clearstatcache() 函数拷贝文件。

### 语法

copy(source,destination)

|  |  |
| --- | --- |
| **参数** | **描述** |
| source | 必需。规定要复制的文件。 |
| destination | 必需。规定复制文件的目的地。 |

### 说明

将文件从 *source* 拷贝到 *destination*。如果成功则返回 TRUE，否则返回 FALSE。

## 提示和注释

提示：如果要移动文件的话，请使用 [rename() 函数](http://www.w3school.com.cn/php/func_filesystem_rename.asp)。

注释：从 PHP 4.3.0 开始，如果启用了 "fopen wrappers" 的话，source 和 *destination* 都可以是 URL。更多信息见 [fopen()](http://www.w3school.com.cn/php/func_filesystem_fopen.asp)。如果 *destination* 是一个 URL，则如果封装协议不支持覆盖已有的文件时拷贝操作会失败。

重要事项：如果目标文件已存在，将会被覆盖。

## 例子

<?php

echo copy("source.txt","target.txt");

?>

输出：

1

**dirname() 函数**

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

**定义和用法**

dirname() 函数返回路径中的目录部分。

**语法**

dirname(path)

|  |  |
| --- | --- |
| **参数** | **描述** |
| path | 必需。规定要检查的路径。 |

**说明**

*path* 参数是一个包含有指向一个文件的全路径的字符串。该函数返回去掉文件名后的目录名。

**例子**

<?php

echo dirname("c:/testweb/home.php");

echo dirname("/testweb/home.php");

?>

输出：

c:/testweb

/testweb

**disk\_free\_space() 函数**

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

**定义和用法**

disk\_free\_space() 函数返回目录中的可用空间

**语法**

disk\_free\_space(directory)

|  |  |
| --- | --- |
| **参数** | **描述** |
| directory | 必需。规定要检查的目录。 |

**说明**

*directory* 参数是一个目录的字符串。该函数将根据相应的文件系统或磁盘分区返回可用的字节数。

**例子**

<?php

echo disk\_free\_space("C:");

?>

输出：

209693288558

**disk\_total\_space() 函数**

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

**定义和用法**

disk\_total\_space() 函数返回指定目录的磁盘总大小。

**语法**

disk\_total\_space(directory)

|  |  |
| --- | --- |
| **参数** | **描述** |
| directory | 必需。规定要检查的目录。 |

**说明**

*directory* 参数是一个目录的字符串。该函数将根据相应的文件系统或磁盘分区返回所有的字节数。

**提示和注释**

提示：本函数返回的是该目录所在的磁盘分区的总大小，因此在给出同一个磁盘分区的不同目录作为参数所得到的结果完全相同。在 Unix 和 Windows 200x/XP 中都支持将一个磁盘分区加载为一个子目录，这时正确使用本函数就很有意义。

**例子**

<?php

echo disk\_total\_space("C:");

?>

输出类似这样：

509693888668

**diskfreespace() 函数**

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

**定义和用法**

diskfreespace() 函数返回目录中的可用空间。该函数是 [disk\_free\_space()](http://www.w3school.com.cn/php/func_filesystem_disk_free_space.asp) 函数的别名。

**语法**

diskfreespace(directory)

|  |  |
| --- | --- |
| **参数** | **描述** |
| directory | 必需。规定要检查的目录。 |

**说明**

*directory* 参数是一个目录的字符串。该函数将根据相应的文件系统或磁盘分区返回可用的字节数。

**例子**

<?php

echo diskfreespace("C:");

?>

输出：

209693288558

**fclose() 函数**

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

**定义和用法**

fclose() 函数关闭一个打开文件。

**语法**

fclose(file)

|  |  |
| --- | --- |
| **参数** | **描述** |
| file | 必需。规定要关闭的文件。 |

**说明**

*file* 参数是一个文件指针。fclose() 函数关闭该指针指向的文件。

如果成功则返回 true，否则返回 false。

文件指针必须有效，并且是通过 [fopen()](http://www.w3school.com.cn/php/func_filesystem_fclose.asp) 或 [fsockopen()](http://www.w3school.com.cn/php/func_filesystem_fclose.asp) 成功打开的。

**例子**

<?php

$file = fopen("test.txt","r");

//执行的一些代码...

fclose($file);

?>

**feof() 函数**

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

**定义和用法**

feof() 函数检测是否已到达文件末尾 (eof)。

如果文件指针到了 EOF 或者出错时则返回 TRUE，否则返回一个错误（包括 socket 超时），其它情况则返回 FALSE。

**语法**

feof(file)

|  |  |
| --- | --- |
| **参数** | **描述** |
| file | 必需。规定要检查的打开文件。 |

**说明**

*file* 参数是一个文件指针。这个文件指针必须有效，并且必须指向一个由 [fopen()](http://www.w3school.com.cn/php/func_filesystem_fopen.asp) 或 fsockopen() 成功打开（但还没有被 [fclose()](http://www.w3school.com.cn/php/func_filesystem_fclose.asp) 关闭）的文件。

**提示和注释**

提示：feof() 函数对遍历长度未知的数据很有用。

注意：如果服务器没有关闭由 fsockopen() 所打开的连接，feof() 会一直等待直到超时而返回 TRUE。默认的超时限制是 60 秒，可以使用 stream\_set\_timeout() 来改变这个值。

注意：如果传递的文件指针无效可能会陷入无限循环中，因为 EOF 不会返回 TRUE。

**例子**

<?php

$file = fopen("test.txt", "r");

//输出文本中所有的行，直到文件结束为止。

while(! feof($file))

{

echo fgets($file). "<br />";

}

fclose($file);

?>

输出：

Hello, this is a test file.

There are three lines here.

This is the last line.

**fflush() 函数**

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

**定义和用法**

fflush() 函数将缓冲内容输出到文件。

**语法**

fflush(file)

|  |  |
| --- | --- |
| **参数** | **描述** |
| file | 必需。规定要检查的文件流。 |

**说明**

本函数强制将所有缓冲的输出写入 *file* 文件句柄所指向的资源。如果成功则返回 true，否则返回 false。

文件指针必须有效，并且必须指向一个由 [fopen()](http://www.w3school.com.cn/php/func_filesystem_fopen.asp) 或 fsockopen() 成功打开（但还没有被 [fclose()](http://www.w3school.com.cn/php/func_filesystem_fclose.asp) 关闭）的文件。

**例子**

<?php

file = fopen("test.txt","r+");

// 一些代码

fflush($file);

?>

**fgetc() 函数**

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

**定义和用法**

fgetc() 函数从文件指针中读取一个字符。

**语法**

fgetc(file)

|  |  |
| --- | --- |
| **参数** | **描述** |
| file | 必需。规定要检查的文件。 |

**说明**

返回一个包含有一个字符的字符串，该字符从 *file* 指向的文件中得到。碰到 EOF 则返回 false。

文件指针必须有效，并且必须指向一个由 [fopen()](http://www.w3school.com.cn/php/func_filesystem_fopen.asp) 或 fsockopen() 成功打开（但还没有被 [fclose()](http://www.w3school.com.cn/php/func_filesystem_fclose.asp) 关闭）的文件。

**提示和注释**

注意：该函数可能返回布尔值 false，但也可能返回一个与 false 等值的非布尔值，例如 0 或者 ""。

注释：该函数可安全用于二进制对象。

**例子**

**例子 1**

<?php

$file = fopen("test.txt","r");

echo fgetc($file);

fclose($file);

?>

输出类似：

H

**例子 2**

<?php

$file = fopen("test.txt","r");

while (! feof ($file))

{

echo fgetc($file);

}

fclose($file);

?>

输出类似：

Hello, this is a test file.

**fgetcsv() 函数**

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

**定义和用法**

fgetcsv() 函数从文件指针中读入一行并解析 CSV 字段。

与 [fgets()](http://www.w3school.com.cn/php/func_filesystem_fgets.asp) 类似，不同的是 fgetcsv() 解析读入的行并找出 CSV 格式的字段，然后返回一个包含这些字段的数组。

fgetcsv() 出错时返回 FALSE，包括碰到文件结束时。

注释：从 PHP 4.3.5 起，fgetcsv() 的操作是二进制安全的。

**语法**

fgetcsv(file,length,separator,enclosure)

|  |  |
| --- | --- |
| **参数** | **描述** |
| file | 必需。规定要检查的文件。 |
| length | 可选。规定行的最大长度。必须大于 CVS 文件内最长的一行。  在 PHP 5 中该参数是可选的。在 PHP 5 之前是必需的。  如果忽略（在 PHP 5.0.4 以后的版本中设为 0）该参数的话，那么长度就没有限制，不过可能会影响执行效率。 |
| separator | 可选。设置字段分界符（只允许一个字符），默认值为逗号。 |
| enclosure | 可选。设置字段环绕符（只允许一个字符），默认值为双引号。  该参数是在 PHP 4.3.0 中添加的。 |

**提示和注释**

注释：CSV 文件中的空行将被返回为一个包含有单个 null 字段的数组，不会被当成错误。

注释：该函数对区域设置是敏感的。比如说 LANG 设为 en\_US.UTF-8 的话，单字节编码的文件就会出现读取错误。

注释：如果碰到 PHP 在读取文件时不能识别 Macintosh 文件的行结束符，可以激活 auto\_detect\_line\_endings 运行时配置选项。

**例子**

**例子 1**

<?php

$file = fopen("contacts.csv","r");

print\_r(fgetcsv($file));

fclose($file);

?>

CSV 文件：

George, John, Thomas, USA

James, Adrew, Martin, USA

输出类似：

Array

(

[0] => George

[1] => John

[2] => Thomas

[3] => USA

)

