

File Handling in PHP

- In PHP File System allows us to create file, read file line by line, read file character by character, write file, append file, delete file and close file.
- PHP supports following functions to handle(Create, Read, Write, Delete and Close) files.
 - fopen()
 - fread()
 - fwrite()
 - fclose()
 - unlink()

1. fopen() function

- fopen() function is used to open file or URL.
- The fopen() function accepts two arguments: **\$filename** and **\$mode**.
- The **\$filename** represents the file to be opened and **\$mode** represents the file mode for example read-only, read-write, write-only etc.

– Syntax:

fopen (string \$filename , string \$mode)

- Supported modes are:

r - Read Only **r+** - Read & Write

w - Write Only **w+** - Read & Write

a - Write Only(Append).

Example:

```
<?php
```

```
$filename = "D:\\madhu\\file.txt";
```

```
$fp = fopen($filename, "r");//open file in read mode
```

```
?>
```

2. fread() function

- fread() function is used to read data of the file. It requires two arguments: **file resource** and **file size**.

-Syntax:

fread (resource \$fp, int \$length)

\$fp represents file pointer that is created by fopen() function.

\$length represents length of file (bytes)to be read.

Example:

```
<?php
```

```
$filename = "D:\\madhu\\file.txt";
```

```
$fp = fopen($filename, "r");//open file in read mode
```

```
$content = fread($fp, filesize($filename));//read file
```

```
echo "$content";//printing data of file
```

```
fclose($fp);//close file
```

```
?>
```

Note: To read content of file, we need to open file with
r , r+ , w+ mode.

3. fwrite() function:

- fwrite() function is used to write content (string) into file.
- fwrite() function will erase the previous data of the file and writes the new data.
- It requires two arguments: **file resource** and **data(string)**.
 - `fwrite(resource $fp, string $data)`
\$fp represents file pointer that is created by fopen() function.
\$data represents text to be write.
- To write data into file, you need to use **w,r+,w+** mode

Example:

```
<?php
```

```
$filename = "D:\\madhu\\file.txt";
```

```
$fp = fopen($filename, "w");//open file in write mode
```

```
fwrite($fp,'PHP is a Scripting');
```

```
fclose($fp);//close file
```

```
?>
```

o/p: PHP is a Scripting

Example:

```
<?php
```

```
$filename = "D:\\madhu\\file.txt";
```

```
$fp = fopen($filename, "a");//open file in write mode
```

```
fwrite($fp,'language at server side');
```

```
fclose($fp);//close file
```

```
?>
```

o/p: PHP is a Scripting language at server side

4.fclose() function:

- fclose() function is used to close an open file.
- It requires one argument i.e. “File Pointer”.

– Syntax:

`fclose ( resource $fp )`

\$fp represents file pointer that is created by fopen() function.

Example:

```
<?php
```

```
fclose($fp);
```

```
?>
```

5.unlink() function:

- In PHP, we can delete any file using unlink() function. The unlink() function accepts one argument only: file name.

– Syntax:

`unlink ( string $filename )`

\$filename represents the name of the file to be deleted.

Example:

```
<?php
$status=unlink('data.txt');
if($status){
    echo "File deleted successfully";
}else{
    echo "Sorry!";
}
?>
```

File Upload

File Upload

- PHP allow you to upload any type of a file i.e. image, binary or text files.etc.,
- PHP has one in built global variable i.e. **\$_FILES**, it contains all information about file.
- By the help of **\$_FILES** global variable, we can get file name, file type, file size and temp file name associated with file.
- In HTML, File Upload control can be created by using following:
 - `<input type="file" name="fileupload"/>`

- `$_FILES['filename']['name']`
returns file name.
- `$_FILES['filename']['type']`
returns MIME type of the file.
- `$_FILES['filename']['size']`
returns size of the file (in bytes).
- `$_FILES['filename']['tmp_name']`
returns temporary file name of the file which was stored on the server.

- **move_uploaded_file() function**

- ❖ The move_uploaded_file() function moves the uploaded file to a new location.
- ❖ It moves the file if it is uploaded through the POST request.

Syntax

move_uploaded_file (string \$filename , string \$destination)

- ✓ **Configure The "php.ini" File**

- First, ensure that PHP is configured to allow file uploads.
- In your "php.ini" file, search for the **file_uploads** directive, and set it to **"On"**

i.e. **file_uploads = On**

“Fileupload.php”

```
<html>
```

```
<head>
```

```
<title>File Upload</title>
```

```
</head>
```

```
<body>
```

```
<form method="post" enctype="multipart/form-data">
```

Select File:

```
<input type="file" name="fileToUpload"/>
```

```
<input type="submit" value="Upload Any File"  
  name="submit"/>
```

```
</form>
```

```
<?php
```

```
if(isset($_POST["submit"]))
```

```
{
```

```
$target_path = "D:/";
```

```
$target_path=$target_path.basename($_FILES['fileToUpload'] ['name'] );
```

```
if(move_uploaded_file($_FILES['fileToUpload']['tmp_name'],  
    $target_path))
```

```
{
```

```
    echo "File uploaded successfully!";
```

```
}
```

```
else
```

```
{
```

```
    echo "Sorry, file not uploaded, please try again!";
```

```
}
```

```
}
```

```
?>
```

```
</body>
```

```
</html>
```

Output:

