Understanding and Working with files and directories in php programming.

In PHP, you can use various functions to work with files and directories on your server. Here are some of the most commonly used functions:

1. Opening and Closing Files: You can use the fopen() function to open a file in PHP. It takes two arguments: the file name (including the path) and the mode in which you want to open the file. The modes include read, write, and append, among others. After you've finished working with a file, you should close it using the fclose() function.

// open file in read mode

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

// close file

fclose($file);

1. Reading from Files: To read from a file in PHP, you can use the, which reads a line from fgets() function the file and returns it as a string. Alternatively, you can use the fread() function to read a specified number of bytes from the file. You can also use the file() function to read the entire file into an array.

// read a line from the file

$line = fgets($file);

// read 10 bytes from the file

$data = fread($file, 10);

// read the entire file into an array

$lines = file("file.txt");

1. Writing to Files: To write to a file in PHP, you can use the fwrite() function, which writes a string to the file. You can also use the file\_put\_contents() function to write an entire string to a file.

// write a string to the file

fwrite($file, "Hello, world!");

// write an entire string to a file

file\_put\_contents("file.txt", "Hello, world!");

1. Creating and Deleting Files: To create a new file in PHP, you can use the touch() function. To delete a file, you can use the unlink() function.

// create a new file

touch("new\_file.txt");

// delete a file

unlink("file.txt");

1. Working with Directories: To create a new directory in PHP, you can use the mkdir() function. To remove a directory, you can use the rmdir() function. You can also use the opendir() function to open a directory, and then loop through its contents using the readdir() function.

// create a new directory

mkdir("new\_directory");

// remove a directory

rmdir("old\_directory");

// open a directory

$dir = opendir("directory");

// loop through directory contents

while (($file = readdir($dir)) !== false) {

echo $file;

}

// close directory

closedir($dir);

These are just a few examples of the functions available in PHP for working with files and directories. By using these functions, you can create, read, write, and delete files, as well as create and remove directories on your server.

PHP File Handling

PHP File System allows us to create file, read file line by line, read file character by character, write file, append file, delete file and close file.

PHP Open File - fopen()

The PHP fopen() function is used to open a file.

**Syntax**

1. resource fopen ( string $filename , string $mode [, bool $use\_include\_path = false [, resource $context ]]
2. <?php
3. $handle = fopen("c:\\folder\\file.txt", "r");
4. ?>

PHP Close File - fclose()

The PHP fclose() function is used to close an open file pointer.

**Syntax**

fclose ( resource $handle )

fclose(filepointer)

**Example**

1. <?php
2. fclose($handle);
3. ?>

PHP Read File - fread()

The PHP fread() function is used to read the content of the file. It accepts two arguments: resource and file size.

**Syntax**

1. string fread ( resource $handle , int $length )

**Example**

1. <?php
2. $filename = "c:\\myfile.txt";
3. $handle = fopen($filename, "r");//open file in read mode
5. $contents = fread($handle, filesize($filename));//read file
7. echo $contents;//printing data of file
8. fclose($handle);//close file
9. ?>

PHP Read File

PHP provides various functions to read data from file. There are different functions that allow you to read all file data, read data line by line and read data character by character.

The available PHP file read functions are given below.

* fread()
* fgets()

fgetc()

PHP Read File - fread()

The PHP fread() function is used to read data of the file. It requires two arguments: file resource and file size.

Syntax

1. string fread (resource $handle , int $length )

**$handle** represents file pointer that is created by fopen() function.

PlayNext

Unmute

Current TimeÂ 0:00

/

DurationÂ 18:10

Loaded: 0.37%

Â

Fullscreen

Backward Skip 10sPlay VideoForward Skip 10s

**$length** represents length of byte to be read.

Example

<?php

1. $filename = "c:\\file1.txt";
2. $fp = fopen($filename, "r");//open file in read mode
4. $contents = fread($fp, filesize($filename));//read file
6. echo "<pre>$contents</pre>";//printing data of file
7. fclose($fp);//close file
8. ?>
9. this is first line
10. this is another line
11. this is third line
12. Output
13. this is first line
14. this is another line
15. this is third line

PHP Read File - fgets()

The PHP fgets() function is used to read single line from the file.