**例子 2**

<?php

$file = fopen("contacts.csv","r");

while(! feof($file))

{

print\_r(fgetcsv($file));

}

fclose($file);

?>

CSV 文件：

George, John, Thomas, USA

James, Adrew, Martin, USA

输出类似：

Array

(

[0] => George

[1] => John

[2] => Thomas

[3] => USA

Array

(

[0] => James

[1] => Adrew

[2] => Martin

[3] => USA

)

# fgets() 函数

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

## 定义和用法

fgets() 函数从文件指针中读取一行。

### 语法

fgets(file,length)

|  |  |
| --- | --- |
| **参数** | **描述** |
| file | 必需。规定要读取的文件。 |
| length | 可选。规定要读取的字节数。默认是 1024 字节。 |

### 说明

从 *file* 指向的文件中读取一行并返回长度最多为 *length* - 1 字节的字符串。碰到换行符（包括在返回值中）、EOF 或者已经读取了 *length* - 1 字节后停止（要看先碰到那一种情况）。如果没有指定 *length*，则默认为 1K，或者说 1024 字节。

若失败，则返回 false。

## 提示和注释

注释：*length* 参数从 PHP 4.2.0 起成为可选项，如果忽略，则行的长度被假定为 1024 字节。从 PHP 4.3 开始，忽略掉 *length* 将继续从流中读取数据直到行结束。如果文件中的大多数行都大于 8 KB，则在脚本中指定最大行的长度在利用资源上更为有效。

注释：从 PHP 4.3 开始本函数可以安全用于二进制文件。早期的版本则不行。

注释：如果碰到 PHP 在读取文件时不能识别 Macintosh 文件的行结束符，可以激活 auto\_detect\_line\_endings 运行时配置选项。

## 例子

### 例子 1

<?php

$file = fopen("test.txt","r");

echo fgets($file);

fclose($file);

?>

输出类似：

Hello, this is a test file.

### 例子 2

<?php

$file = fopen("test.txt","r");

while(! feof($file))

{

echo fgets($file). "<br />";

}

fclose($file);

?>

输出类似：

Hello, this is a test file.

There are three lines here.

This is the last line.

# fgetss() 函数

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

## 定义和用法

fgetss() 函数从打开的文件中读取一行并过滤掉 HTML 和 PHP 标记。

与 [fgets()](http://www.w3school.com.cn/php/func_filesystem_fgets.asp) 相同，不同的是 fgetss 尝试从读取的文本中去掉任何 HTML 和 PHP 标记。

### 语法

fgetss(file,length,tags)

|  |  |
| --- | --- |
| **参数** | **描述** |
| file | 必需。规定要读取的文件。 |
| length | 可选。规定要读取的字节数。默认是 1024 字节。该参数在 PHP 5 之前是必需的。 |
| tags | 可选。规定将被删除的标签。 |

### 说明

可以用可选的第三个参数 *tags* 指定哪些标记不被去掉。

若失败，则返回 false。

## 例子

### 例子 1

<?php

$file = fopen("test.htm","r");

echo fgetss($file);

fclose($file);

?>

输出类似：

This is a paragraph.

### 例子 2

<?php

$file = fopen("test.htm","r");

echo fgetss($file,1024,"<p>,<b>");

fclose($file);

?>

输出类似：

**This is a paragraph.**

输出的源代码是：

<p><b>This is a paragraph.</b></p>

# file() 函数

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

## 定义和用法

file() 函数把整个文件读入一个数组中。

与 [file\_get\_contents()](http://www.w3school.com.cn/php/func_filesystem_file_get_contents.asp) 类似，不同的是 file() 将文件作为一个数组返回。数组中的每个单元都是文件中相应的一行，包括换行符在内。

如果失败，则返回 false。

### 语法

file(path,include\_path,context)

|  |  |
| --- | --- |
| **参数** | **描述** |
| path | 必需。规定要读取的文件。 |
| include\_path | 可选。如果也想在 include\_path 中搜寻文件的话，可以将该参数设为 "1"。 |
| context | 可选。规定文件句柄的环境。  context 是一套可以修改流的行为的选项。若使用 null，则忽略。 |

### 说明

对 *context* 的支持是 PHP 5.0.0 添加的。

返回的数组中每一行都包括了行结束符，因此如果不需要行结束符时还需要使用 rtrim() 函数。

## 提示和注释

注释：从 PHP 4.3.0 开始，可以用 [file\_get\_contents()](http://www.w3school.com.cn/php/func_filesystem_file_get_contents.asp) 来将文件读入到一个字符串并返回。

注释：从 PHP 4.3.0 开始，file() 可以安全用于二进制文件。

注释：如果碰到 PHP 在读取文件时不能识别 Macintosh 文件的行结束符，可以激活 auto\_detect\_line\_endings 运行时配置选项。

## 例子

<?php

print\_r(file("test.txt"));

?>

输出：

Array

(

[0] => Hello World. Testing testing!

[1] => Another day, another line.

[2] => If the array picks up this line,

[3] => then is it a pickup line?

)

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

# file\_exists() 函数

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

## 定义和用法

file\_exists() 函数检查文件或目录是否存在。

如果指定的文件或目录存在则返回 true，否则返回 false。

### 语法

file\_exists(path)

|  |  |
| --- | --- |
| **参数** | **描述** |
| path | 必需。规定要检查的路径。 |

## 例子

<?php

echo file\_exists("test.txt");

?>

输出：

1

# file\_get\_contents() 函数

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

## 定义和用法

file\_get\_contents() 函数把整个文件读入一个字符串中。

和 [file()](http://www.w3school.com.cn/php/func_filesystem_file.asp) 一样，不同的是 file\_get\_contents() 把文件读入一个字符串。

file\_get\_contents() 函数是用于将文件的内容读入到一个字符串中的首选方法。如果操作系统支持，还会使用内存映射技术来增强性能。

### 语法

file\_get\_contents(*path*,*include\_path*,*context*,*start*,*max\_length*)

|  |  |
| --- | --- |
| **参数** | **描述** |
| path | 必需。规定要读取的文件。 |
| include\_path | 可选。如果也想在 include\_path 中搜寻文件的话，可以将该参数设为 "1"。 |
| context | 可选。规定文件句柄的环境。  context 是一套可以修改流的行为的选项。若使用 null，则忽略。 |
| start | 可选。规定在文件中开始读取的位置。该参数是 PHP 5.1 新加的。 |
| max\_length | 可选。规定读取的字节数。该参数是 PHP 5.1 新加的。 |

### 说明

对 *context* 参数的支持是 PHP 5.0.0 添加的。

## 提示和注释

注释：本函数可安全用于二进制对象。

## 例子

<?php

echo file\_get\_contents("test.txt");

?>

输出：

This is a test file with test text.

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

# file\_put\_contents() 函数

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

## 定义和用法

file\_put\_contents() 函数把一个字符串写入文件中。

与依次调用 fopen()，fwrite() 以及 fclose() 功能一样。

### 语法

file\_put\_contents(file,data,mode,context)

|  |  |
| --- | --- |
| **参数** | **描述** |
| file | 必需。规定要写入数据的文件。如果文件不存在，则创建一个新文件。 |
| data | 可选。规定要写入文件的数据。可以是字符串、数组或数据流。 |
| mode | 可选。规定如何打开/写入文件。可能的值：   * FILE\_USE\_INCLUDE\_PATH * FILE\_APPEND * LOCK\_EX |
| context | 可选。规定文件句柄的环境。  context 是一套可以修改流的行为的选项。若使用 null，则忽略。 |

### 说明

参数 *data* 可以是数组（但不能是多维数组）。

自 PHP 5.1.0 起，*data* 参数也可以被指定为 stream 资源，stream 中所保存的缓存数据将被写入到指定文件中，这种用法就相似于使用 stream\_copy\_to\_stream() 函数。

对 *context* 参数的支持是 PHP 5.0.0 添加的。

### 返回值

该函数将返回写入到文件内数据的字节数。

## 提示和注释

提示：使用 FILE\_APPEND 可避免删除文件中已有的内容。

注释：本函数可安全用于二进制对象。

## 例子

<?php

echo file\_put\_contents("test.txt","Hello World. Testing!");

?>

输出：

26

# fileatime() 函数

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

## 定义和用法

fileatime() 函数返回指定文件的上次访问时间。

该函数返回文件上次被访问的时间。如果出错则返回 false。时间以 Unix 时间戳的方式返回。

### 语法

fileatime(filename)

|  |  |
| --- | --- |
| **参数** | **描述** |
| filename | 必需。规定要检查的文件。 |

## 提示和注释

提示：本函数的结果会被缓存。请使用 [clearstatcache()](http://www.w3school.com.cn/php/func_filesystem_clearstatcache.asp) 来清除缓存。

注释：文件的 atime 应该在不论何时读取了该文件中的数据块时被更改。当一个应用程序定期访问大量文件或目录时很影响性能。有些 Unix 文件系统可以在加载时关闭 atime 的更新以提高这类程序的性能。USENET 新闻组假脱机是一个常见的例子。在这种文件系统下，本函数没有用处。

## 例子

<?php

echo fileatime("test.txt");

echo "Last access: ".date("F d Y H:i:s.",fileatime("test.txt"));

?>

输出：

1140684501

Last access: February 23 2006 09:48:21.

