Handling Files, Upload and EMAIL

PROF. NALINI N
AP(SR)
SCOPE
VIT

Overview

- When you are manipulating files you must be very careful because you can do a lot of damage if you do something wrong. Common errors include editing the wrong file, filling a hard-drive with garbage data, and accidentally deleting a file's contents.

Opening a File

- In PHP, a file is created using a command that is also used to open files.
- 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.

How to create a file ?

- fopen() binds a named resource, specified by filename, to a stream.
- Returns a file pointer resource on success, or FALSE on error.

```
<?php
        $file=fopen("welcome.txt","r");
?>
```

- Note: If the fopen() function is unable to open the specified file, it returns 0 (false).

```
<?php

$file=fopen("welcome.txt","r") or exit("Unable to open file!");

?>
```

How to create a file ?

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. Opens and clears the contents of file; or creates a new file if it doesn't exist
w+	Read/Write. Opens and clears the contents of file; or creates a new file if it doesn't exist
a	Append. Opens 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

Closing a File

bool **fclose** (resource *$handle*)

- The file pointed to by ***handle*** is closed.
- Returns **TRUE** on success or **FALSE** on failure.

```
<?php
        $file = fopen("test.txt","r");

        //some code to be executed

        fclose($file);
?>
```

PHP fread() Function

```
fread(file,length)
```

- The fread() reads from an open file.
- The function will stop at the end of the file or when it reaches the specified length, whichever comes first.
- This function returns the read string, or FALSE on failure.

Parameter	Description
file	Required. Specifies the open file to read from
length	Required. Specifies the maximum number of bytes to read

PHP fread() Function

```
<?php

$file = fopen("test.txt","r");

fread($file,"10");//fread($file,filesize("test.txt"));

fclose($file);

?>
```

Check End-of-file

- The feof() function checks if the "end-of-file" (EOF) has been reached.The feof() function is useful for looping through data of unknown length.

bool **feof** (resource $handle)

```
<?php
    $file = “name.txt”;
    if (feof($file))
    echo "End of file";
?>
```

- Note: You cannot read from files opened in w, a, and x mode!

Reading a File Line by Line

string fgets (resource $handle [, int $length])

- Gets a line from file pointer.
- Returns a string of up to ***length*** - 1 bytes read from the file pointed to by ***handle***. If there is no more data to read in the file pointer, then **FALSE** is returned.
- If an error occurs, **FALSE** is returned.

Reading a File Line by Line

```
<?php
    $file = fopen("welcome.txt", "r") or exit("Unable to open file!");
    //Output a line of the file until the end is reached
    while(!feof($file))
      {
      echo fgets($file). "<br>";
      }
    fclose($file);
    ?>
```

Reading a File Character by Character

string **fgetc** (resource $handle)

- Gets a character from the given file pointer.
- Returns a string containing a single character read from the file pointed to by ***handle***. Returns **FALSE** on EOF.

Reading a File Character by Character

```
<?php
   $file=fopen("welcome.txt","r") or exit("Unable to open file!");
   while (!feof($file))
     {
     echo fgetc($file);
     }
   fclose($file);
?>
```

Reading/writing an entire file


```
# reverse a file
$text = file_get_contents("poem.txt");
$text = strrev($text);
file_put_contents("poem.txt", $text);
```

- `file_get_contents` returns entire contents of a file as a string
- `file_put_contents` writes a string into a file, replacing any prior contents

Example explode


```
Harry Potter, J.K. Rowling
The Lord of the Rings, J.R.R. Tolkien
Dune, Frank Herbert
```

contents of input file books.txt

```
<?php foreach (file("books.txt") as $book) {
      list($title, $author) = explode(",", $book);
      ?>
      <p> Book title: <?= $title ?>, Author: <?=
$author ?> </p>
<?php
}
?>
```

Fixed-length files, `file` and `list`


```
XXX
(919)685-2181
570-86-7326
```

contents of file personal.txt

```
list($name, $phone, $ssn) = file("personal.txt");
```

- reads the file into an array of lines and unpacks the lines into variables
- Need to know a file's exact length/format

Write

```
<?php
$myfile = fopen("newfile.txt", "w") or die("Unable
to open file!");
$txt = “xxx\n";
fwrite($myfile, $txt);
$txt = “yyy\n";
fwrite($myfile, $txt);
fclose($myfile);
?>
```

Delete File - unlink()

- The PHP unlink() function is used to delete file.

