

# PHP File Handling

## Overview of PHP File Handling

All modern software requires to interact with files. They either require to accept inputs in the form of files or either generate output and add it to the file. In either scenario, the capability of integrating with files has become an integral feature of almost all software that is used to run businesses. For any application, the handling of files is necessary. The file must be processed for some tasks to be performed. File handling in PHP is similar to file handling by any language such as C. PHP has a lot of normal file functions to work with.

## Use Case for PHP File Handling Capacity

As an example, banks would require software that would help them generate reports like bank account statements for 3 months or 6 months period, e-commerce companies would require to print reports related to stock inventory and sales and last but not the least applications related to stock market trade would require daily stock prices to be provided in the form of readable file. I am sure with this example, You would agree that any software that supports a business function would require you to read or write data to a file.

Since [file handling capacity](#) is almost a necessity in the modern-day application, all prominent programming languages like Python, Java, C#, and PHP provide inbuilt file handling functions that are utilized by developers to develop interactive applications.

## File Handling Capability in PHP

PHP supports the following file formats for reading and write operations.

- **Text Files:** Files with extension .txt
- **Log Files:** Files with extension .log
- **Custom Extensions:** Files with a custom extension like .abc
- **CSV Files:** Files with extension .csv
- **Image Files:** Files with extension .jpg/png/gif
- **File with Initialization Setting:** Files with extension .ini

## File Handling Functions in PHP

PHP provides a wide range of in-built functions to perform various file operations. These file functions work well with all OS systems like Linus, Unix, MAC, and Windows. However, file names in MAC OS and Windows are not case sensitive while those in Unix and Linux are case sensitive. Hence to avoid any confusion or errors, it is considered a best practice to names all files in lower cases as it ensures complete platform compatibility.

Now that we have a high-level understanding of how php file handling functions work, let's understand these functions one by one.

Popular Course in this category

[PHP Training \(5 Courses, 3 Project\)5 Online Courses | 3 Hands-on Project | 28+ Hours | Verifiable Certificate of Completion | Lifetime Access](#)  
[4.5 \(6,613 ratings\)](#)

[Course Price](#)

[\\$29 \\$399](#)

[View Course](#)

---

Related Courses

[Java Servlet Training \(6 Courses, 12 Projects\)](#)[All in One Software Development Bundle \(600+ Courses, 50+ projects\)](#)

### 1. file\_exists() Function

The function is used to verify the existence of the filename supplied to it as its parameter. It is used to avoid errors that could be caused by attempting to read or write a non-existent file.

#### Syntax:

```
<?php
file_exists($file_name) //where file_name would be a file with
one of the supported extensions
?>
```

The file\_exists() would return a True value if the file exists else would return false if the file does not exist.

Now let's use this function in a code spec to check the existence of a file. Let's place a file by the name "mysettings.ini" in the root folder and try to access it with the following code.

#### Code:

```
<?php
if (file_exists('mysettings.ini'))
{
echo 'yay! file found!';
}
else
{
echo 'Sorry! mysettings.ini does not exist';
}
?>
```

#### Output:

Now, if we delete the file from that location and run the above code, we would see the following output.

## 2. fopen() Function

The fopen function is [used in php](#) to open files which are to be read by the application.

### Syntax:

```
<?php
fopen($fname, $mode, $use_include_path, $context);
?>
```

In the above syntax, \$fname stands for the file name, \$mode stands for the mode in which we'd like to open the file. \$mode can be either of the following values.

- **r:** For opening the file in only read-only mode. It returns false if the file name supplied is not found on the location supplied.
- **r+:** For opening the file in both read and write mode. Similar to 'r', it also returns false if a file is not found.
- **w:** For opening the file in only write-only mode. If the file supplied doesn't exist, it attempts to create one.
- **w+:** For opening the file in both read and write mode. Similar to 'w', it also attempts to create a file if the file name supplied is not found.
- **a:** For opening the file in write-only mode and appending to the end of the file. If the file supplied doesn't exist, it attempts to create one.
- **a+:** For opening the file in both read and write mode. Similar to 'a', it also attempts to create a file if the file name supplied is not found.

## 3. fwrite() Function

As the name suggests, this function is used to write content to files.

### Syntax:

```
<?php
fwrite($handle, $data_string, $len);
?>
```

Where \$handle is the file location, \$data\_string is the text string we would like to write to the file and \$len is the optional parameter to specify the maximum length of the file.

## 4. fclose() Function

The fclose() function is used in php when the read/write operations on file are completed and we would like to close the file.

### Syntax:

```
<?php  
fclose($file_handle);  
?>
```

Where \$file\_handle stands for the file pointer.

### **5. fgets() Function**

The fgets() function is used in php to read the file line by line.

#### **Syntax:**

```
<?php  
fgets($file_handle);  
?>
```

Where \$file\_handle stands for the file pointer.

### **6. copy() Function**

The copy() function allows us to copy a file in php.

#### **Syntax:**

```
<?php  
copy($file1, $file2);  
?>
```

Where \$file1 is the original file and \$file2 is the copied file location.

### **7. unlink() Function**

The unlink() function in Php is used to delete a file.

#### **Syntax:**

```
<?php  
unlink($filename);  
?>
```

Where the \$filename is the filename to be deleted.

## **Conclusion**

With the above example, we can easily conclude that php has a wide variety of in-built functions that simplify reading and writing operations on the file. The most commonly used function include fopen() to open the file in different modes, fwrite() to write data to the file, fread() to read the file content, fclose() to close the file once the necessary operation is done, copy() to copy one file content to another and unlink to delete the unwanted files.

<https://www.educba.com/php-file-handling/>