

PHP - FILES & I/O

http://www.tutorialspoint.com/php/php_files.htm

Copyright © tutorialspoint.com

This chapter will explain following functions related to files:

- Opening a file
- Reading a file
- Writing a file
- Closing a file

Opening and Closing Files

The PHP **fopen** function is used to open a file. It requires two arguments stating first the file name and then mode in which to operate.

Files modes can be specified as one of the six options in this table.

Mode	Purpose
r	Opens the file for reading only. Places the file pointer at the beginning of the file.
r+	Opens the file for reading and writing. Places the file pointer at the beginning of the file.
w	Opens the file for writing only. Places the file pointer at the beginning of the file. and truncates the file to zero length. If files does not exist then it attempts to create a file.
w+	Opens the file for reading and writing only. Places the file pointer at the beginning of the file. and truncates the file to zero length. If files does not

	exist then it attempts to create a file.
a	<p>Opens the file for writing only.</p> <p>Places the file pointer at the end of the file.</p> <p>If files does not exist then it attempts to create a file.</p>
a+	<p>Opens the file for reading and writing only.</p> <p>Places the file pointer at the end of the file.</p> <p>If files does not exist then it attempts to create a file.</p>

If an attempt to open a file fails then **fopen** returns a value of **false** otherwise it returns a **file pointer** which is used for further reading or writing to that file.

After making a changes to the opened file it is important to close it with the **fclose** function. The **fclose** function requires a file pointer as its argument and then returns **true** when the closure succeeds or **false** if it fails.

Reading a file

Once a file is opened using **fopen** function it can be read with a function called **fread**. This function requires two arguments. These must be the file pointer and the length of the file expressed in bytes.

The files's length can be found using the **filesize** function which takes the file name as its argument and returns the size of the file expressed in bytes.

So here are the steps required to read a file with PHP.

- Open a file using **fopen** function.
- Get the file's length using **filesize** function.
- Read the file's content using **fread** function.
- Close the file with **fclose** function.

The following example assigns the content of a text file to a variable then displays those contents on the web page.

```
<html>
<head>
<title>Reading a file using PHP</title>
</head>
```

```

<body>

<?php
$filename = "/home/user/guest/tmp.txt";
$file = fopen( $filename, "r" );
if( $file == false )
{
    echo ( "Error in opening file" );
    exit();
}
$filesize = filesize( $filename );
$filetext = fread( $file, $filesize );

fclose( $file );

echo ( "File size : $filesize bytes" );
echo ( "<pre>$filetext</pre>" );
?>

</body>
</html>

```

Writing a file

A new file can be written or text can be appended to an existing file using the PHP **fwrite** function. This function requires two arguments specifying a **file pointer** and the string of data that is to be written. Optionally a third integer argument can be included to specify the length of the data to write. If the third argument is included, writing would stop after the specified length has been reached.

The following example creates a new text file then writes a short text heading inside it. After closing this file its existence is confirmed using **file_exists** function which takes file name as an argument

```

<?php
$filename = "/home/user/guest/newfile.txt";
$file = fopen( $filename, "w" );
if( $file == false )
{
    echo ( "Error in opening new file" );
    exit();
}
fwrite( $file, "This is a simple test\n" );
fclose( $file );
?>

<html>
<head>
<title>Writing a file using PHP</title>
</head>
<body>

<?php
if( file_exists( $filename ) )

```

```
{
    $filesize = filesize( $filename );
    $msg = "File  created with name $filename ";
    $msg .= "containing $filesize bytes";
    echo ($msg );
}
else
{
    echo ("File $filename does not exit" );
}
?>
</body>
</html>
```

We have covered all the function related to file input and out in [PHP File System Function](#) chapter.