IT1100

Internet and Web technologies

Lecture 07 – part II

PHP - File Handling

Faculty of Computing

SLIIT

Discover Your Future

SLIIT
Discover Your Future

Lecture 07 – part II
PHP - File Handling

IT1100 Internet and Web technologies

Content

File handling

1. File upload
2. File creation
3. File read
4. File write
5. Directory/Folder operations

1. File Handling

- Files are external resources, therefore, processing the files and content in files can cause runtime exceptions
- Applications can perform upload/download, create, name/rename, delete operations on files
- Applications can perform Create, Read, Update, Delete (CRUD) operations on the content of the files in the server

2. File Handling
File upload (Client-side)

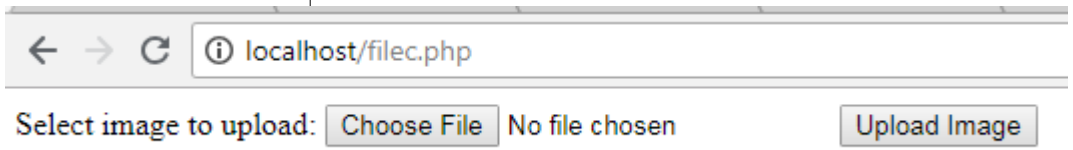
- To upload file(s),a form is used in the client-side, with a file upload field

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<form action="file.php"  method="post" enctype="multipart/form-data">
Select image to upload:
        <input type="file"  name="fileFieldName"  id="fileToUpload">
        <input type="submit"  value="Upload Image"  name="submit">
</form>

</body>
</html>
```

Output



2. File Handling
File upload (Server-side)

- To access the uploading files, the **$_FILES[ ]** superglobal variable is used
  - **$_FILES[“fileFieldName”]**

- The availability of the uploading file can be checked
  - **isset($_FILES["fileFieldName"])**

- The maximum file size can be configured
  - **Php.ini** - **upload_max_filesize=50M**

2. File Handling
Server-side– useful properties

- The name of the file
  - **$_FILES[“fileFieldName”][“name”]**

- The file will be first uploaded into a temporary location with a temporary name
  - **$_FILES[“fileFieldName”] ["tmp_name"]**

- The size of the file
  - **$_FILES[“fileFieldName”] ["size"]**

2. File upload

Server-side– useful functions

- The pathinfo() function returns an array that contains information about a path.
  - **print_r(pathinfo("/testweb/test.txt"));**
- To check if the file already exists
  - **file_exists($target_file)**
- To move the file from the temporary location to the actual location
  - **move_uploaded_file($_FILES["fileToUpload"]["tmp_name"], $target_file)**

2. File upload

Server-side complete code

```
<!DOCTYPE html>
<html>
<body>
<?php
$target_dir = "uploads/";
$target_file = $target_dir . basename($_FILES["fileFieldName"]["name"]);
if(isset($_FILES["fileFieldName"])) {
        if (move_uploaded_file($_FILES["fileFieldName"]["tmp_name"],$target_file)){
                echo "The file ". basename( $_FILES["fileFieldName"]["name"]). " is uploaded.";
        }
        else {
                echo "Error while uploading your file.";
        }
}else{
        echo "File not available";
}
?>
</body>
</html>
```

2. File upload
Server-side– validation

- It is always good to validate the file against different properties before saving it
  - Extension, size, existence, file format, etc…

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

3. File creation

- It is good to test if the file exists, before performing file processing

```
<?php
   echo file_exists("test.txt");
?>
```

- Under this section we will learn
  - How to create file
  - How to rename file
  - How to remove file

3. File creation
Create file