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

# filectime() 函数

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

## 定义和用法

filectime() 函数返回指定文件的上次 inode 修改时间。

该函数返回文件上次 inode 被修改的时间。如果出错则返回 false。时间以 Unix 时间戳的方式返回。

### 语法

fileatime(filename)

|  |  |
| --- | --- |
| **参数** | **描述** |
| filename | 必需。规定要检查的文件。 |

## 提示和注释

提示：本函数的结果会被缓存。请使用 [clearstatcache()](http://www.w3school.com.cn/php/func_filesystem_clearstatcache.asp) 来清除缓存。

注意：在大多数 Unix 文件系统中，当一个文件的 inode 数据被改变时则该文件被认为是修改了。也就是说，当文件的权限，所有者，所有组或其它 inode 中的元数据被更新时。参见 [filemtime()](http://www.w3school.com.cn/php/func_filesystem_filemtime.asp)（这才是你想用于在 Web 页面中建立“最后更新时间”脚注的函数）和 [fileatime()](http://www.w3school.com.cn/php/func_filesystem_fileatime.asp)。

注释：某些 Unix 说明文本中把 ctime 说成是该文件建立的时间，这是错的。在大多数 Unix 文件系统中，没有 Unix 文件的建立时间。

## 例子

<?php

echo filectime("test.txt");

echo "Last change: ".date("F d Y H:i:s.",filectime("test.txt"));

?>

输出：

1138609592

Last change: January 30 2006 09:26:32.

# filegroup() 函数

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

## 定义和用法

filegroup() 函数返回指定文件的组 ID。

若成功，则返回指定文件所属组的 ID。若失败，则返回 false 以及一个 E\_WARNING 级别的错误。

组 ID 以数字格式返回。

### 语法

filegroup(filename)

|  |  |
| --- | --- |
| **参数** | **描述** |
| filename | 必需。规定要检查的文件。 |

## 提示和注释

提示：本函数的结果会被缓存。请使用 [clearstatcache()](http://www.w3school.com.cn/php/func_filesystem_clearstatcache.asp) 来清除缓存。

提示：请使用 posix\_getgrgid() 来将组 ID 转换为组名。

## 例子

<?php

echo filegroup("test.txt");

?>

# fileinode() 函数

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

## 定义和用法

fileinode() 函数返回文件的 inode 编号。

若成功，则返回指定文件的 inode 节点号。若失败，则返回 false。

### 语法

fileinode(filename)

|  |  |
| --- | --- |
| **参数** | **描述** |
| filename | 必需。规定要检查的文件。 |

## 提示和注释

提示：本函数的结果会被缓存。请使用 [clearstatcache()](http://www.w3school.com.cn/php/func_filesystem_clearstatcache.asp) 来清除缓存。

## 例子

<?php

echo fileinode("test.txt");

?>

# filemtime() 函数

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

## 定义和用法

filemtime() 函数返回文件内容上次的修改时间。

若成功，则时间以 Unix 时间戳的方式返回。若失败，则返回 false。

### 语法

filemtime(filename)

|  |  |
| --- | --- |
| **参数** | **描述** |
| filename | 必需。规定要检查的文件。 |

### 说明

本函数返回文件中的数据块上次被写入的时间，也就是说，文件的内容上次被修改的时间。

## 提示和注释

提示：本函数的结果会被缓存。请使用 [clearstatcache()](http://www.w3school.com.cn/php/func_filesystem_clearstatcache.asp) 来清除缓存。

## 例子

<?php

echo filemtime("test.txt");

echo "Last modified: ".date("F d Y H:i:s.",filemtime("test.txt"));

?>

输出：

1139919766

Last modified: February 14 2006 13:22:46.

# fileowner() 函数

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

## 定义和用法

fileowner() 函数返回文件的所有者。

若成功，则返回文件所有的用户 ID。若失败，则返回 false。用户 ID 以数字格式返回。

### 语法

fileowner(filename)

|  |  |
| --- | --- |
| **参数** | **描述** |
| filename | 必需。规定要检查的文件。 |

## 提示和注释

提示：本函数的结果会被缓存。请使用 [clearstatcache()](http://www.w3school.com.cn/php/func_filesystem_clearstatcache.asp) 来清除缓存。

提示：用户 ID 以数字格式返回，请使用 posix\_getpwuid() 来把用户 ID 转换为用户名。

## 例子

<?php

echo fileowner("test.txt");

?>

# fileperms() 函数

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

## 定义和用法

fileperms() 函数返回文件或目录的权限。

若成功，则返回文件的访问权限。若失败，则返回 false。

### 语法

fileperms(filename)

|  |  |
| --- | --- |
| **参数** | **描述** |
| filename | 必需。规定要检查的文件。 |

## 提示和注释

提示：本函数的结果会被缓存。请使用 [clearstatcache()](http://www.w3school.com.cn/php/func_filesystem_clearstatcache.asp) 来清除缓存。

## 例子

### 例子 1

<?php

echo fileperms("test.txt");

?>

输出：

33206

### 例子 2

以八进制值返回权限：

<?php

echo substr(sprintf("%o",fileperms("test.txt")),-4);

?>

输出：

1777

# filesize() 函数

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

## 定义和用法

filesize() 函数返回指定文件的大小。

若成功，则返回文件大小的字节数。若失败，则返回 false 并生成一条 E\_WARNING 级的错误。

### 语法

filesize(filename)

|  |  |
| --- | --- |
| **参数** | **描述** |
| filename | 必需。规定要检查的文件。 |

## 提示和注释

提示：本函数的结果会被缓存。请使用 [clearstatcache()](http://www.w3school.com.cn/php/func_filesystem_clearstatcache.asp) 来清除缓存。

## 例子

<?php

echo filesize("test.txt");

?>

输出：

20

# filetype() 函数

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

## 定义和用法

filetype() 函数返回指定文件或目录的类型。

若成功，则返回 7 种可能的值。若失败，则返回 false。

可能的值：

* fifo
* char
* dir
* block
* link
* file
* unknown

### 语法

filetype(filename)

|  |  |
| --- | --- |
| **参数** | **描述** |
| filename | 必需。规定要检查的文件。 |

## 提示和注释

提示：本函数的结果会被缓存。请使用 [clearstatcache()](http://www.w3school.com.cn/php/func_filesystem_clearstatcache.asp) 来清除缓存。

## 例子

### 例子 1

<?php

echo filetype("test.txt");

?>

输出：

file

### 例子 2

<?php

echo filetype("images");

?>

输出：

dir

# flock() 函数

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

## 定义和用法

flock() 函数锁定或释放文件。

若成功，则返回 true。若失败，则返回 false。

### 语法

flock(file,lock,block)

|  |  |
| --- | --- |
| **参数** | **描述** |
| file | 必需。规定要锁定或释放的已打开的文件。 |
| lock | 必需。规定要使用哪种锁定类型。 |
| block | 可选。若设置为 1 或 true，则当进行锁定时阻挡其他进程。 |

### 说明

flock() 操作的 *file* 必须是一个已经打开的文件指针。

*lock* 参数可以是以下值之一：

* 要取得共享锁定（读取的程序），将 *lock* 设为 LOCK\_SH（PHP 4.0.1 以前的版本设置为 1）。
* 要取得独占锁定（写入的程序），将 *lock* 设为 LOCK\_EX（PHP 4.0.1 以前的版本中设置为 2）。
* 要释放锁定（无论共享或独占），将 *lock* 设为 LOCK\_UN（PHP 4.0.1 以前的版本中设置为 3）。
* 如果不希望 flock() 在锁定时堵塞，则给 *lock* 加上 LOCK\_NB（PHP 4.0.1 以前的版本中设置为 4）。

## 提示和注释

提示：可以通过 [fclose()](http://www.w3school.com.cn/php/func_filesystem_fclose.asp) 来释放锁定操作，代码执行完毕时也会自动调用。

注释：由于 flock() 需要一个文件指针， 因此可能不得不用一个特殊的锁定文件来保护打算通过写模式打开的文件的访问（在 fopen() 函数中加入 "w" 或 "w+"）。

## 例子

<?php

$file = fopen("test.txt","w+");

// 排它性的锁定

if (flock($file,LOCK\_EX))

{

fwrite($file,"Write something");

// release lock

flock($file,LOCK\_UN);

}

else

{

echo "Error locking file!";

}

fclose($file);

?>

# fnmatch() 函数

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

## 定义和用法

fnmatch() 函数根据指定的模式来匹配文件名或字符串。

### 语法

fnmatch(pattern,string,flags)

|  |  |
| --- | --- |
| **参数** | **描述** |
| pattern | 必需。规定要检索的模式。 |
| string | 必需。规定要检查的字符串或文件。 |
| flags | 可选。 |

### 说明

此函数对于文件名尤其有用，但也可以用于普通的字符串。普通用户可能习惯于 shell 模式或者至少其中最简单的形式 '?' 和 '\*' 通配符，因此使用 fnmatch() 来代替 ereg() 或者 preg\_match() 来进行前端搜索表达式输入对于非程序员用户更加方便。

## 提示和注释

重要事项：目前该函数无法在 Windows 或其它非 POSIX 兼容的系统上使用。

## 例子

根据 shell 通配符来检查颜色名：

<?php

$txt = "My car is darkgrey..."

if (fnmatch("\*gr[ae]y",$txt))

{

echo "some form of gray ...";

}

?>

# fopen() 函数

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

## 定义和用法

fopen() 函数打开文件或者 URL。

如果打开失败，本函数返回 FALSE。

### 语法

fopen(filename,mode,include\_path,context)

|  |  |
| --- | --- |
| **参数** | **描述** |
| filename | 必需。规定要打开的文件或 URL。 |
| mode | 必需。规定要求到该文件/流的访问类型。可能的值见下表。 |
| include\_path | 可选。如果也需要在 include\_path 中检索文件的话，可以将该参数设为 1 或 TRUE。 |
| context | 可选。规定文件句柄的环境。Context 是可以修改流的行为的一套选项。 |