Syntax

1. string fgets ( resource $handle [, int $length ] )

Example

1. <?php
2. $fp = fopen("c:\\file1.txt", "r");//open file in read mode
3. echo fgets($fp);
4. fclose($fp);
5. ?>

Output

this is first line

PHP Read File - fgetc()

The PHP fgetc() function is used to read single character from the file. To get all data using fgetc() function, use !feof() function inside the while loop.

Syntax

1. string fgetc ( resource $handle )

Example

1. <?php
2. $fp = fopen("c:\\file1.txt", "r");//open file in read mode
3. **while**(!feof($fp)) {
4. echo fgetc($fp);
5. }
6. fclose($fp);
7. ?>

Output

this is first line this is another line this is third line

PHP Write File

PHP fwrite() and fputs() functions are used to write data into file. To write data into file, you need to use w, r+, w+, x, x+, c or c+ mode.

PHP Write File - fwrite()

The PHP fwrite() function is used to write content of the string into file.

**Syntaxgghjjkkllll**

1. int fwrite ( resource $handle , string $string [, int $length ] )
2. ?php
3. $fp = fopen('data.txt', 'w');//opens file in write-only mode
4. fwrite($fp, 'welcome ');
5. fwrite($fp, 'to php file write');
6. fclose($fp);
8. echo "File written successfully";
9. ?>

Output: data.txt

welcome to php file write

PHP Overwriting File

If you run the above code again, it will erase the previous data of the file and writes the new data. Let's see the code that writes only new data into data.txt file.

1. <?php
2. $fp = fopen('data.txt', 'w');//opens file in write-only mode
3. fwrite($fp, 'hello');
4. fclose($fp);
6. echo "File written successfully";
7. ?>

Output: data.txt

hello

### How does Append File work in PHP?

Appending File in PHP is a very important concept of PHP Programming in order to handle file data based on our/user requirement. In order to append file text/file to the file, a mode or a+ mode is needed to be mentioned in the fopen() function to handle the file data. The “read()” function is used to read the data which is present in the text file. The “fwrite()” function of the PHP Programming Language is used to write the data to the text file/other. Check out the examples below so that you can understand how the append file is working

PHP Append to File

If you use **a** mode, it will not erase the data of the file. It will write the data at the end of the file.

You can append data into file by using a or a+ mode in fopen() function. Let's see a simple example that appends data into data.txt file.

Let's see the data of file first.

data.txt

welcome to php file write

## PHP Append to File - fwrite()

The PHP fwrite() function is used to write and append data into file

**Example**

1. <?php
2. $fp = fopen('data.txt', 'a+');//opens file in append mode
3. fwrite($fp, ' this is additional text ');
4. fwrite($fp, 'appending data');
5. fclose($fp);
7. echo "File appended successfully";
8. ?>

Output: data.txt

welcome to php file write this is additional text appending data

# PHP Delete File

In PHP, we can delete any file using unlink() function. The unlink() function accepts one argument only: file name. It is similar to UNIX C unlink() function.

PHP unlink() generates E\_WARNING level error if file is not deleted. It returns TRUE if file is deleted successfully otherwise FALSE.

**Syntax**

1. bool unlink ( string $filename [, resource $context ] )

**$filename** represents the name of the file to be deleted.

## PHP Delete File Example

1. <?php
2. $status=unlink('data.txt');
3. **if**($status){
4. echo "File deleted successfully";
5. }**else**{
6. echo "Sorry!";
7. }
8. ?>

Output

File deleted successfully

**What is the rename() function in PHP?**

The **rename()** function in PHP renames an existing file or directory.

## Syntax

rename(string ＄oldname, string ＄newname, resource ＄context = ?): bool

## Parameters

oldname: This specifies the old name of the file or directory we want to rename.

newname: This specifies the new name of a file or directory.

context: This is an optional parameter that is a set of options able to alter the behavior of a stream.

## Return value

rename() returns true if the name is successfully changed. Otherwise, in case of any error, it returns false.

Note:

* If a file is being renamed and the newname specified already exists, the file is overwritten.
* If a directory is being renamed and the newname specified already exists as a directory, a warning is produced.

## Code

The following code shows how to use the rename() function.

