

File Handling With PHP

Objective

In this chapter learner able to understand:

- File Handling

File Handling

We frequently use files for storing information which can be processed by our programs. In order to store information permanently and retrieve it we need to use files .

A file is collection of related data .Placed on the disk . Files are not only used for data. Our programs are also stored in files.

Why we use file Handling

- The input and out operation that we have performed so far were done through screen and keyboard only .
- after the termination of program all the entered data is lost because primary memory is volatile .
- if the data has to be used later ,then it becomes necessary to keep it in permanent storage device
- Languages are provide the concept of file handling through which data can be stored on the disk or secondary storage device. the stored data can be read whenever required.

File Handling with PHP

PHP includes a lot of built-in functions for handling files and directories .You can read, write, delete, and get lots of information on files through the use of these functions.

Open/Close a File

- A file is opened with **fopen()** as a “stream”, and PHP returns a ‘handle’ to the file that can be used to reference the open file in other functions.
- Each file is opened in a particular mode.
- A file is closed with **fclose()** or when your script ends.

File Open Modes

'r'	Open for reading only. Start at beginning of file.
'r+'	Open for reading and writing. Start at beginning of file.
'w'	Open for writing only. Remove all previous content, if file doesn't exist, create it.
'w+'	Open for writing mode. Truncates the file (effectively overwriting it). If the file doesn't exist, fopen will attempt to create the file. it deletes all information in the file when the file is opened.
'a'	Open writing, but start at END of current content.
'a+'	Open for reading and writing, start at END and create file if necessary.

File Open/Close Demo

```
<?php
    $storead = fopen('some/file.txt','r');
    $towrite = fopen('some/file.txt','w'); fclose($storead);
    fclose($towrite);
?>
```


Now what..?

- If you open a file to read, you can use more in-built PHP functions to read data..
- If you open the file to write, you can use more in-built PHP functions to write..

Reading Data

There are three main functions to read data:

- **fgetc (resource \$handle)**

The fgetc() function is used to read a single character from a file.

- **fgets(\$handle,\$bytes)**

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

- **fread(\$handle,\$bytes)**

Reads up to \$bytes of data, stops at EOF.

Note: After a call to this function the file pointer has moved to the next line.

Reading Data

- We need to be aware of the End Of File (EOF) point..
feof(\$handle)
- Whether the file has reached the EOF point. Returns true if have reached EOF.

Data Reading Example

```
<?php
    $handle = fopen('people.txt', 'r');
    while (!feof($handle))
    {
        echo fgets($handle, 1024);
        echo '<br />';
    }
    fclose($handle);
?>
```

File Open shortcuts..

- There are two 'shortcut' functions that don't require a file to be opened:
`$lines = file($filename)`
- Reads entire file into an array with each line a separate entry in the array.
`$str = file_get_contents($filename)`
- Reads entire file into a single string.

Writing Data

To write data to a file use:

int fwrite(\$handle,\$data)

Write \$data to the file.

int file_put_contents(string \$filename,\$data)

Write \$data directly into any file.

Data Writing Example

```
<?php
    $handle = fopen('people.txt', 'a');
    fwrite($handle,"hello");
    fclose($handle);
?>
```

Other File Operations

- Delete file

unlink('filename');

- Rename (file or directory)

rename('old name', 'new name');

- Copy file

copy('source', 'destination');

- File Size

int filesize (string \$filename)

Other File Operations

Rewind

bool rewind (resource \$handle)

Rewind the position of a file pointer fseek

int fseek (resource \$handle, int \$offset , int \$whence);

whence values are:

- SEEK_SET - Set position equal to offset bytes.
- SEEK_CUR - Set position to current location plus offset.
- SEEK_END - Set position to end-of-file plus offset.

If whence is not specified, it is assumed to be SEEK_SET.

Dealing With Directories

Open a directory

```
$handle = opendir('dirname');
```

- \$handle 'points' to the directory

Read contents of directory

```
readdir($handle)
```

- Returns name of next file in directory
- Files are sorted as on filesystem

Close a directory

```
closedir($handle)
```

- Close directory 'stream'

Directory Reading Example

```
<?php
    $handle = opendir('.');
    while(false !== ($file=readdir($handle)))
    {
        echo "$file <br/>";
    }
    closedir($handle);
?>
```

Other Directory Operations

Get current directory

- **getcwd()**

Change Directory

- **chdir('dirname');**

Create directory

- **mkdir('dirname');**

Delete directory (MUST be empty)

- **rmdir('dirname');**

Chapter Summary

In this chapter, you have learned:

- The `fopen()` function is used to open a file.
- The `fclose()` function is used to close a file.
- Reading and writing a file using pre-defined function of file handling of PHP.
- Managing the directory using pre-defined function of file handling of PHP.

Thank you