Lecture 6 – ICT1233

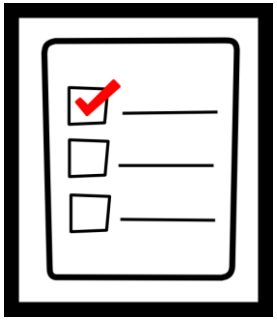
PHP File Handling

Department of ICT

Faculty of Technology

Objective

- After the successful completion of this lecture, students should be able to,
  - Identify all the types of file manipulations
  - Handle files using php scripts



Introduction

- Types of file manipulation
  - Create, Open, Close
- Other file operations
  - Read, Write, Append, Truncate, Upload
- **Note**
  - When manipulating files you must be very careful as you can do a lot of damage if you do something wrong

fopen() Modes – Type of Access

Modes	Description
r	**Open a file for read only**. File pointer starts at the beginning of the file
w	**Open a file for write only**. 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
a	**Open a file for write only**. The existing data in file is preserved. 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
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
a+	**Open a file for read/write**. The existing data in file is preserved. 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 read/write**. Returns FALSE and an error if file already exists

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

```
$my_file = 'file.txt';
//implicitly creates file
$handle = fopen($my_file, 'w') or die('Cannot
open file: '.$my_file);
```

Write in a File

- The **fwrite()** function is used to write to a file
- The first parameter of **fwrite()** contains the handler of the file to write to and the second parameter is the string to be written

```
$my_file = 'file.txt';
$handle = fopen($my_file, 'w') or die('Cannot open
file: '.$my_file);
 $data = 'This is the data';
fwrite($handle, $data);
```

Closing a File

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

```
$my_file = 'file.txt';
$handle = fopen($my_file, 'w') or die('Cannot open file:
'.$my_file);
//write some data here
fclose($handle);
```

PHP fread() Function

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

Syntax

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

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

PHP fread() Function - Example

```
<html>
        <head>
                <title>Reading a file using PHP</title>
        </head>
        <body>
                <?php
                        $filename = "tmp.txt";
                        $file = fopen( $filename, "r" );
                        if( $file == false ) {
                                echo ( "Error in opening file" );
                                exit();
                }
                 $filesize = filesize( $filename );
                 $filetext = fread( $file, $filesize ); fclose( $file );
                        echo ( "File size : $filesize bytes" );
                        echo ( "<pre>$filetext</pre>" );
                ?>
        </body>
</html>
```

PHP fwrite() Function - Example

```php
<?php
        $filename = "/home/user/guest/newfile.txt";
        $file = fopen( $filename, "w" );
        if( $file == false ) {
                echo ( "Error in opening new file" );
                exit();
        }
        fwrite( $file, "This is a simple test\n" );
        fclose( $file );
?>
```

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

Syntax:

bool feof (resource $handle)

Check the End-of-file

Syntax:

bool feof (resource $handle)

- Tests for end-of-file on a file pointer
- Returns **TRUE** if the file pointer is at EOF, otherwise returns **FALSE**

```
<?php

    $file = “name.txt”;
    ……….
    if (feof($file)) echo "End of file";

?>
```

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 - Example

```
<?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***

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);
?>
```

Delete a File

- The unlink() function deletes a file.
- This function returns TRUE on success, or FALSE on failure

```
$my_file = 'file.txt';
unlink($my_file);
```

Summary

1. Create a File: fopen()
2. Open a File: fopen()
3. Read a File: fread()
4. Write to a File: fwrite()
5. Append to a File: fwrite()
6. Close a File: fclose()
7. Delete a File: unlink()

Questions…