8- PHP File() Handling & Functions

What is a File?

A file is simply a resource for storing information on a computer.

Files are usually used to store information such as:

- Configuration settings of a program
- Simple data such as contact names against the phone numbers.
- Images, Pictures, Photos, etc.

PHP File Formats Support

PHP file functions support a wide range of file formats that include:

- File.txt
- File.log
- File.custom_extension i.e. file.xyz
- File.csv
- File.gif, file.jpg etc
- Files provide a permanent cost effective data storage solution for simple data compared to databases that require other software and skills to manage DBMS systems.
- You want to store simple data such as server logs for later retrieval and analysis
- You want to store program settings i.e. program.ini

PHP file() Function

PHP provides a convenient way of working with files via its rich collection of built in functions.

Operating systems such as Windows and MAC OS are not case sensitive while Linux or Unix operating systems are case sensitive.

Adopting a naming conversion such as lower case letters only for file naming is a good practice that ensures maximum cross platform compatibility.

Let's now look at some of the most commonly used PHP file functions.

PHP file_exists() Function

This function is used to determine whether a file exists or not.

- It comes in handy when we want to know if a file exists or not before processing it.
- You can also use this function when creating a new file and you want to ensure that the file does not already exist on the server.

The file_exist function has the following syntax.

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

HERE,

- "file_exists()" is the PHP function that returns true if the file exists and false if it does not exist.
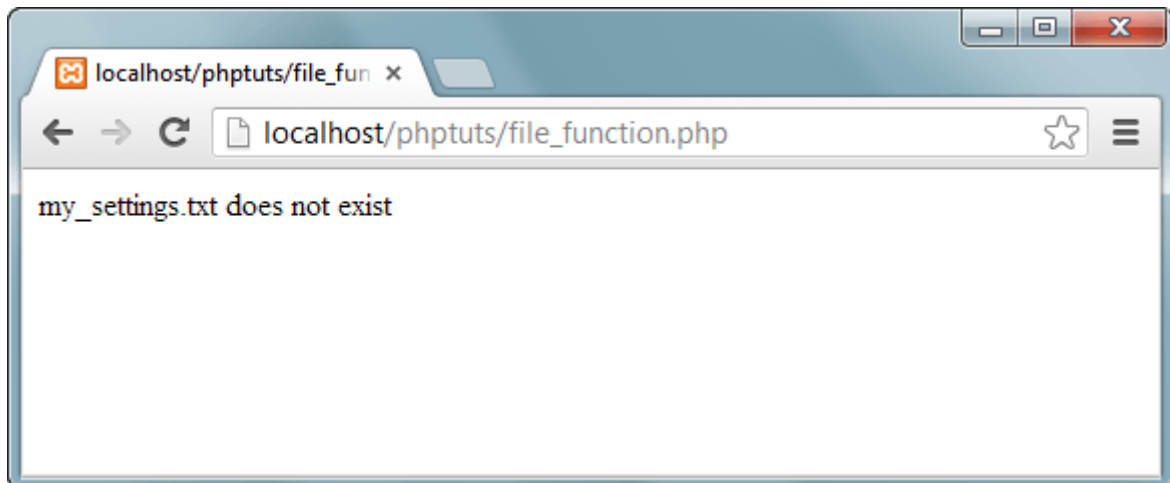- "$file_name" is the path and name of the file to be checked

The code below uses file_exists function to determine if the file my_settings.txt exists.

```php
<?php
if (file_exists('my_settings.txt'))
 {
```

```
            echo 'file found!';
        }  else
        {
            echo 'my_settings.txt does not exist';
        }
    ?>
```

Save the above code in a file named file_function.php Assuming you saved the file in phptuts folder in htdocs, open the

URL **http://localhost/phptuts/file_function.php** in your browser You will get the following results.

PHP fopen() Function

The fopen function is used to open files. It has the following syntax

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

HERE,

- "fopen" is the PHP open file function

- "$file_name" is the name of the file to be opened
- "$mode" is the mode in which the file should be opened, the table below shows the modes

Mode	Description
r	☐ Read file from beginning. ☐ Returns false if the file doesn't exist. ☐ Read only
r+	☐ Read file from beginning ☐ Returns false if the file doesn't exist. ☐ Read and write
w	☐ Write to file at beginning ☐ truncate file to zero length ☐ If the file doesn't exist attempt to create it. ☐ Write only
w+	☐ Write to file at beginning, truncate file to zero length ☐ If the file doesn't exist attempt to create it. ☐ Read and Write
a	☐ Append to file at end ☐ If the file doesn't exist attempt to create it. ☐ Write only
a+	☐ Php append to file at end ☐ If the file doesn't exist attempt to create it

	□ Read and write

- "$use_include_path" is optional, default is false, if set to true, the function searches in the include path too.
- "$context" is optional, can be used to specify the context support.

PHP fwrite() Function

The fwrite function is used to write files.

It has the following syntax

```
<?php fwrite($handle, $string,
$length);
?>
```

HERE,

- "fwrite" is the PHP function for writing to files
- "$handle" is the file pointer resource
- "$string" is the data to be written in the file.
- "$length" is optional, can be used to specify the maximum file length.

PHP fclose() Function

The fclose() function is used to close a file in php which is already open

It has the following syntax.