Syntax :

- bool unlink (string $filename [, resource $context])

Example

```
<?php
unlink('data.txt');

echo "File deleted successfully";
?>
```

Some File Functions

1) **fopen(filename,mode)** - used for opening a file with specific mode.

2) **fclose(fp)** - used to close the file. (fp -> filepointer)

3) **feof(fp)** - used to find the End of File

4) **fgetc(fp)** - Gets character from file pointer

5) **fgets(fp)** - Gets a line from file pointer

6) **fread(fp,size)** - Binary-safe file read

7) **fscanf(fp,format)** - Parses input from a file according to a format

8) **fwrite(fp,string)** - Binary-safe file write

9) **file_put_contents(fp,string)** - Write a string to a file

10) **file_get_contents(fp)** - Reads entire file into a string

11) **file(fp)** - Reads entire file into an array

12) **file_exists(fp)** - Checks whether a file or directory exists

Some File Functions

13) **is_readable(fp)** - Tells whether the filename is readable

14) **is_writable(fp)** - Tells whether the filename is writable

15) **is_file(path)** - Tells whether the filename is a regular file

16) **is_dir(path)** - Tells whether the filename is a directory

17) **is_link(path)** - Tells whether the filename is a symbolic link

18) **readlink(path)** - Returns the target of a symbolic link

19) **readdir** - Read entry from directory handle

20) **glob** - Find pathnames matching a pattern

21) **filesize(fp)** - Gets file size

22) **filetype(fp)** - Gets file type

23) **fprintf(fp,format)** - Write a formatted string to a stream

24) **fstat(fp)** - Gets information about a file using an open file pointer

File Upload

❖ With PHP, it is possible to upload files to the server. HTML SCRIPT to allow users to upload files from a form.

❖ By using the global PHP $_FILES array you can upload files from a client computer to the remote server.

❖ The first parameter is the form's input name and the second index can be either "name", "type", "size", "tmp_name" or "error". Like this:

$_FILES["file"]["name"] - the name of the uploaded file
$_FILES["file"]["type"] - the type of the uploaded file
$_FILES["file"]["size"] - the size in bytes of the uploaded file
$_FILES["file"]["tmp_name"] - the name of the temporary copy of the file stored on the server
$_FILES["file"]["error"] - the error code resulting from the file upload

Configure The "php.ini" File->file_uploads = On

File Upload

E.g.

```
<html>
<body>
<form action="upload_file.php"
method="post"
enctype="multipart/form-data">
<label for="file">Filename:</label>
<input type="file" name="myfile"
id="file" />
<br />
<input type="submit"
name="submit" value="Submit" />
</form>
</body>
</html>
```

The **enctype** attribute of the <form> tag specifies which content-type to use when submitting the form. "**multipart/form-data**" is used when a form requires binary data, like the contents of a file, to be uploaded.

The **type="file"** attribute of the <input> tag specifies that the input should be processed as a file.

```
<html> <form method=post action="upload.php" enctype=multipart/form-data>
select a file<input type=file name=userfile >
 <input type=submit value=submit>
</form> </html>
```

```
<?php
if($_FILES['userfile']['size']<1000000) AND  $_FILES['userfile']['type']=="text/plain")
{
//if(@copy($_FILES['userfile']['tmp_name'],"./upload/".$_FILES['userfile']['name']))
if(move_uploaded_file($_FILES['userfile']['tmp_name'],"./upload/".$_FILES['userfile']['n
ame']))
echo "file uploaded";
else
echo "fail1";
}
else
echo "fail2";
?>
```