### mode 参数的可能的值

|  |  |
| --- | --- |
| **mode** | **说明** |
| "r" | 只读方式打开，将文件指针指向文件头。 |
| "r+" | 读写方式打开，将文件指针指向文件头。 |
| "w" | 写入方式打开，将文件指针指向文件头并将文件大小截为零。如果文件不存在则尝试创建之。 |
| "w+" | 读写方式打开，将文件指针指向文件头并将文件大小截为零。如果文件不存在则尝试创建之。 |
| "a" | 写入方式打开，将文件指针指向文件末尾。如果文件不存在则尝试创建之。 |
| "a+" | 读写方式打开，将文件指针指向文件末尾。如果文件不存在则尝试创建之。 |
| "x" | 创建并以写入方式打开，将文件指针指向文件头。如果文件已存在，则 fopen() 调用失败并返回 FALSE，并生成一条 E\_WARNING 级别的错误信息。如果文件不存在则尝试创建之。  这和给底层的 open(2) 系统调用指定 O\_EXCL|O\_CREAT 标记是等价的。  此选项被 PHP 4.3.2 以及以后的版本所支持，仅能用于本地文件。 |
| "x+" | 创建并以读写方式打开，将文件指针指向文件头。如果文件已存在，则 fopen() 调用失败并返回 FALSE，并生成一条 E\_WARNING 级别的错误信息。如果文件不存在则尝试创建之。  这和给底层的 open(2) 系统调用指定 O\_EXCL|O\_CREAT 标记是等价的。  此选项被 PHP 4.3.2 以及以后的版本所支持，仅能用于本地文件。 |

### 说明

fopen() 将 *filename* 指定的名字资源绑定到一个流上。如果 *filename* 是 "*scheme://...*" 的格式，则被当成一个 URL，PHP 将搜索协议处理器（也被称为封装协议）来处理此模式。如果该协议尚未注册封装协议，PHP 将发出一条消息来帮助检查脚本中潜在的问题并将 *filename* 当成一个普通的文件名继续执行下去。

如果 PHP 认为 *filename* 指定的是一个本地文件，将尝试在该文件上打开一个流。该文件必须是 PHP 可以访问的，因此需要确认文件访问权限允许该访问。如果激活了安全模式或者 open\_basedir 则会应用进一步的限制。

如果 PHP 认为 *filename* 指定的是一个已注册的协议，而该协议被注册为一个网络 URL，PHP 将检查并确认 allow\_url\_fopen 已被激活。如果关闭了，PHP 将发出一个警告，而 fopen 的调用则失败。

对 *context* 的支持是 PHP 5.0.0 添加的。

## 提示和注释

注释：不同的操作系统家族具有不同的行结束习惯。当写入一个文本文件并想插入一个新行时，需要使用符合操作系统的行结束符号。基于 Unix 的系统使用 \n 作为行结束字符，基于 Windows 的系统使用 \r\n 作为行结束字符，基于 Macintosh 的系统使用 \r 作为行结束字符。 如果写入文件时使用了错误的行结束符号，则其它应用程序打开这些文件时可能会表现得很怪异。

Windows 下提供了一个文本转换标记（"t"）可以透明地将 \n 转换为 \r\n。与此对应还可以使用 "b" 来强制使用二进制模式，这样就不会转换数据。要使用这些标记，要么用 "b" 或者用 "t" 作为 mode 参数的最后一个字符。

默认的转换模式依赖于 SAPI 和所使用的 PHP 版本，因此为了便于移植鼓励总是指定恰当的标记。如果是操作纯文本文件并在脚本中使用了 \n 作为行结束符，但还要期望这些文件可以被其它应用程序例如 Notepad 读取，则在 mode 中使用 "t"。在所有其它情况下使用 "b"。

在操作二进制文件时如果没有指定 "b" 标记，可能会碰到一些奇怪的问题，包括坏掉的图片文件以及关于 \r\n 字符的奇怪问题。

注释：为移植性考虑，强烈建议在用 fopen() 打开文件时总是使用 "b" 标记。

注释：再一次，为移植性考虑，强烈建议你重写那些依赖于 "t" 模式的代码使其使用正确的行结束符并改成 "b" 模式。

## 例子

<?php

$file = fopen("test.txt","r");

$file = fopen("/home/test/test.txt","r");

$file = fopen("/home/test/test.gif","wb");

$file = fopen("http://www.example.com/","r");

$file = fopen("ftp://user:password@example.com/test.txt","w");

?>

# fpassthru() 函数

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

## 定义和用法

fpassthru() 函数输出文件指针处的所有剩余数据。

该函数将给定的文件指针从当前的位置读取到 EOF，并把结果写到输出缓冲区。

### 语法

fpassthru(file)

|  |  |
| --- | --- |
| **参数** | **描述** |
| file | 必需。规定要读取的打开文件或资源。 |

### 说明

如果发生错误， fpassthru() 返回 false。否则 fpassthru() 返回从 *file* 读取并传递到输出的字符数目。

文件指针必须有效，并且必须指向一个由 [fopen()](http://www.w3school.com.cn/php/func_filesystem_fopen.asp) 或 fsockopen() 成功打开（但还没有被 [fclose()](http://www.w3school.com.cn/php/func_filesystem_fclose.asp) 关闭）的文件。

## 提示和注释

提示：如果已经向文件写入数据，就必须调用 [rewind()](http://www.w3school.com.cn/php/func_filesystem_rewind.asp) 来将文件指针指向文件头。

提示：如果既不修改文件也不在特定位置检索，只想将文件的内容下载到输出缓冲区，应该使用 [readfile()](http://www.w3school.com.cn/php/func_filesystem_readfile.asp)，这样可以省去 fopen() 调用。

注释：当在 Windows 系统中将 fpassthru() 用于二进制文件时，要确保在用 fopen() 打开文件时在 mode 中附加了 b 来将文件以二进制方式打开。鼓励在处理二进制文件时使用 b 标志，即使系统并不需要，这样可以使脚本的移植性更好。

## 例子

### 例子 1

<?php

$file = fopen("test.txt","r");

// 读取第一行

fgets($file);

// 把文件的其余部分发送到输出缓存

echo fpassthru($file);

fclose($file);

?>

输出：

There are three lines in this file.

This is the last line.59

注：59 指示被传递的字符数。

### 例子 2

转储 www 服务器的 index 页：

<?php

$file = fopen("http://www.example.com","r");

fpassthru($file);

?>

# fputcsv() 函数

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

## 定义和用法

fputcsv() 函数将行格式化为 CSV 并写入一个打开的文件。

该函数返回写入字符串的长度。若出错，则返回 false。。

### 语法

fputcsv(file,fields,seperator,enclosure)

|  |  |
| --- | --- |
| **参数** | **描述** |
| file | 必需。规定要写入的打开文件。 |
| fields | 必需。规定要从中获得数据的数组。 |
| seperator | 可选。规定字段分隔符的字符。默认是逗号 (,)。 |
| enclosure | 可选。规定字段环绕符的字符。默认是双引号 "。 |

### 说明

fputcsv() 将一行（用 *fields* 数组传递）格式化为 CSV 格式并写入由 *file* 指定的文件。

## 提示和注释

提示：参见 fgetcsv() 函数。

## 例子

<?php

$list = array

(

"George,John,Thomas,USA",

"James,Adrew,Martin,USA",

);

$file = fopen("contacts.csv","w");

foreach ($list as $line)

{

fputcsv($file,split(',',$line));

}

fclose($file);

?>

以上代码执行后，CSV 文件会类似这样：

George,John,Thomas,USA

James,Adrew,Martin,USA

# fputs() 函数

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

## 定义和用法

fputs() 函数写入文件（可安全用于二进制文件）。

fputs() 函数是 [fwrite()](http://www.w3school.com.cn/php/func_filesystem_fwrite.asp) 函数的别名。

### 语法

fputs(file,string,length)

|  |  |
| --- | --- |
| **参数** | **描述** |
| file | 必需。规定要写入的打开文件。 |
| string | 必需。规定要写入文件的字符串。 |
| length | 可选。规定要写入的最大字节数。 |

### 说明

fwrite() 把 *string* 的内容写入文件指针 *file* 处。 如果指定了 *length*，当写入了 *length* 个字节或者写完了 *string* 以后，写入就会停止，视乎先碰到哪种情况。

fwrite() 返回写入的字符数，出现错误时则返回 false。

## 例子

<?php

$file = fopen("test.txt","w");

echo fputs($file,"Hello World. Testing!");

fclose($file);

?>

输出：

21

# fread() 函数

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

## 定义和用法

fread() 函数读取文件（可安全用于二进制文件）。

### 语法

fread(file,length)

|  |  |
| --- | --- |
| **参数** | **描述** |
| file | 必需。规定要读取打开文件。 |
| length | 必需。规定要读取的最大字节数。 |

### 说明

fread() 从文件指针 *file* 读取最多 *length* 个字节。该函数在读取完最多 *length* 个字节数，或到达 EOF 的时候，或（对于网络流）当一个包可用时，或（在打开用户空间流之后）已读取了 8192 个字节时就会停止读取文件，视乎先碰到哪种情况。

返回所读取的字符串，如果出错返回 false。

## 提示和注释

提示：如果只是想将一个文件的内容读入到一个字符串中，请使用 [file\_get\_contents()](http://www.w3school.com.cn/php/func_filesystem_file_get_contents.asp)，它的性能比 fread() 好得多。

## 例子

### 例子 1

从文件中读取 10 个字节：

<?php

$file = fopen("test.txt","r");

fread($file,"10");

fclose($file);

