# PHP File Handling

Dr. Jenila Livingston L.M.
VIT Chennai

## File Handling

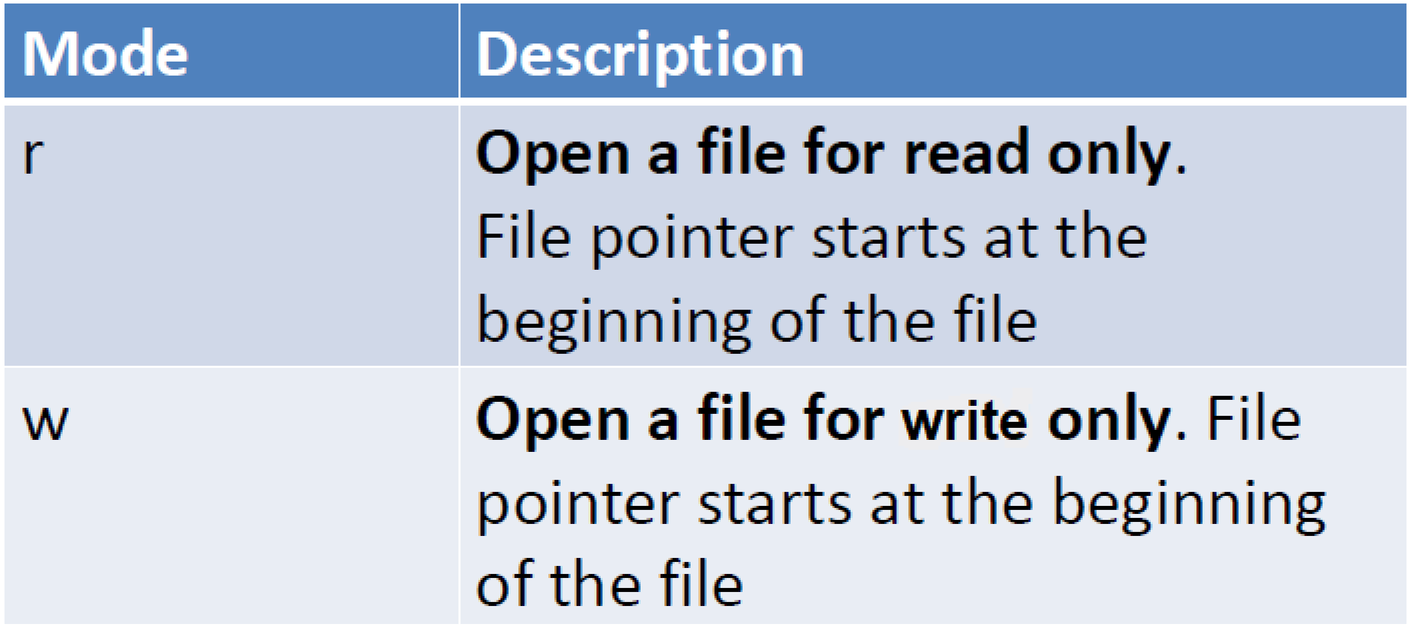
- This involves 5 tasks
  1. Opening a file
  2. Reading data from a file
     - Displaying the read data
  3. Writing contents to another file
  4. Closing a file
  5. File Uploading

## 1. Opening a file

- $fp = fopen(‘filename’,’mode’);
- E.g.