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

HERE,

- "fclose" is the PHP function for closing an open file □ "$handle" is the file pointer resource.

Let's now look at an example that creates my_settings.txt.

We will use the following functions.

- Fopen
- Fwrite
- fclose

The code below "create_my_settings_file.php" implements the above example.
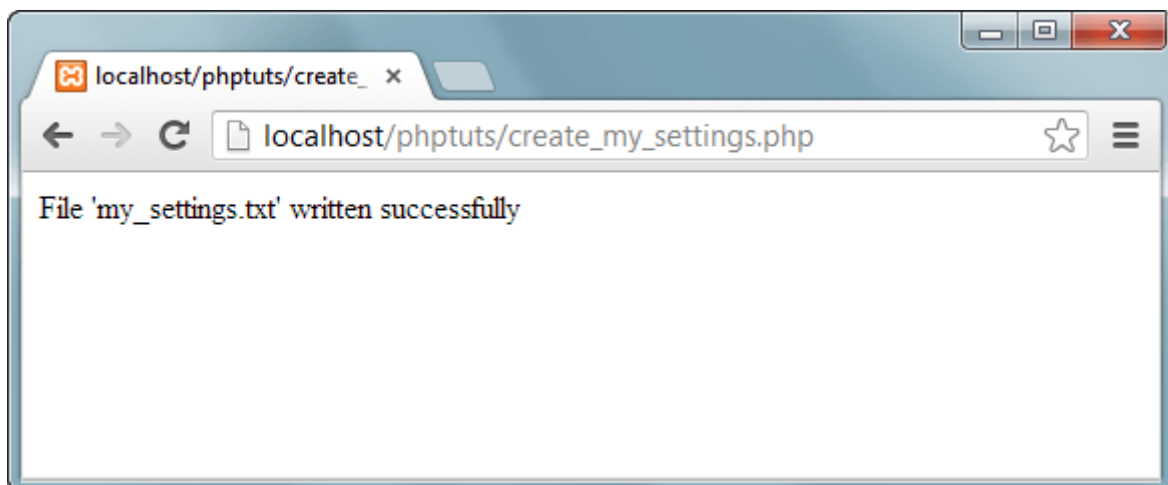
Open a file	<?php $fh = fopen("my_settings.txt", 'w') or die("Failed to create file"); ?>
Closing a file	<?php fclose($fh); ?>

Create File	```php
	<?php
	$fh = fopen("my_settings.txt", 'w') or die("Failed to create file");
	$text = <<<_END
	localhost;root;pwd1234;my_database
	_END;
	fwrite($fh, $text) or die("Could not write to file"); fclose($fh); echo "File 'my_settings.txt' written successfully";
	?>
	``

Testing the code

Open the   URL **http://localhost/phptuts/create_my_settings.php** in your browser.

You will get the following page

Note: if your disk is full or you do not have permission to write files, you will get an error message.

Switch back to the URL **http://localhost/phptuts/file_function.php** .

What results do you get?

PHP fgets() Function

The fgets function is used to read php files line by line. It has the following basic syntax. fgets($handle); HERE,

- "$fgets" is the PHP function for reading file lines ☐   "$handle" is the file pointer resource.

Let's now look at an example that reads my_settings.txt file using the fopen and fgets functions.

The code below read_my_settings.php implements the above example.

```php
<?php
$fh = fopen("my_settings.txt", 'r') or die("File does not exist or
you lack permission to open it");
$line = fgets($fh); echo
$line; fclose($fh);
 ?>
```

HERE,

- "fopen" function returns the pointer to the file specified in the file path
- "die()" function is called if an error occurs. It displays a message and exists execution of the script

PHP copy() Function

The PHP copy function is used to copy files. It has the following basic syntax. copy($file,$copied_file); HERE,

- "$file" specifies the file path and name of the file to be copied.
- "copied_file" specified the path and name of the copied file

The code below illustrates the implementation

```php
<?php copy('my_settings.txt',     'my_settings_backup.txt')
  or die("Could not copy file"); echo "File successfully
copied to 'my_settings_backup.txt'";
?>
```

Deleting a file

The unlink function is used to delete the file. The code below illustrates the implementation.

```php
<?php if
(!unlink('my_settings_backup.txt'))
{     echo "Could not delete
file";
 } else {     echo "File 'my_settings_backup.txt' successfully
deleted";
}
?>
```

PHP file_get_contents() Function

The file_get_contents function is used to read the entire file contents.

The code below illustrates the implementation.

The difference between file_get_contents and fgets is that file_get_contents returns the file data as a string while fgets reads the file line by line.

```php
<?php echo "<pre>"; // Enables display of
line feeds echo
file_get_contents("my_settings.txt"); echo
"</pre>"; // Terminates pre tag
?>
```

Summary

- A file is a resource for storing data

- PHP has a rich collection of built in functions that simplify working with files.

- Common file functions include fopen, fclose, file_get_contents □
        The table below shows a summary of the functions covered

Function	Description
File_exists	Used to determine if a file exists or not
Fopen	Used to open a file. Returns a pointer to the opened file
Fwrite	Used to write to files
Fclose	Used to open closed files
Fgets	Used to read a file line by line
Copy	Used to copy an existing file
Unlink	Used to delete an existing file

file_get_contents	Used to return the contents of a file as a string