?>

### 例子 2

读取整个文件：

<?php

$file = fopen("test.txt","r");

fread($file,filesize("test.txt"));

fclose($file);

?>

# fscanf() 函数

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

## 定义和用法

fscanf() 函数根据指定的格式对来自打开的文件的输入进行解析。

### 语法

fscanf(file,format,mixed)

|  |  |
| --- | --- |
| **参数** | **描述** |
| file | 必需。规定要检查的文件。 |
| format | 必需。规定格式。 |
| mixed | 可选。 |

### 说明

fscanf() 函数与 sscanf() 相似，但是它从与 *file* 关联的文件中接受输入并根据指定的 *format* 来解释输入。如果只给此函数传递了两个参数，解析后的值会被作为数组返回。否则，如果提供了可选参数，此函数将返回被赋值的数目。可选参数必须用引用传递。

## 提示和注释

注释：格式字符串中的任何空白会与输入流中的任何空白匹配。这意味着甚至格式字符串中的制表符 \t 也会与输入流中的一个空格字符匹配。

注释：在 PHP 4.3.0 之前，从文件中读入的最大字符数是 512（或者第一个 \n，看先碰到哪种情况）。从 PHP 4.3.0 起可以读取任意长的行。

# fseek() 函数

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

## 定义和用法

fseek() 函数在打开的文件中定位。

该函数把文件指针从当前位置向前或向后移动到新的位置，新位置从文件头开始以字节数度量。

成功则返回 0；否则返回 -1。注意，移动到 EOF 之后的位置不会产生错误。

### 语法

fseek(file,offset,whence)

|  |  |
| --- | --- |
| **参数** | **描述** |
| file | 必需。规定要在其中定位的文件。 |
| offset | 必需。规定新的位置（从文件头开始以字节数度量）。 |
| whence | 可选。可能的值：   * SEEK\_SET - 设定位置等于 *offset* 字节。默认。 * SEEK\_CUR - 设定位置为当前位置加上 *offset*。 * SEEK\_END - 设定位置为文件末尾加上 *offset* （要移动到文件尾之前的位置，*offset* 必须是一个负值）。 |

### 说明

*whence* 参数是 PHP 4.0.0 之后增加的。

## 提示和注释

提示：通过使用 [ftell()](http://www.w3school.com.cn/php/func_filesystem_ftell.asp) 来找到当前位置。

## 例子

<?php

$file = fopen("test.txt","r");

// 读取第一行

fgets($file);

// 倒回文件的开头

fseek($file,0);

?>

# fstat() 函数

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

## 定义和用法

fstat() 函数返回关于打开文件的信息。

### 语法

fstat(file)

|  |  |
| --- | --- |
| **参数** | **描述** |
| pipe | 必需。规定要检查的打开文件。 |

### 说明

获取由文件指针 handle 所打开文件的统计信息。

该函数返回的数组具有该文件的统计信息，该数组包含以下元素：

|  |  |  |
| --- | --- | --- |
| **数字下标** | **关联键名（自 PHP 4.0.6）** | **说明** |
| 0 | dev | 设备名 |
| 1 | ino | 号码 |
| 2 | mode | inode 保护模式 |
| 3 | nlink | 被连接数目 |
| 4 | uid | 所有者的用户 id |
| 5 | gid | 所有者的组 id |
| 6 | rdev | 设备类型，如果是 inode 设备的话 |
| 7 | size | 文件大小的字节数 |
| 8 | atime | 上次访问时间（Unix 时间戳） |
| 9 | mtime | 上次修改时间（Unix 时间戳） |
| 10 | ctime | 上次改变时间（Unix 时间戳） |
| 11 | blksize | 文件系统 IO 的块大小 |
| 12 | blocks | 所占据块的数目 |

## 提示和注释

提示：本函数与 [stat()](http://www.w3school.com.cn/php/func_filesystem_stat.asp) 函数相似，不同的是，它是作用于已打开的文件指针而不是文件名。

## 例子

<?php

$file = fopen("test.txt","r");

print\_r(fstat($file));

fclose($file);

?>

输出类似：

Array

(

[0] => 0

[1] => 0

[2] => 33206

[3] => 1

[4] => 0

[5] => 0

[6] => 0

[7] => 92

[8] => 1141633430

[9] => 1141298003

[10] => 1138609592

[11] => -1

[12] => -1

[dev] => 0

[ino] => 0

[mode] => 33206

[nlink] => 1

[uid] => 0

[gid] => 0

[rdev] => 0

[size] => 92

[atime] => 1141633430

[mtime] => 1141298003

[ctime] => 1138609592

[blksize] => -1

[blocks] => -1

)

# ftell() 函数

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

## 定义和用法

ftell() 函数在打开文件中的当前位置。

该函数返回文件指针的当前位置。若失败，则返回 false。

### 语法

ftell(file)

|  |  |
| --- | --- |
| **参数** | **描述** |
| file | 必需。规定要检查的已打开文件。 |

### 说明

文件指针 *file* 必须是有效的，且必须指向一个通过 [fopen()](http://www.w3school.com.cn/php/func_filesystem_fopen.asp) 或 [popen()](http://www.w3school.com.cn/php/func_filesystem_popen.asp) 成功打开的文件。

在附加模式（加参数 "a" 打开文件）中 ftell() 会返回未定义错误。

## 例子

<?php

$file = fopen("test.txt","r");

// 输出当前位置

echo ftell($file);

// 改变当前位置

fseek($file,"15");

// 再次输出当前位置

echo ftell($file);

fclose($file);

?>

输出：

0

15

# ftruncate() 函数

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

## 定义和用法

ftruncate() 函数把文件截断到指定的长度。

### 语法

ftruncate(file,size)

|  |  |
| --- | --- |
| **参数** | **描述** |
| file | 必需。规定要截断的打开文件。 |
| size | 必需。规定新的文件大小。 |

### 说明

接受文件指针 *file* 作为参数，并将文件大小截取为 *size*。如果成功则返回 TRUE，否则返回 FALSE。

## 提示和注释

注释：文件只会在 append 模式下改变。在 write 模式下，必须加上 [fseek()](http://www.w3school.com.cn/php/func_filesystem_fseek.asp) 操作。

注释：在 PHP 4.3.3 之前，ftruncate() 在成功时返回一个整数值 1，而不是布尔值的 TRUE。

## 例子

<?php

//检查文件大小

echo filesize("test.txt");

echo "<br />";

$file = fopen("test.txt", "a+");

ftruncate($file,100);

fclose($file);

//清空缓存，再次检查文件大小

clearstatcache();

echo filesize("test.txt");

?>

输出类似：

792

100

# fwrite() 函数

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

## 定义和用法

fwrite() 函数写入文件（可安全用于二进制文件）。

### 语法

fwrite(file,string,length)

|  |  |
| --- | --- |
| **参数** | **描述** |
| file | 必需。规定要写入的打开文件。 |
| string | 必需。规定要写入文件的字符串。 |
| length | 可选。规定要写入的最大字节数。 |

### 说明

fwrite() 把 *string* 的内容写入文件指针 *file* 处。 如果指定了 *length*，当写入了 *length* 个字节或者写完了 *string* 以后，写入就会停止，视乎先碰到哪种情况。

fwrite() 返回写入的字符数，出现错误时则返回 false。

## 例子

<?php

$file = fopen("test.txt","w");

echo fwrite($file,"Hello World. Testing!");

fclose($file);

?>

输出：

21

# glob() 函数

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

## 定义和用法

glob() 函数返回匹配指定模式的文件名或目录。

该函数返回一个包含有匹配文件 / 目录的数组。如果出错返回 false。

### 语法

glob(pattern,flags)

|  |  |
| --- | --- |
| **参数** | **描述** |
| file | 必需。规定检索模式。 |
| size | 可选。规定特殊的设定。   * GLOB\_MARK - 在每个返回的项目中加一个斜线 * GLOB\_NOSORT - 按照文件在目录中出现的原始顺序返回（不排序） * GLOB\_NOCHECK - 如果没有文件匹配则返回用于搜索的模式 * GLOB\_NOESCAPE - 反斜线不转义元字符 * GLOB\_BRACE - 扩充 {a,b,c} 来匹配 'a'，'b' 或 'c' * GLOB\_ONLYDIR - 仅返回与模式匹配的目录项 * GLOB\_ERR - 停止并读取错误信息（比如说不可读的目录），默认的情况下忽略所有错误   注释：GLOB\_ERR 是 PHP 5.1 添加的。 |

## 例子

### 例子 1

<?php

print\_r(glob("\*.txt"));

?>

输出类似：

Array

(

[0] => target.txt

[1] => source.txt

[2] => test.txt

[3] => test2.txt

)

### 例子 2

<?php

print\_r(glob("\*.\*"));

?>

输出类似：

Array

(

[0] => contacts.csv

[1] => default.php

[2] => target.txt

[3] => source.txt

[4] => tem1.tmp

[5] => test.htm

[6] => test.ini

[7] => test.php

[8] => test.txt

[9] => test2.txt

)

# is\_dir() 函数

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

## 定义和用法

is\_dir() 函数检查指定的文件是否是目录。

### 语法

is\_dir(file)

|  |  |
| --- | --- |
| **参数** | **描述** |
| file | 必需。规定要检查的文件。 |

### 说明

如果文件名存在并且为目录，则返回 true。如果 *file* 是一个相对路径，则按照当前工作目录检查其相对路径。

## 提示和注释

注释：本函数的结果会被缓存。请使用 [clearstatcache()](http://www.w3school.com.cn/php/func_filesystem_clearstatcache.asp) 来清除缓存。

## 例子

<?php

$file = "images";

if(is\_dir($file))

{

echo ("$file is a directory");