```
<?php
  $file = fopen("test.txt","r");
  $file = fopen("/home/test/test.txt","r");
  $file = fopen("http://www.example.com/","r");
  $file = fopen("ftp://user:password@example.com/test.txt","w");
?>
```

- If the file already exists, then it will be opened for writing, where if it is not existing, a new file will be created

- “w” = write mode.
  - To create a file, the mode should be write

3. File creation

Rename a file

rename(currentName, newName)

- E.g
  - **rename(“test.txt”,”best.txt”);**

- This function returns a Boolean value

3. File creation

Remove a file

unlink(fileName)

- E.g
  - **unlink(“test.txt”);**

- This function returns a Boolean value

4. File read

There are 2 ways of reading a file

1. Read the complete file

2. Read part by part

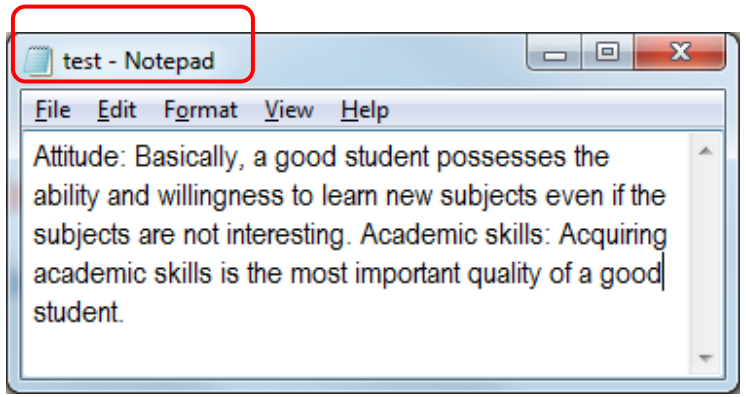
4. File read

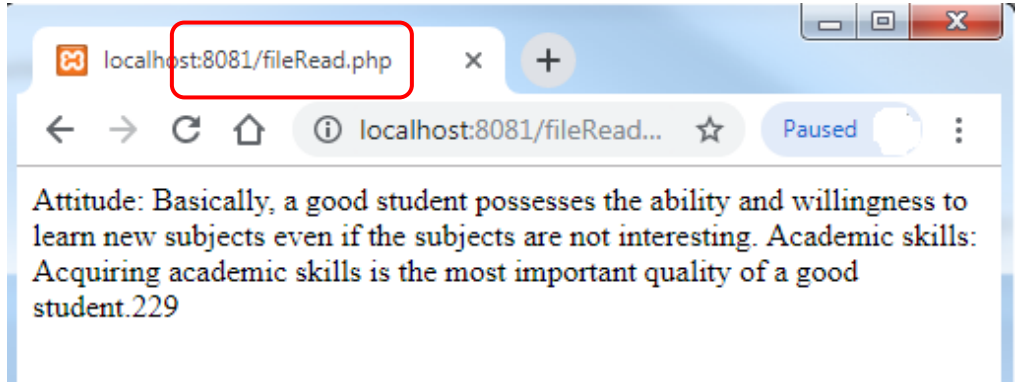
Read the complete file

```
<?php
        echo readfile("test.txt");
?>
```

- This will return the content of the file

Output





4. File read
Open file / Close file

$fileHandler = fopen(“fileName.txt", “MODE“) or die(“Error!");

- This returns a file handler
- There are different **MODE**s (next slide)

fclose($fileHandler);

- Close the file handler, after using the file

4. File read

Open file - MODE

Modes	Description
r	Open a file for read only.
w	Open a file for write only.
a	Open a file for write only.
x	Creates a new file for write only.
r+	Open a file for read/write.
w+	Open a file for read/write.
a+	Open a file for read/write.
x+	Creates a new file for read/write.

https://www.w3schools.com/php/php_file_open.asp

4. File read

Read the content

fread($fileHandler, filesize(“fileName.txt"));

- Reads the whole file

fgets($fileHandler);

- Reads a single line

fgetc($fileHandler);

- Reads a single character

4. File read

Read the content – lines and chars

- When reading a file line by line or char by char, we can use a loop to traverse the file
- The loop may iterate till the **end of file**

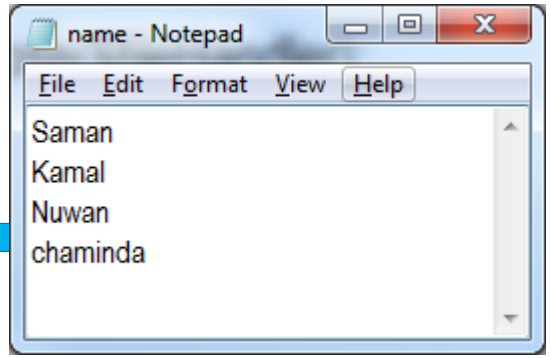
```
$ fileHandler = fopen(“fileName.txt", "r") or die(“Error");
while (!feof($fileHandler)) {
        echo fgetc($fileHandler);
}
fclose($fileHandler);
```

4. File read
Read the content

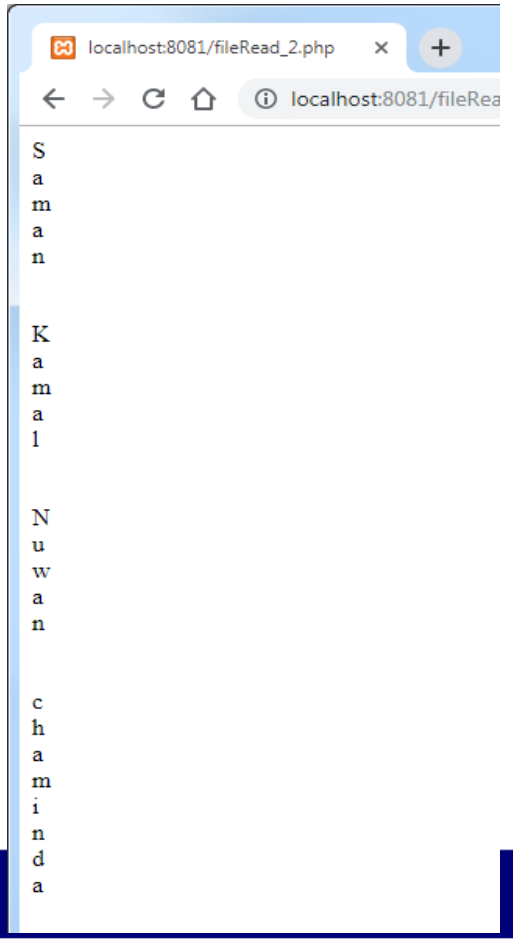
```
<!DOCTYPE html>
  <html>
  <body>
    <?php