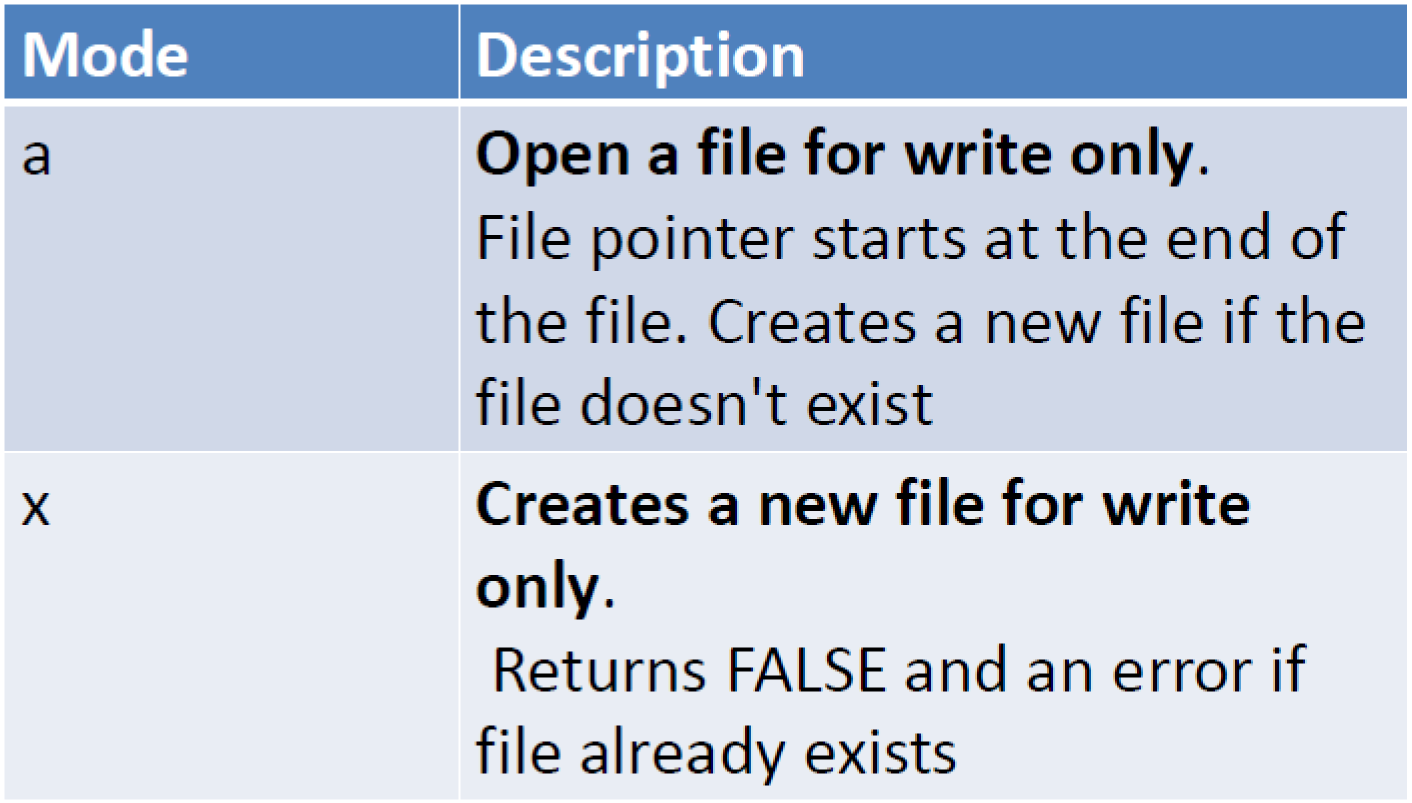
  $fp = fopen(‘c:\abc.txt’,’r’);
  - This opens a file abc.txt in read only mode
- Available modes:
  - r – read only
  - w – write only
  - r+ / w+ - read write
  - a – append – adding to the end
  - x- write by creating new file

# File modes and description



| Mode | Description |
|---|---|
| r | **Open a file for read only.** File pointer starts at the beginning of the file |
| w | **Open a file for write only**. File pointer starts at the beginning of the file |

# File modes and description



| Mode | Description |
|---|---|
| a | **Open a file for write only.** File pointer starts at the end of the file. Creates a new file if the file doesn't exist |
| x | **Creates a new file for write only.** Returns FALSE and an error if file already exists |

## File modes and description

| Mode | Description |
|---|---|
| r+ | **Open a file for read/write.** File pointer starts at the beginning of the file |
| w+ | **Open a file for read/write.** Erases the contents of the file or creates a new file if it doesn't exist. File pointer starts at the beginning of the file |

# fopen

- <?php
- $myfile = fopen("abc.txt", "r") or die("Unable to open file!");
- echo '<b>'. fread($myfile,filesize("abc.txt")).'</b>';
- fclose($myfile);
- ?>

## 2. Reading a file

- Several methods are available
  - fread(filepointer,no of bytes to read)
  - fgetc(filepointer) – Reads character by character
  - fgets(filepointer) – Reads line by line
- The read content can be stored in a variable
- $data = fread($fp,10) – this reads 10 characters from file pointed by file pointer $fp and stores in $data
- If we want to read characters till end, we need to use a loop with condition checking for End of File

## Reading a File – fgets()

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

```
<?php
$myfile = fopen(“abc.txt", "r") or die("Unable to
   open file!");
echo fgets($myfile);
fclose($myfile);
?>
```

## feof()

- feof(fp) – Checks for end of file.
- Returns –1 if EOF is reached. Otherwise returns 0

```
<?php
$myfile = fopen("abc.txt", "r") or die("Unable to
   open file!");
while(!feof($myfile)) {
echo fgets($myfile) . "<br>";
}
fclose($myfile);
?>
```