}

else

{

echo ("$file is not a directory");

}

?>

输出：

images is a directory

# is\_executable() 函数

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

## 定义和用法

is\_executable() 函数检查指定的文件是否可执行。

### 语法

is\_executable(file)

|  |  |
| --- | --- |
| **参数** | **描述** |
| file | 必需。规定要检查的文件。 |

### 说明

如果文件存在且可执行，则返回 true。

## 提示和注释

注释：本函数的结果会被缓存。请使用 [clearstatcache()](http://www.w3school.com.cn/php/func_filesystem_clearstatcache.asp) 来清除缓存。

注释：is\_executable() 自 PHP 5.0.0 版起可用于 Windows。

## 例子

<?php

$file = "setup.exe";

if(is\_executable($file))

{

echo ("$file is executable");

}

else

{

echo ("$file is not executable");

}

?>

输出：

setup.exe is executable

# is\_file() 函数

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

## 定义和用法

is\_file() 函数检查指定的文件名是否是正常的文件。

### 语法

is\_file(file)

|  |  |
| --- | --- |
| **参数** | **描述** |
| file | 必需。规定要检查的文件。 |

### 说明

如果文件存在且为正常的文件，则返回 true。

## 提示和注释

注释：本函数的结果会被缓存。请使用 [clearstatcache()](http://www.w3school.com.cn/php/func_filesystem_clearstatcache.asp) 来清除缓存。

## 例子

### 例子 1

<?php

$file = "test.txt";

if(is\_file($file))

{

echo ("$file is a regular file");

}

else

{

echo ("$file is not a regular file");

}

?>

输出：

test.txt is a regular file

### 例子 2

<?php

var\_dump(is\_file('a\_file.txt')) . "\n";

var\_dump(is\_file('/usr/bin/')) . "\n";

?>

输出：

bool(true)

bool(false)

# is\_link() 函数

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

## 定义和用法

is\_link() 函数判断指定文件名是否为一个符号连接。

### 语法

is\_link(file)

|  |  |
| --- | --- |
| **参数** | **描述** |
| file | 必需。规定要检查的文件。 |

### 说明

如果文件存在并且是一个符号连接，则返回 true。

## 提示和注释

注释：本函数的结果会被缓存。请使用 [clearstatcache()](http://www.w3school.com.cn/php/func_filesystem_clearstatcache.asp) 来清除缓存。

## 例子

<?php

$link = "images";

if(is\_link($link))

{

echo ("$link is a link");

}

else

{

echo ("$link is not a link");

}

?>

输出：

images is not a link

# is\_readable() 函数

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

## 定义和用法

is\_readable() 函数判断指定文件名是否可读。

### 语法

is\_readable(file)

|  |  |
| --- | --- |
| **参数** | **描述** |
| file | 必需。规定要检查的文件。 |

### 说明

如果由 *file* 指定的文件或目录存在并且可读，则返回 TRUE。

## 提示和注释

注释：本函数的结果会被缓存。请使用 [clearstatcache()](http://www.w3school.com.cn/php/func_filesystem_clearstatcache.asp) 来清除缓存。

## 例子

<?php

$file = "test.txt";

if(is\_readable($file))

{

echo ("$file is readable");

}

else

{

echo ("$file is not readable");

}

?>

输出：

test.txt is readable

# is\_uploaded\_file() 函数

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

## 定义和用法

is\_uploaded\_file() 函数判断指定的文件是否是通过 HTTP POST 上传的。

### 语法

is\_uploaded\_file(file)

|  |  |
| --- | --- |
| **参数** | **描述** |
| file | 必需。规定要检查的文件。 |

### 说明

如果 *file* 所给出的文件是通过 HTTP POST 上传的则返回 TRUE。

该函数可以用于确保恶意的用户无法欺骗脚本去访问本不能访问的文件，例如 /etc/passwd。

这种检查显得格外重要，如果上传的文件有可能会造成对用户或本系统的其他用户显示其内容的话。

## 提示和注释

注释：本函数的结果会被缓存。请使用 [clearstatcache()](http://www.w3school.com.cn/php/func_filesystem_clearstatcache.asp) 来清除缓存。

## 例子

<?php

$file = "test.txt";

if(is\_uploaded\_file($file))

{

echo ("$file is uploaded via HTTP POST");

}

else

{

echo ("$file is not uploaded via HTTP POST");

}

?>

输出：

test.txt is not uploaded via HTTP POST

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

窗体顶端

Search:

# is\_writable() 函数

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

## 定义和用法

is\_writable() 函数判断指定的文件是否可写。

### 语法

is\_writable(file)

|  |  |
| --- | --- |
| **参数** | **描述** |
| file | 必需。规定要检查的文件。 |

### 说明

如果文件存在并且可写则返回 true。*file* 参数可以是一个允许进行是否可写检查的目录名。

## 提示和注释

注释：本函数的结果会被缓存。请使用 [clearstatcache()](http://www.w3school.com.cn/php/func_filesystem_clearstatcache.asp) 来清除缓存。

## 例子

<?php

$file = "test.txt";

if(is\_writable($file))

{

echo ("$file is writeable");

}

else

{

echo ("$file is not writeable");

}

?>

输出：

test.txt is writeable

# is\_writeable() 函数

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

## 定义和用法

is\_writeable() 函数判断指定的文件是否可写。

该函数是 [is\_writable()](http://www.w3school.com.cn/php/func_filesystem_is_writable.asp) 函数的别名。

### 语法

is\_writeable(file)

|  |  |
| --- | --- |
| **参数** | **描述** |
| file | 必需。规定要检查的文件。 |

### 说明

如果文件存在并且可写则返回 true。*file* 参数可以是一个允许进行是否可写检查的目录名。

## 提示和注释

注释：本函数的结果会被缓存。请使用 [clearstatcache()](http://www.w3school.com.cn/php/func_filesystem_clearstatcache.asp) 来清除缓存。

## 例子

<?php

$file = "test.txt";

if(is\_writeable($file))

{

echo ("$file is writeable");

}

else

{

echo ("$file is not writeable");

}

?>

输出：

test.txt is writeable

# link() 函数

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

## 定义和用法

link() 函数建立一个硬连接。

如果成功，则返回 true，失败则返回 false。

提示：创建的连接不是 HTML 链接，而是文件系统中的连接。

### 语法

link(target,link)

|  |  |
| --- | --- |
| **参数** | **描述** |
| target | 必需。 |
| link | 必需。 |

## 提示和注释

注释：本函数不能作用于远程文件，被检查的文件必须通过服务器的文件系统访问。

注释：本函数不能工作在 windows 平台上。

# linkinfo() 函数

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

## 定义和用法

linkinfo() 函数返回连接的信息。

本函数返回设备 ID。若出错，则返回 0 或 FALSE。

### 语法

linkinfo(path)

|  |  |
| --- | --- |
| **参数** | **描述** |
| path | 必需。规定要检查的路径 |

## 提示和注释

注释：本函数不能工作在 windows 平台上。

# lstat() 函数

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

## 定义和用法

lstat() 函数返回关于文件或符号连接的信息。

### 语法

lstat(file)

|  |  |
| --- | --- |
| **参数** | **描述** |
| file | 必需。规定要检查的文件。 |

### 说明

获取由 *file* 参数指定的文件或符号连接的统计信息。

### lstat() 的返回格式

|  |  |  |
| --- | --- | --- |
| **数字下标** | **关联键名（自 PHP 4.0.6）** | **说明** |
| 0 | dev | 设备名 |
| 1 | ino | 号码 |
| 2 | mode | inode 保护模式 |
| 3 | nlink | 被连接数目 |
| 4 | uid | 所有者的用户 id |
| 5 | gid | 所有者的组 id |
| 6 | rdev | 设备类型，如果是 inode 设备的话 |
| 7 | size | 文件大小的字节数 |
| 8 | atime | 上次访问时间（Unix 时间戳） |
| 9 | mtime | 上次修改时间（Unix 时间戳） |
| 10 | ctime | 上次改变时间（Unix 时间戳） |
| 11 | blksize | 文件系统 IO 的块大小 |
| 12 | blocks | 所占据块的数目 |

## 提示和注释

提示：本函数与 [stat()](http://www.w3school.com.cn/php/func_filesystem_stat.asp) 函数相同，不同之处只有一点：如果 *file* 参数是符号连接的话，则该符号连接的状态被返回，而不是该符号连接所指向的文件的状态。

注释：本函数的结果会被缓存。请使用 [clearstatcache()](http://www.w3school.com.cn/php/func_filesystem_clearstatcache.asp) 来清除缓存。

## 例子

<?php

print\_r(lstat("test.txt"));

?>

输出类似：

Array

(

[0] => 0

[1] => 0

[2] => 33206

[3] => 1

[4] => 0

[5] => 0

[6] => 0

[7] => 92

[8] => 1141633430

[9] => 1141298003

[10] => 1138609592

[11] => -1

[12] => -1

[dev] => 0

[ino] => 0

[mode] => 33206

[nlink] => 1

[uid] => 0

[gid] => 0

[rdev] => 0

[size] => 92

[atime] => 1141633430

[mtime] => 1141298003

[ctime] => 1138609592

[blksize] => -1

[blocks] => -1

)

# mkdir() 函数

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

## 定义和用法

mkdir() 函数创建目录。

若成功，则返回 true，否则返回 false。

### 语法

mkdir(path,mode,recursive,context)

|  |  |
| --- | --- |
| **参数** | **描述** |
| path | 必需。规定要创建的目录的名称。 |
| mode | 必需。规定权限。默认是 0777。 |
| recursive | 必需。规定是否设置递归模式。 |
| context | 必需。规定文件句柄的环境。Context 是可修改流的行为的一套选项。 |