**<?php**

echo "contents of task1.txt:\n";

readfile("task.txt");

task

echo "\ncontents of task2.txt: \n";

readfile("newtask.txt");

rename("task1.txt","task2.txt");

echo "\nAfter rename: ";

readfile("task2.txt");

**?>**

<?php

// Old Name Of The file

$old\_name = "gfg.txt" ;

// New Name For The File

$new\_name = "newgfg.txt" ;

// using rename() function to rename the file

rename( $old\_name, $new\_name) ;

?>

**Splitting a File Path Name into Parts in PHP**

**Q**

How To Break a File Path Name into Parts in PHP?

**A**



If you have a file name, and want to get different parts of the file name, you can use the pathinfo() function. It breaks the file name into 3 parts: directory name, file base name and file extension; and returns them in an array. Here is a PHP script example on how to use pathinfo():

<?php

$pathName = "/temp/download/todo.txt";

$parts = pathinfo($pathName);

print\_r($parts);

print("\n");

?>

This script will print:

Array

(

[dirname] => /temp/download

[basename] => todo.txt

[extension] => txt

)

**⇒** [Processing Web Forms in PHP](http://www.dev.fyicenter.com/1000152_Processing_Web_Forms_in_PHP.html)

**⇐** [Retrieving Directory Name from File Path Name in PHP](http://www.dev.fyicenter.com/1000150_Retrieving_Directory_Name_from_File_Path_Name_in_PHP.html)

**⇑** [Working with Directories and Files in PHP](http://www.dev.fyicenter.com/1000143_Working_with_Directories_and_Files_in_PHP.html)

**Reading and writing characters in file:-**

### **Example**[**Get your own PHP Server**](https://www.w3schools.com/spaces/)

Read one character from the open file:

<?php  
$file = fopen("test.txt","r");  
echo fgetc($file);  
fclose($file);  
?>

## Definition and Usage

The fgetc() function returns a single character from an open file.

**Note:** This function is slow and should not be used on large files. If you need to read one character at a time from a large file, use [fgets()](https://www.w3schools.com/php/func_filesystem_fgets.asp) to read data one line at a time and then process the line one single character at a time with fgetc().

## Syntax

fgetc(file)

## Parameter Values

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| File | Required. Specifies the open file to return a single character from |

## More Examples

### **Example**

Read open file, character by character:

<?php  
$file = fopen("test.txt","r");  
while (! feof($file)) {  
  echo fgetc($file);  
  }  
fclose($file);  
?>

# PHP fgetcsv() Function

### **Example**[**Get your own PHP Server**](https://www.w3schools.com/spaces/)

Read and output one line from the open CSV file:

<?php  
$file = fopen("contacts.csv","r");  
print\_r(fgetcsv($file));  
fclose($file);  
?>

## Definition and Usage

The fgetcsv() function parses a line from an open file, checking for CSV fields.

**Tip:** Also see the [fputcsv()](https://www.w3schools.com/php/func_filesystem_fputcsv.asp) function.

## Syntax

fgetcsv(file, length, separator, enclosure)

## Parameter Values

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| File | Required. Specifies the open file to return and parse a line from |
| length | Optional. Specifies the maximum length of a line. Must be greater than the longest line (in characters) in the CSV file. Omitting this parameter (or setting it to 0) the line length is not limited, which is slightly slower. **Note:** This parameter is required in versions prior to PHP 5 |
| separator | Optional. Specifies the field separator. Default is comma ( , ) |
| enclosure | Optional. Specifies the field enclosure character. Default is " |
| escape | Optional. Specifies the escape character. Default is "\\" |

# PHP fgets() Function

### **Example**[**Get your own PHP Server**](https://www.w3schools.com/spaces/)

Read one line from the open file:

<?php  
$file = fopen("test.txt","r");  
echo fgets($file);  
fclose($file);  
?>

[Run Example »](https://tryphp.w3schools.com/showphp.php?filename=demo_func_filesystem_fgets)

## Definition and Usage

The fgets() function returns a line from an open file.

## Syntax

fgets(file, length)

## Parameter Values

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| File | Required. Specifies the open file to return a line from |
| length | Optional. Specifies the number of bytes to read. Reading stops when length-1 bytes have been reached, or when a new line occurs, or on EOF. If no length is specified, it reads until end of the line |

## More Examples

### **Example**

Read open file, line by line:

<?php  
$file = fopen("test.txt","r");  
  
while(! feof($file))  
  {  
  echo fgets($file). "<br />";  
  }  
  
fclose($file);  
?>

File permissions

you will learn how to deal with PHP file permissions**,** including checking and changing file permissions.