- $file = "abc.txt";
- // read file contents into string
- $str = **file_get_contents**($file) or die("Can not read from file");
- **echo** $str ."<br>";
- // read file into array
- **$array = file($file)** or die("Can not read from file");

## 3. Writing to file

- We can use echo $data, to print the contents read from the file to browser
- Or we can open another file in write mode and put the contents to that file using either of these methods
  - fwrite(filepoiner,data);
  - fputc(filepointer,char); - writes character by character
  - fputs(filepointer,line); - writes line by line
- Eg - fwrite($fpw,$data);

## Writing a file – fwrite()

- The fwrite(filename, text to be written) function is used to write to a file.

```
$myfile = fopen(“abc.txt", "w") or die("Unable to
   open file!");
$txt = "File Concepts\n";
fwrite($myfile, $txt);
fclose($myfile);
```

## 4. Closing a file

- To close a file use fclose(filepointer) method
- Eg. fclose($fp);
  - This closes the file pointed by $fp.

## file_exists and filesize()

```
<?php
if (file_exists("abc.txt"))
echo 'file exists';
else
echo 'filedoes not exist';
echo filesize("abc.txt");
?>
```

## Delete a file

```
<?php
if (file_exists("abc.txt"))
echo 'file exists';
else
echo 'filedoes not exist';
echo filesize("abc.txt");
unlink("abc.txt");
if (file_exists("abc.txt"))
echo 'not deleted';
else
echo 'got deleted';
?>
```

## File Uploading

In your "php.ini" file, search for the file_uploads directive, and set it to On:

```
file_uploads = On
```

## Five Variables

1. **$_FILES['file']['name']** – the actual name of the uploaded file.
2. **$_FILES['file']['tmp_name']** – the uploaded file in the temporary directory on the web server.
3. **$_FILES['file']['size']** – the size in bytes of the uploaded file.
4. **$_FILES['file']['type']** – the MIME type of the uploaded file.
5. **$_FILES['file']['error']** – the error code associated with this file upload.

# File Uploading - Steps

## Step 1: Create The HTML Form

Some rules to follow for the HTML form :

- Make sure that the form uses **method="post"**
- The form also needs the following attribute: **enctype="multipart/form-data".** It specifies which content-type to use when submitting the form

## Step 2: Create The Upload File PHP Script

- **$target_dir** = "uploads/" - specifies the directory where the file is going to be placed
- **$target_file** specifies the path of the file to be uploaded
- **$imageFileType** holds the file extension of the file

## Step 3: Upload the file

**move_uploaded_file** – upload file

## Create HTML form (fileup.html)

```
<html><body>
<form action="fileup1.php" method="post"
  enctype="multipart/form-data">
 Select File to Upload:
 <BR><input type="file" name="fileToUpload"
  id="fileToUpload">
  <BR><input type="submit" value="Upload File"
  name="submit">
</form>
```

# File upload (fileup1.php)

```
<?php
$target_dir = "e:/uploads/";

$target_file = $target_dir . basename($_FILES["fileToUpload"]["name"]);

$imageFileType = pathinfo($target_file,PATHINFO_EXTENSION);

move_uploaded_file($_FILES["fileToUpload"]["tmp_name"], $target_file);

echo "The file". basename( $_FILES["fileToUpload"]["name"]). " has been
    uploaded successfully";?>

?>
```

# Check if file already exists

- if (file_exists($target_file)) {
- echo "Sorry, file already exists.";}
- $uploadOk = 0;

## Limit File Size

-
  ```
  // Check file size
  if ($_FILES["fileToUpload"]["size"] > 500000) {
      echo "Sorry, your file is too large.";
        $uploadOk = 0;
  }
  ```

- If the file is larger than 500KB, an error message is displayed, and $uploadOk is set to 0

## Limit File Type

- // Allow certain file formats
  **$imageFileType** =
  **pathinfo**($target_file,**PATHINFO_EXTENSION**);
- if($imageFileType != "jpg" && $imageFileType
  != "png" && $imageFileType != "jpeg"
  && $imageFileType != "gif" ) {
      echo "Sorry, only JPG, JPEG, PNG & GIF files
  are allowed.";

$uploadOk = 0;
  }

- // Check if $uploadOk is set to 0 by an error
  if ($uploadOk == 0) {
    echo "Sorry, your file was not uploaded.";
  // if everything is ok, try to upload file
  } else {
    if (move_uploaded_file($_FILES["fileToUploa
  d"]["tmp_name"], $target_file)) {
      echo "The file ". htmlspecialchars(
  basename( $_FILES["fileToUpload"]["name"])).
   " has been uploaded.";}

## Reference

- https://www.w3schools.com/php/php_file_upload.asp