### 说明

mkdir() 尝试新建一个由 *path* 指定的目录。

默认的 *mode* 是 0777，意味着最大可能的访问权。

## 提示和注释

注释：*mode* 在 Windows 下被忽略。自 PHP 4.2.0 起成为可选项。

注释：对 *context* 的支持是 PHP 5.0.0 添加的。

注释：*recursive* 参数是 PHP 5.0.0 添加的。

## 例子

<?php

mkdir("testing");

?>

# move\_uploaded\_file() 函数

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

## 定义和用法

move\_uploaded\_file() 函数将上传的文件移动到新位置。

若成功，则返回 true，否则返回 false。

### 语法

move\_uploaded\_file(file,newloc)

|  |  |
| --- | --- |
| **参数** | **描述** |
| file | 必需。规定要移动的文件。 |
| newloc | 必需。规定文件的新位置。 |

### 说明

本函数检查并确保由 *file* 指定的文件是合法的上传文件（即通过 PHP 的 HTTP POST 上传机制所上传的）。如果文件合法，则将其移动为由 *newloc* 指定的文件。

如果 *file* 不是合法的上传文件，不会出现任何操作，move\_uploaded\_file() 将返回 false。

如果 *file* 是合法的上传文件，但出于某些原因无法移动，不会出现任何操作，move\_uploaded\_file() 将返回 false，此外还会发出一条警告。

这种检查显得格外重要，如果上传的文件有可能会造成对用户或本系统的其他用户显示其内容的话。

## 提示和注释

注释：本函数仅用于通过 HTTP POST 上传的文件。

注意：如果目标文件已经存在，将会被覆盖。

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

# parse\_ini\_file() 函数

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

## 定义和用法

parse\_ini\_file() 函数解析一个配置文件，并以数组的形式返回其中的设置。

### 语法

parse\_ini\_file(file,process\_sections)

|  |  |
| --- | --- |
| **参数** | **描述** |
| file | 必需。规定要检查的 ini 文件。 |
| process\_sections | 可选。如果设置为 true，则返回一个多维数组，包括了配置文件中每一节的名称和设置。默认是 false。 |

### 说明

ini 文件的结构和 php.ini 的相似。

常量也可以在 ini 文件中被解析，因此如果在运行 parse\_ini\_file() 之前定义了常量作为 ini 的值，将会被集成到结果中去。只有 ini 的值会被求值。

由数字组成的键名和小节名会被 PHP 当作整数来处理，因此以 0 开头的数字会被当作八进制而以 0x 开头的会被当作十六进制。

## 提示和注释

注释：本函数可以用来读取你自己的应用程序的配置文件。本函数与 php.ini 文件没有关系，该文件在运行脚本时就已经处理过了。

注释：如果 ini 文件中的值包含任何非字母数字的字符，需要将其括在双引号中（"）。

注释：有些保留字不能作为 ini 文件中的键名，包括：null，yes，no，true 和 false。值为 null，no 和 false 等效于 ""，值为 yes 和 true 等效于 "1"。字符 {}|"~![()" 也不能用在键名的任何地方，而且这些字符在选项值中有着特殊的意义。

注释：自 PHP 5.0 版本开始，该函数也处理选项值内的新行。

## 例子

### 例子 1

"test.ini" 的内容：

[names]

me = Robert

you = Peter

[urls]

first = "http://www.example.com"

second = "http://www.w3school.com.cn"

PHP 代码：

<?php

print\_r(parse\_ini\_file("test.ini"));

?>

输出：

Array

(

[me] => Robert

[you] => Peter

[first] => http://www.example.com

[second] => http://www.w3school.com.cn

)

### 例子 2

"test.ini" 的内容：

[names]

me = Robert

you = Peter

[urls]

first = "http://www.example.com"

second = "http://www.w3school.com.cn"

PHP 代码（*process\_sections* 设置为 true）：

<?php

print\_r(parse\_ini\_file("test.ini",true));

?>

输出：

Array

(

[names] => Array

(

[me] => Robert

[you] => Peter

)

[urls] => Array

(

[first] => http://www.example.com

[second] => http://www.w3school.com.cn

)

)

# pathinfo() 函数

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

## 定义和用法

pathinfo() 函数以数组的形式返回文件路径的信息。

### 语法

pathinfo(path,options)

|  |  |
| --- | --- |
| **参数** | **描述** |
| path | 必需。规定要检查的路径。 |
| process\_sections | 可选。规定要返回的数组元素。默认是 all。  可能的值：   * PATHINFO\_DIRNAME - 只返回 dirname * PATHINFO\_BASENAME - 只返回 basename * PATHINFO\_EXTENSION - 只返回 extension |

### 说明

pathinfo() 返回一个关联数组包含有 *path* 的信息。

包括以下的数组元素：

* [dirname]
* [basename]
* [extension]

## 提示和注释

注释：如果不是要求取得所有单元，则 pathinfo() 函数返回字符串。

## 例子

### 例子 1

<?php

print\_r(pathinfo("/testweb/test.txt"));

?>

输出：

Array

(

[dirname] => /testweb

[basename] => test.txt

[extension] => txt

)

### 例子 2

<?php

print\_r(pathinfo("/testweb/test.txt",PATHINFO\_BASENAME));

?>

输出：

test.txt

# pclose() 函数

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

## 定义和用法

pclose() 函数关闭由 popen() 打开的管道。

### 语法

pclose(pipe)

|  |  |
| --- | --- |
| **参数** | **描述** |
| pipe | 必需。规定由 popen() 打开的管道。 |

### 说明

该函数返回运行的进程的终止状态。

若出错，则返回 false。

## 例子

<?php

$file = popen("/bin/ls","r");

//一些要执行的代码

pclose($file);

?>

# popen() 函数

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

## 定义和用法

popen() 函数打开进程文件指针。

### 语法

popen(command,mode)

|  |  |
| --- | --- |
| **参数** | **描述** |
| command | 必需。规定要执行的命令。 |
| mode | 必需。规定连接模式。 可能的值：   * r: 只读。 * w: 只写 (打开并清空已有文件或创建一个新文件) |

### 说明

打开一个指向进程的管道，该进程由派生指定的 *command* 命令执行而产生。

返回一个和 [fopen()](http://www.w3school.com.cn/php/func_filesystem_fopen.asp) 所返回的相同的文件指针，只不过它是单向的（只能用于读或写）并且必须用 [pclose()](http://www.w3school.com.cn/php/func_filesystem_pclose.asp) 来关闭。此指针可以用于 [fgets()](http://www.w3school.com.cn/php/func_filesystem_fgets.asp)，[fgetss()](http://www.w3school.com.cn/php/func_filesystem_fgetss.asp) 和 [fwrite()](http://www.w3school.com.cn/php/func_filesystem_fwrite.asp)。

若出错，则返回 false。

## 例子

<?php

$file = popen("/bin/ls","r");

//一些要执行的代码

pclose($file);

?>

# readfile() 函数

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

## 定义和用法

readfile() 函数输出一个文件。

该函数读入一个文件并写入到输出缓冲。

若成功，则返回从文件中读入的字节数。若失败，则返回 false。您可以通过 @readfile() 形式调用该函数，来隐藏错误信息。

### 语法

readfile(filename,include\_path,context)

|  |  |
| --- | --- |
| **参数** | **描述** |
| filename | 必需。规定要读取的文件。 |
| include\_path | 必需。如果也想在 include\_path 中搜索文件，可以使用该参数并将其设为 true。 |
| context | 必需。规定文件句柄的环境。Context 是可以修改流的行为的一套选项。 |

### 说明

对 *context* 参数的支持是 PHP 5.0.0 添加的。

## 提示和注释

提示：如果在 php.ini 文件中 "fopen wrappers" 已经被激活，则在本函数中可以把 URL 作为文件名来使用。

## 例子

<?php

echo readfile("test.txt");

?>

输出：

There are two lines in this file.

This is the last line.

57

# readlink() 函数

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

## 定义和用法

readlink() 函数返回符号连接指向的目标。

若成功，则该函数返回连接的目标。若失败，则返回 false。

### 语法

readlink(linkpath)

|  |  |
| --- | --- |
| **参数** | **描述** |
| linkpath | 必需。规定要检查的连接路径。 |

## 提示和注释

注释：本函数未在 Windows 平台下实现。。

## 例子

<?php

echo readlink("/user/testlink");

?>

# realpath() 函数

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

## 定义和用法

realpath() 函数返回绝对路径。

该函数删除所有符号连接（比如 '/./', '/../' 以及多余的 '/'），返回绝对路径名。

若失败，则返回 false。比如说文件不存在的话。

### 语法

readlink(linkpath)

|  |  |
| --- | --- |
| **参数** | **描述** |
| linkpath | 必需。规定要检查的连接路径。 |

### 说明

在 BSD 系统上，如果仅仅是 *linkpath* 不存在的话，PHP 并不会像其它系统那样返回 false。

## 例子

<?php

echo realpath("test.txt");

?>

输出：

C:\Inetpub\testweb\test.txt

# rename() 函数

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

## 定义和用法

rename() 函数重命名文件或目录。

若成功，则该函数返回 true。若失败，则返回 false。

### 语法

rename(oldname,newname,context)

|  |  |
| --- | --- |
| **参数** | **描述** |
| oldname | 必需。规定要重命名的文件或目录。 |
| newname | 必需。规定文件或目录的新名称。 |
| context | 必需。规定文件句柄的环境。Context 是可修改流的行为的一套选项。 |