    $fileHandler=fopen("Name.txt","r")or die("Error");

    while(!feof($fileHandler)){
       echo fgetc($fileHandler);
    }
       fclose($fileHandler);
    ?>
    </body>
</html>
```



Output



5. File write

- To write into a file, the file should be opened using a writable mode
- There are 2 writable modes
  1. w – replace/overwrite
  2. a – append

5. File write

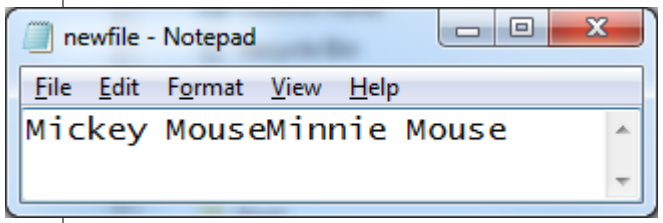
```
<?php
    $FH = fopen(“myFile.txt", "w") or die(“Error");
    fwrite($FH , “Hello\n”);
    fwrite($FH , “World\n”);
    fclose($FH );
?>
```

- If there is some content in the file, they will be overwritten
- If the mode is “**a**”, then the new content will be appended at the end of the current content

5. File write

```
<!DOCTYPE html>
  <html>
  <body>
    <?php
      $myfile = fopen("newfile.txt", "w") or die("Unable to open file!");
      $txt = "Mickey Mouse\n";
      fwrite($myfile, $txt);
      $txt = "Minnie Mouse\n";
      fwrite($myfile, $txt);
      fclose($myfile);
  ?>
</body>
</html>
```

Output



6. Directory/folder operations

- Often, when processing files, the directories are also needed to be processed

is_dir($target_dir)

- Checks if the target is a directory, and it can be used to check if the directory exists
- Returns a Boolean value

6. Directory/folder operations

Content in a directory

- There are two ways of accessing the content
  1. **scandir(“dirPath”)** — List files and directories inside the specified path
  2. **glob(“pattern”)** — Find pathnames matching a pattern

- Both functions return the output as an array

6. Directory/folder operations
scandir()

- $**dir = "/dirName/";**

  // Sort in ascending order - this is default
  $a = scandir($dir);

  // Sort in descending order
  $b = scandir($dir,1);

6. Directory/folder operations

scandir()

```
<!DOCTYPE html>
<html>
<body>
    <?php
        $dir = "uploads/";
        // Sort in ascending order - this is default
        $a = scandir($dir);
        // Sort in descending order
        $b = scandir($dir,1);
        print_r($a);
        print_r($b);
    ?>
</body>
</html>
```

Output

```
Array ( [0] => . [1] => .. [2] => 123 [3] => 4.html )
Array ( [0] => 4.html [1] => 123 [2] => .. [3] => . )
```

6. Directory/folder operations

glob()

$a = glob("*.txt")

- Get all the .txt files in the current directory

$a = glob("**myDir**/*.*")

- Get all the files in **myDir** directory

6. Directory/folder operations

glob() – use a foreach loop

Get all the text files in the current directory

```
$a = glob("*.txt");
foreach ($a as &$value)
{
        echo $value;
}
```

6. Directory/folder operations

glob() – use a foreach loop

EX: Do these on the same page

1. Prepare a list of hyperlinks for the text files in the “myDocs” directory
2. Show all the images (.jpg files) in the “images” directory

Summary

1. File upload
2. File creation
3. File read
4. File write
5. Directory/folder operations