Adding Restrictions

```
<?php                                                    //upload_file.php
          if ((($_FILES["myfile"]["type"] == "image/gif")
|| ($_FILES["myfile"]["type"] == "image/jpeg")
|| ($_FILES["myfile"]["type"] == "image/pjpeg"))
&& ($_FILES["myfile"]["size"]/1024 < 20))
 {
 if ($_FILES["myfile"]["error"] > 0)
  {
  echo "Error: " . $_FILES["file"]["error"] . "<br />";
  }
 else
  {
  move_uploaded_file($_FILES['userfile']['tmp_name'],"./upload/".$_FILES['user
file']['name'])

      echo "Upload: " . $_FILES["myfile"]["name"] . "<br />";
       echo "Type: " . $_FILES["myfile"]["type"] . "<br />";
       echo "Size: " . ($_FILES["myfile"]["size"] / 1024) . " Kb<br />";
       echo "Stored in: " . $_FILES["myfile"]["tmp_name"];
      }
    }
   else
    {
    echo "Invalid file";
    }
   ?>
```

Emailing with PHP

If you are in a *NIX environment, you will most likely have sendmail installed on your server. If you are using a hosting service, check with your service provider to make sure sendmail or some equivalent is being used. Once you have your mail server squared away, you'll need to modify your php.ini

1. **SMTP** - Set this to the ip address or DNS name of your SMTP server. default value : localhost
2. **sendmail_from** -The From address used by default by the PHP mail() command
3. **smtp_port** - Set this to the port PHP uses to connect to the SMTP server. default value 25

Email – mail()

It is a function in PHP used to send an email. It returns **true** on success of sending an email, otherwise **false.**

mail(to,sub,message, headers, other_parameters)

(In this, the first three parameters are must.)

Where to- the receiver(s) email address(es). "**,**" used as delimiter to separate the addresses.

Sub – subject of the mail

Message – text which has to delivered to the receiver(s).

Headers – additional parameter used

1. to include BCC & CC email address(es),
2. send the mail as multipart, and
3. contains the attachment file

Other_parameters – used to display some additional information to the receiver(s).

Simple Mail program

```
<? php
$to=“user123@example.com”;
$sub=“Test mail”;
$message1 = “This is a sample test mail";
$mailsent = mail($to,$sub,$message1);
If ($mailsent)
   echo “the mail was sent to $to successfully”;
Else
   echo “the mail couldn’t send. Pl verify your mail settings / connection”;
?>
```

Output:

The mail was sent to user123@example.com successfully.

Email – with cc & Bcc

```
<? php
$to="user123@example.com";
$sub="Test mail";
$message1 = "This is a sample test mail";
$header = "cc:aa@aa.com, ab@aa.com\r\n";
$header .="bcc: ac@aa.com";
$mailsent = mail($to,$sub,$message1,$header);
If ($mailsent)
   echo "the mail was sent to $to and $headersuccessfully";
Else
   echo "the mail couldn't send. Pl verify your mail settings /
connection";
?>
```

Output:

The mail was sent to user123@example.com cc:aa@aa.com, ab@aa.com bcc: ac@aa.com successfully.

Email with an attachment

1. Collect the message and store it in a variable.
2. Add the boundary at the end of that message.
3. Enclose the content type of that message and its transfer encoding method.
4. Now add boundary to mark it as end of message.
5. Open the file which has to be attached with the mail in rb mode and transfer the content into a php variable.
6. Now add the content-type to the message as the type of attached file.

Email with an attachment

7. Then set the name as the file name which has to be sent an attachment.
8. Add "content-disposition: attachment" - which indicates the mail is coming with an attachment.
9. Can add the rename option of the attached file.
10. Add the transfer encoding type.
11. Now it is a time to add the content of the attachment file.
12. Add the boundary with the message as a ended one.

Sample Program


```
<?php
$to="nalini@vit.ac.in";
$sub="fileattachmaent";
$msg="pdf";
$fn="ss.pdf";
$fo=fopen($fn,"rb");
$data=fread($fo,filesize($fn));
fclose($fo);
$data1=chunk_split(base64_encode($data));
//$h="Content-Type:text/plain\r\n";
//$h.="Content-Transfer-Encoding:7bit".$msg."\r\n";
$h="Content-Type:multipart/mixed\r\n";
$h.="Content-Disposition:attachment;filename=$fn\r\n";
$h.="Content-Transfer-Encoding:base64".$data1."\r\n";
$m=mail($to,$sub,$msg,$h);
if($m)
echo "success";
else
echo "fail";
```