## 提示和注释

注释：在 PHP 4.3.3 之前，rename() 不能在基于 \*nix 的系统中跨磁盘分区重命名文件。

注释：用于 oldname 中的封装协议必须和用于 newname 中的相匹配。

注释：对 context 的支持是 PHP 5.0.0 添加的。

## 例子

<?php

rename("images","pictures");

?>

# rewind() 函数

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

## 定义和用法

rewind() 函数将文件指针的位置倒回文件的开头。

若成功，则返回 true。若失败，则返回 false。

### 语法

rewind(file)

|  |  |
| --- | --- |
| **参数** | **描述** |
| file | 必需。规定已打开的文件。 |

## 例子

<?php

$file = fopen("test.txt","r");

//改变文件指针的位置

fseek($file,"15");

//把文件指针设定为 0

rewind($file);

fclose($file);

?>

# rmdir() 函数

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

## 定义和用法

rmdir() 函数删除空的目录。

若成功，则该函数返回 true。若失败，则返回 false。

### 语法

rmdir(dir,context)

|  |  |
| --- | --- |
| **参数** | **描述** |
| dir | 必需。规定要删除的目录。 |
| context | 必需。规定文件句柄的环境。Context 是可修改流的行为的一套选项。 |

### 说明

尝试删除 *dir* 所指定的目录。 该目录必须是空的，而且要有相应的权限。

## 提示和注释

注释：对 *context* 的支持是 PHP 5.0.0 添加的。

## 例子

<?php

$path = "images";

if(!rmdir($path))

{

echo ("Could not remove $path");

}

?>

# set\_file\_buffer() 函数

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

## 定义和用法

set\_file\_buffer() 函数设置打开文件的缓冲大小。

若成功，则该函数返回 0。若失败，则返回 EOF。

### 语法

set\_file\_buffer(file,buffer)

|  |  |
| --- | --- |
| **参数** | **描述** |
| file | 必需。规定打开的文件。 |
| buffer | 必需。规定缓冲大小，以字节计。 |

## 提示和注释

注释：该函数是 stream\_set\_write\_buffer() 的别名。

## 例子

创建无缓冲的流：

<?php

$file = fopen("test.txt","w");

if ($file)

{

set\_file\_buffer($file,0);

fwrite($file,"Hello World. Testing!");

fclose($file);

}

?>

[PHP Files](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

# stat() 函数

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

## 定义和用法

stat() 函数返回关于文件的信息。

### 语法

fstat(file)

|  |  |
| --- | --- |
| **参数** | **描述** |
| file | 必需。规定要检查的文件。 |

### 说明

获取由 *file* 指定的文件的统计信息。如果 *file* 是符号连接，则统计信息是关于被连接文件本身的，而不是符号连接。

如果出错，stat() 返回 false，并且发出一条警告。

返回的数组包含有文件的统计信息，该数组具有以下列出的单元，数组下标从零开始。除了数字索引之外，从 PHP 4.0.6 起还可以通过关联索引来访问。

### stat() 的返回格式

|  |  |  |
| --- | --- | --- |
| **数字下标** | **关联键名（自 PHP 4.0.6）** | **说明** |
| 0 | dev | 设备名 |
| 1 | ino | 号码 |
| 2 | mode | inode 保护模式 |
| 3 | nlink | 被连接数目 |
| 4 | uid | 所有者的用户 id |
| 5 | gid | 所有者的组 id |
| 6 | rdev | 设备类型，如果是 inode 设备的话 |
| 7 | size | 文件大小的字节数 |
| 8 | atime | 上次访问时间（Unix 时间戳） |
| 9 | mtime | 上次修改时间（Unix 时间戳） |
| 10 | ctime | 上次改变时间（Unix 时间戳） |
| 11 | blksize | 文件系统 IO 的块大小 |
| 12 | blocks | 所占据块的数目 |

## 提示和注释

提示：[lstat()](http://www.w3school.com.cn/php/func_filesystem_lstat.asp) 与 stat() 类似，不同的是，它会返回符号连接的状态。

注释：本函数的结果会被缓存。请使用 [clearstatcache()](http://www.w3school.com.cn/php/func_filesystem_clearstatcache.asp) 来清除缓存。

## 例子

<?php

$file = fopen("test.txt","r");

print\_r(stat($file));

fclose($file);

?>

输出类似：

Array

(

[0] => 0

[1] => 0

[2] => 33206

[3] => 1

[4] => 0

[5] => 0

[6] => 0

[7] => 92

[8] => 1141633430

[9] => 1141298003

[10] => 1138609592

[11] => -1

[12] => -1

[dev] => 0

[ino] => 0

[mode] => 33206

[nlink] => 1

[uid] => 0

[gid] => 0

[rdev] => 0

[size] => 92

[atime] => 1141633430

[mtime] => 1141298003

[ctime] => 1138609592

[blksize] => -1

[blocks] => -1

)

# symlink() 函数

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

## 定义和用法

symlink() 函数创建符号连接。

### 语法

link(target,link)

|  |  |
| --- | --- |
| **参数** | **描述** |
| target | 必需。 |
| link | 必需。 |

### 说明

symlink() 对于已有的 *target* 建立一个名为 *link* 的符号连接。

若成功则返回 true，失败则返回 false。

## 提示和注释

注释：本函数未在 Windows 平台下实现。

# tempnam() 函数

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

## 定义和用法

tempnam() 函数创建一个具有唯一文件名的临时文件。

若成功，则该函数返回新的临时文件名。若失败，则返回 false。

### 语法

tempnam(dir,prefix)

|  |  |
| --- | --- |
| **参数** | **描述** |
| dir | 必需。规定创建临时文件的目录。 |
| prefix | 必需。规定文件名的开头。 |

### 说明

在指定目录中建立一个具有唯一文件名的文件。如果该目录不存在，tempnam() 会在系统临时目录中生成一个文件，并返回其文件名。

在 PHP 4.0.6 之前，tempnam() 函数的行为取决于系统。在 Windows 下 TMP 环境变量会越过 dir 参数，在 Linux 下 TMPDIR 环境变量优先，而在 SVR4 下总是使用 dir 参数，如果其指向的目录存在的话。

## 提示和注释

注释：如果 PHP 不能在指定的 *dir* 参数中创建文件，则退回到系统默认值。

注释：本函数的行为在 4.0.3 版中改变了。也会建立一个临时文件以避免竞争情形，即有可能会在产生出作为文件名的字符串与脚本真正建立该文件之间会在文件系统中存在同名文件。注意，如果不再需要该文件则要删除此文件，不会自动删除的。

提示：参见 [tmpfile()](http://www.w3school.com.cn/php/func_filesystem_tmpfile.asp)

## 例子

<?php

echo tempnam("C:\inetpub\testweb","TMP0");

?>

输出：

C:\inetpub\testweb\TMP1.tmp

# tmpfile() 函数

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

## 定义和用法

tmpfile() 函数以读写（w+）模式建立一个具有唯一文件名的临时文件。

文件会在关闭后（用 fclose()）自动被删除，或当脚本结束后。

### 语法

tmpfile()

## 提示和注释

提示：参见 [tempnam()](http://www.w3school.com.cn/php/func_filesystem_tempnam.asp)。

## 例子

<?php

$temp = tmpfile();

fwrite($temp, "Testing, testing.");

//倒回文件的开头

rewind($temp);

//从文件中读取 1k

echo fread($temp,1024);

//删除文件

fclose($temp);

?>

输出：

Testing, testing.

# touch() 函数

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

## 定义和用法

touch() 函数设置指定文件的访问和修改时间。

### 语法

touch(filename,time,atime)

|  |  |
| --- | --- |
| **参数** | **描述** |
| filename | 必需。规定要接触的文件。 |
| time | 可选。设置时间。默认是当前系统时间。 |
| atime | 可选。设置访问时间。默认是当前系统时间。 |

### 说明

尝试将由 *filename* 给出的文件的访问和修改时间设定为指定的时间。如果没有设置可选参数 *time*，则使用当前系统时间。如果给出了第三个参数 *atime*，则指定文件的访问时间会被设为 atime 。

如果成功则返回 true，失败则返回 false。

## 提示和注释

注释：如果文件不存在，则会被创建。

## 例子

<?php

touch("test.txt");

?>

# umask() 函数

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

## 定义和用法

umask() 函数改变当前的 umask。

umask() 将 PHP 的 umask 设定为 mask & 0777 并返回原来的 umask。当 PHP 被作为服务器模块使用时，在每个请求结束后 umask 会被恢复。

无参数调用 umask() 会返回当前的 umask。

### 语法

umask(mask)

|  |  |
| --- | --- |
| **参数** | **描述** |
| mask | 必需。规定新的权限。默认是 0777。 |

## 提示和注释

注释：在多线程的服务器上尽量避免使用这个函数。创建文件后要改变其权限最好还是使用 [chmod()](http://www.w3school.com.cn/php/func_filesystem_chmod.asp)。使用 umask() 会导致并发程序和服务器发生不可预知的情况，因为它们使用相同的 umask。

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

# unlink() 函数

[PHP Filesystem 函数](http://www.w3school.com.cn/php/php_ref_filesystem.asp)

## 定义和用法

unlink() 函数删除文件。

若成功，则返回 true，失败则返回 false。

### 语法

unlink(filename,context)

|  |  |
| --- | --- |
| **参数** | **描述** |
| filename | 必需。规定要删除的文件。 |
| context | 可选。规定文件句柄的环境。Context 是可修改流的行为的一套选项。 |

## 提示和注释

注释：对 *context* 的支持是 PHP 5.0.0 添加的。

## 例子

<?php

$file = "test.txt";

if (!unlink($file))

{

echo ("Error deleting $file");

}

else

{

echo ("Deleted $file");

}

?>