UNIT – 5
PHP FILE HANDLING

UNIVERSITY
OF PETROLEUM
& ENERGY STUDIES

Dr. P S V S Sridhar

Assistant Professor (SS)

Centre for Information Technology

PHP FILE HANDLING

- (1) Introduction, File Open, File Creation
- (2) Writing to files, Reading from File
- (3) Searching a record from a file, Closing a File
- (4) Using PHP With HTML Forms

INTRODUCTION

- PHP offers a large set of functions for storing the data permanently into files, and to perform manipulation like appending the data into a file, to delete the contents present in the file etc.

File Open

- Files are opened in PHP using the **fopen** function. The function takes two parameters, the first is the name of the file and the second is the mode in which it should be opened. The function returns a file pointer if successful, otherwise zero.
- Example:
- $fp=fopen("Employeedetails.txt", "r");
- Employeedetails.txt is the file which you are going to open and "r" denotes that the file has been opened in read mode. If the file exits the function will return a file pointer else it will return 0.

FILE MODES

Modes	Description
r	Read only. Starts at the beginning of the file
r+	Read/Write. Starts at the beginning of the file
w	Write only. Open and clears the contents of file; or creates a new file if it doesn't exist
w+	Read/Write. Open and clears the contents of file; or creates a new file if it doesn't exist
a	Append. Open and writes to the end of the file or creates a new file if it doesn't exist
a+	**Read/Append. Preserves file content by writing to the end of the file**
x	Write only. Creates a new file. Returns FALSE and an error if file already exists
x+	Read/Write. Creates a new file. Returns FALSE and an error if file already exists

File Creation

- In PHP the **fopen** function is used to open files. However, it can also create a file if it does not find the file specified in the function call. So if you use fopen on a file that does not exist, it will create it, given that you open the file for writing or appending. For example:
- **$fp=fopen("Employeedetails.txt", "w");**
- In the above code you have opened the file in the write mode, so if Employeedetails.txt file does not exist, it will create a file.

Opening file

The open operation fails, fopen returns FALSE

```
$handle = fopen (“file.txt”, “r”);
if ($handle)
{
        echo “File opened OK”;
}
```

Writing to the files

- The **fwrite** function is used to write a string, or part of a string to a file. The function takes three parameters, the file handle, the string to write, and the number of bytes to write. If the number of bytes is omitted, the whole string is written to the file.

fwrite(file,string,[length])

length is optional, maximum number of bytes to write

```
$fp=fopen("employeedetails.txt","w");

$name="john";
$id=123;
$address="Bangalore";
$salary=30000.56;
fwrite($fp,$name);
fwrite($fp,$id);
fwrite($fp,$address);
fwrite($fp,$salary);
```

Writing to the files

- You can open the file in the append mode, where it will open the file and moves the file pointer to the end of file.
- $fp=fopen("employeedetails.txt","a");
  $name="amit";
  $id=124;
  $address="Chennai";
  $salary=30000.56;fwrite($fp,"\r\n");
  fwrite($fp,$name);
  fwrite($fp,$id);
  fwrite($fp,$address);
  fwrite($fp,$salary);
- Note: Windows requires a carriage return character as well as a new line character, so if you're using Windows, terminate the string with \r\n.

Reading from Files

- You can read from files opened in r, r+, w+, and a+ mode. The **fread** function is used for getting data out of a file. The function requires a file pointer, which we have, and an integer to tell the function how much data, in bytes, it is supposed to read.

- **$fp=fopen("employeedetails.txt","r");**

  $content=fread($fp,7); // 7 is number of characters

  echo $content;

Reading from Files

- You want to read all the data from the file, then you need to get the size of the file. The **filesize** function is used to find the length of a file, in bytes.
- **$fp=fopen("employeedetails.txt","r");**

  $content=fread($fp,filesize("employeedetails.txt"));

  echo $content;
- you read a line of data at a time from a file with the gets function. If you had separated your data with new lines then you could read one segment of data at a time with the **fgets** function.
- **$fp=fopen("employeedetails.txt","r");**

  $content=fgets($fp); //fgets function read one line of data

  echo $content;

Reading from Files

- **You want to read all the lines from the file using fgets**

```
while(!(feof($fp))) //feof function is used to determine if the end of file is true.

{

$content=fgets($fp);

echo $content;

}
```

- **You can read a single character at a time from a file using fgetc**

- ```
  while(!(feof($fp)))
  {
  $ch=fgetc($fp);
  echo $ch;
  }
  ```
  ```

Reading from Files

fgetc returns FALSE, there are no more characters to read.