File permissions specify what a user can do with a file, e.g., reading, writing, or executing it. Notice that PHP automatically grants appropriate permissions behind the scenes.

For example, if you [create a new file](https://www.phptutorial.net/php-tutorial/php-create-file/) for writing, PHP automatically grants the read and write permissions.

PHP provides some useful functions for checking and changing the file permissions.

## Checking file permissions

PHP has three handy functions that check file permissions:

* is\_readable() function returns true if the file exists and is readable; otherwise, it returns false.
* is\_writable() function returns true if the file exists and is writable; otherwise, it returns false.
* is\_executable() function returns true if the file exists and executable; otherwise, it returns false.

Let’s take a look at the following example:

<?php

$filename = 'readme.txt';

$functions = [

'is\_readable',

'is\_writable',

'is\_executable'

];

foreach ($functions as $f) {

echo $f($filename) ? 'The file ' . $filename . $f : '';

}

Besides those functions, PHP also provides the fileperms() function that returns an integer, which represents the permissions set on a particular file. For example:

<?php

$permissions = fileperms('readme.txt');

echo substr(sprintf('%o', $permissions), -4); //0666

## Changing file permissions

To change the file permission or mode, you use the chmod() function:

chmod ( string $filename , int $permissions ) : boolCode language: PHP (php)

The chmod() function has two parameters:

* $filename is the file that you want to change the permissions.
* $permissions parameter consists of three octal number components that specify access restrictions for the owner, the user group in which the owner is in, and everyone else in this sequence.

The chmod() function returns true on success or false on failure.

The permissions argument is represented by an octal number that contains three digits:

* The first digit specifies what the owner of the file can read, write, or execute the file.
* The second digit specifies what the user group in which the owner is in can read, write, or execute the file.
* The third digit specifies what everyone else can read, write, or execute the file.

The following table illustrates the value of each digit that represents the access permission for particular users ( owner, user group, or everyone else) :

| **Value** | **Permission** |
| --- | --- |
| 0 | cannot read, write or execute |
| 1 | can only execute |
| 2 | can only write |
| 3 | can write and execute |
| 4 | can only read |
| 5 | can read and execute |
| 6 | can read and write |
| 7 | can read, write and execute |

The following example sets permission that the only owner can read and write a file, everyone else only can read the file:

<?php

$filename = './readme.txt';

chmod($filename, 0644);

Notice that we put 0 before 644 to instruct PHP to treat it as an octal number.

## Summary

* Use the is\_readable(), is\_writable(), is\_executable() to check if a file exists and readable, writable, and executable.
* Use the chmod() function to set permissions for a file.

Which functions are reading and writing characters in files in PHP?

The **fread()** function can read a string of characters from a file. It takes two arguments: a file handle and the number of characters to read. The function reads the specified number of characters or less if the end of the file is reached and returns them as a string.

## Introduction

The fread() function can read a string of characters from a file.

It takes two arguments: a file handle and the number of characters to read.

The function reads the specified number of characters or less if the end of the file is reached and returns them as a string.

For example:

$handle = fopen(**"data.txt"** ,**"r"** );

$data = fread($handle, 10);

This code reads the first ten characters from data.txt and assigns them to $data as a string.

When working with binary files a character is always one byte long, so ten characters equals ten bytes.

If there are less than ten characters left to read in the file, fread() simply reads and returns as many as there are.

By the way, if you want to read only one character at a time you can use the fgetc() function.

fgetc() takes a single argument - a file handle - and returns just one character from the file it points to; it returns false when it reaches the end of the file:

$one\_char = fgetc($handle);

You can use the fwrite() function to write data to a file.

It requires two arguments: a file handle and a string to write to the file.

The function writes the contents of the string to the file, returning the number of characters written or false if there's an error.

For example:

$handle = fopen(**"data.txt"** ,**" w"** );

fwrite($handle,**"ABCxyz"** );

The first line opens the file data.txt for writing, which erases any existing data in the file.

The second line writes the character string "ABCxyz" to the beginning of the file.

You can limit the number of characters written by specifying an integer as a third argument.

The function stops writing after that many characters.

For example, the following code writes the first four characters of "abcd" to the file:

fwrite($handle,**"abcdefghij"** , 4);