```
$handle = fopen (“file.txt”, “r”);
while ($char = fgetc($handle))       // to access all the characters from file
echo “$char”;
```

Searching a record from a file

- The **preg_match** PHP function is used to search a string, and return 1 or 0. If the search was successful 1 will be returned, and if it was not found 0 will be returned.

preg_match(search_pattern, your_string)

- ```
  $fp=fopen("employeedetails.txt","r");
  while(!(feof($fp)))
  {
  $content=fgets($fp);

  if (preg_match('/amit/',$content))

  echo $content."<br>";
  }
  ```

/amit/ is a search pattern
  ```

Common Perl-Compatible Pattern Constructs

Common Perl-Compatible Pattern Constructs

Construct	Interpretation
Simple literal character matches	If the character involved is not special, Perl will match characters in sequence. The example pattern `/abc/` matches any string that has the substring `"abc"` in it.
Character class matches: [<*list of characters*>]	Will match a single instance of any of the characters between the brackets. For example, `/[xyz]/` matches a single character, as long as that character is either `x`, `y`, or `z`. A sequence of characters (in ASCII order) is indicated by a hyphen, so that a class matching all digits is `[0-9]`.
Predefined character class abbreviations	The patterns `\d` will match a single digit (from the character class `[0-9]`), and the pattern `\s` matches any whitespace character.
Multiplier patterns	Any pattern followed by * means: "Match this pattern 0 or more times." Any pattern followed by ? means: "Match this pattern exactly once." Any pattern followed by + means: "Match this pattern 1 or more times."
Anchoring characters	The caret character ^ at the beginning of a pattern means that the pattern must start at the beginning of the string; the $ character at the end of a pattern means that the pattern must end at the end of the string. The caret character at the beginning of a character class `[^abc]` means that the set is the complement of the characters listed (that is, any character that is not in the list).
Escape character '\'	Any character that has a special meaning to regex can be treated as a simple matching character by preceding it with a backslash. The special characters that might need this treatment are: . \ + * ? [ ] ^ $ ( ) { } = ! < > | :
Parentheses	A parenthesis grouping around a portion of any pattern means: "Add the substring that matches this pattern to the list of substring matches."

Reading a whole File at Once with file_get_contents

You can read the entire contents of a file with the **file_get_contents** function

file_get_contents(filename, [use_include_path [,context [,offset [,maxlen]]]]);

Filename is name of the file name

use_include_path is set to TRUE if want to search path,

Context is a context for operation(behaviour of the stream), you can leave it as NULL

Offset is the file at which to start reading

Maxlen is the maximum length

```
<?php
$text = file_get_contents("file.txt");
echo $text;
$fixed_text = str_replace("\n","<br>", $text);
echo $fixed_text;
?>
```

Reading a File into an Array with file

- **file** function to read a file into an array at once. Each line becomes an element in the array.
- file (filename, [,use_includ_path [,context]]);

```
<?php
$data = file("file.txt");
foreach($data as $line)
{
        echo $line, "<br>";
}
?>
```

Closing a File

- The **fclose()** function is used to close an open file.

- $fp=fopen("employeedetails.txt","r");

  $content=fread($fp,filesize("employeedetails.txt"));

  echo $content;

  fclose($fp);

Checking if a file exists with file_exists

- You are working on files that doesn't exist, you w'll get error. To avoid, you can check if a file exists with the **file_exists** function
- **file_exists(filename);**
- It returns TRUE if the file exists, FALSE otherwise.

```
<?php
$filename = "does_not_exists.txt";
if (file_exists($filename))
{
        $data = file($filename);
        print_r($data);
}
else
        echo "filename does not exists";
?>
```

Getting File size with filesize

- You can get file size by **filesize** function, returned as an integer (bytes) back or FALSE if the file doesn't exists.
- filesize(filename);

filesize("file.txt");

Reading Binary Reads with fread

- To handle files in binary way
- **fread(handle, length);**
- This function reads up to **length** bytes from the file referenced by **handle.** Reading stops when length bytes have been read, of the EOF is reached.

```
$handle = fopen("file.txt", "rb");

$text = fread($handle, filesize("file.txt");

$fixed_text = str_replace("\n", "<br>", $text);

echo $fixed_text;

fclose($handle);
```

Parsing Files with fscanf

- Parse files with the fscanf
- E.g This function takes a file, the format in the file is “%s\t%s\n” (string, tab, string,newline)
- It will return to an array

```
$handle = fopen(“actors.txt”, “r”);

while ($name = fscanf($handle, ““%s\t%s\n”))      //$name is an array

{

list($firstname, $lastname) = $name;

echo $firstname, $lastname;

}

fclose($handle);
```

File manipulation functions

- **rewind()** The rewind function is used to move the file pointer to the beginning of the file.

```
<?php
$handle = fopen('output.txt', 'r+');

fwrite($handle, 'Really long sentence.');
rewind($handle);
fwrite($handle, 'Foo');
rewind($handle);

echo fread($handle, filesize('output.txt'));

fclose($handle);
?>
```

Output:

Foolly long sentence.

File manipulation functions

fputs() The fputs() function is an alias of the fwrite() function.

```
<?php
$fp=fopen("studentdetails.txt","a");
fputs($fp,$rollno);
fputs($fp," ");
fputs($fp,$name);
fputs($fp," ");
fputs($fp,$age);
fputs($fp," ");
fputs($fp,$city);
echo "details inserted";
fclose($fp);
?>
```

File manipulation functions

- **fprintf()** The fprintf() function writes a formatted string to a file.

fprintf(stream,format,arg1,arg2,arg++)

format Required. Specifies the string and how to format the variables in it.Possible format values:
- %% - Returns a percent sign
- %b - Binary number
- %c - The character according to the ASCII value
- %d - Signed decimal number
- %e - Scientific notation (e.g. 1.2e+2)
- %u - Unsigned decimal number
- %f - Floating-point number (local settings aware)
- %F - Floating-point number (not local settings aware)
- %o - Octal number
- %s - String
- %x - Hexadecimal number (lowercase letters)
- %X - Hexadecimal number (uppercase letters)

```
<?php
$str = "Hello";
$number = 123;
$file = fopen("test.txt","w");
echo fprintf($file,"%s world. Day number
%u",$str,$number);
?>
```

The following text will be written to the file "test.txt":
Hello world. Day number 123

File manipulation functions

- **fscanf()** The fscanf() function parses the input from an open file according to the specified format.
- **fscanf(file,format,mixed)**

Copying Files with copy

copy(source, destination)

This function returns TRUE if it was successful, FALSE otherwise.

```
<?php
$file = ‘file.txt’;
$copy = ‘copy.txt’;
If (copy($file, $copy))
{
        echo “copied $file”;
}
else
{       echo “could not copy $file”;
}
?>
```

Deleting Files with unlink

unlink(filename [,context]) – Delete a file

This function returns TRUE if the file was deleted, FALSE otherwise

```
if (unlink(“copy.txt”))
{           echo “Deleted the file”;              }
```

Writing to a File with fwrite

fwrite(handle, string [,length]) – To write a string to a file

This function returns the number of bytes written, or FALSE if there was an error.

```
<?

$handle = fopen(“data.txt”, “w”);

$text = “here\nis\nthe\text”;

if (fwrite ($handle, $text) = FALSE) {

echo “Can not write data.txt”;

}
```

Using PHP with HTML Forms

- When a form is submitted to a PHP script, the information from that form is automatically made available to the PHP script.
- **Customer.html**

```
<html>
<body>
<form action="customer.php“>
Name<input type="text" name="name"/><br />
Age<input type="text" name="age"/><br />
Gender:<input type="radio" name="gender" value="male"/> male
<input type="radio" name="gender" value="female"/> female <br />
Address <input type="text" name="address"/><br />
<input type="submit" value="save"/>
</form>
</body>
</html>
```

When you submit this form it will move on to the customer.php page.

Since we have not specified any method name, by default it would be GET.

- **Customer.php**
- <?php
  $name=$_GET["name"];
  $age=$_GET["age"];
  $gender=$_GET["gender"];
  $address=$_GET["address"];
  echo "Name:".$name."
";
  echo "Age:".$age.“
";
  echo "Gender:".$gender.“
";
  echo "Address:".$address;
  ?>

Thanks