Week 10: Files

CIS 086 • PHP and MySQL • Mission College

Upcoming Topics

- This week: Files
- Next week: MySQL from the command line
- Then: MySQL from within PHP
- December: Cookies and Sessions

Drop the class?

- The last day is November 18.
- For some reason, this is a Saturday.
- If you may need to do this in person at the records office, make sure to do it during the week before Saturday.

- http://www.missioncollege.edu/class_schedule/2017_summer-fall/documents/calendar_summer-fall_2017.pdf

Chapter 7: Files

- Reading and writing file streams (tricky the first time)
- Reading and writing entire files (easy)
- Copying, Moving / Renaming, and Deleting files
- Locking files
- Uploading files (tricky)
- File Types (easy)
- Directories
- Permissions (tricky)

File Types

- Text vs. Binary
- Line Endings
  - Unix
  - Mac
  - Windows
- A file with the wrong line endings will still show up properly as a web page.
  - But it may show up strangely when you try to view or edit the source code: double spaced or all togerher on one line.

Reading Files all at once

- file ($path) reads the file into an array, with one line per array element.

- file_get_contents ($path) reads the file into one long string. This method can be easier for searching the file.

- readfile ($path) reads the file and prints it into your HTML page.
  - Equivalent to doing file_get_contents() then echo.

Writing Files all at once

- file_put_contents ($path, $string) writes the contents of the string to a file.
  - This is useful if you previously read the file using file_get_contents and you just want to append another line at the end.

- There is no equivalent to write an array to a file all at once. You have two choices:
  - Combine the array elements into a string using join() or implode(), then write the resulting string to the file.
  - Use a file stream and fwrite() to write the array elements to the file one at a time.

File Streams

- fopen ($path, $mode)
  - Mode can be read, write, append, or others
  - Sometimes you can use fopen to read from a URL

- fgets () = read one line from the file
- fread () = read a certain number of bytes from the file
- fwrite () = write a string to the file

- fclose () = close the file when you are done
  - Important!

Other useful file functions

- file_exists ($path)
  - Make sure the file exists before trying to read from it.
  - Make sure the file doesn't exist before possibly overwriting it.

- is_file ($path) = is it a regular file?
- is_dir ($path) = is it a directory?
- is_link ($path) = is it a symbolic link?

- is_readable ($path) = exists and has read permissions
- is_writeable ($path) = exists and has write permissions

- stat ($path) = an array that includes userid, groupid, file size, modification time, etc.

More useful functions

- copy ($from, $to)
- rename ($from, $to)
- unlink ($path) // remove

// lock a file so you have exclusive access for a short time

- flock ($handle, LOCK_EX)
- flock ($handle, LOCK_UN)

Directory Functions

- http://us3.php.net/manual/en/ref.dir.php

The easy way:

- scandir (path)

The hard way:

- opendir (path)
- readdir (handle)
- closedir (handle)

Permissions (1 of 5)

- Three one-bit flags:
  - 4 = read
  - 2 = write
  - 1 = execute

- Add them together (or bitwise “or”):

Number	Read	Write	Execute
7	x	x	x
6	x	x	
5	x		x
4	x		
3		x	x
2		x	
1			x
0			

File Permissions (2 of 5)

- When you create a file or folder manually, by choose “New Folder” or using Brackets to create a file, you are the owner and you have the default owner permissions for the file.
  - File = Read and Write permissions
  - Folder = Read, write, and execute permissions (so you can look into the folder)
- When a PHP program creates or edits a file, the owner of the file is the Apache web server program. This is consider an “other” user (3rd tier). The default permissions for the “other” user are:
  - File = Read only
  - Folder = Read only

Permissions (3 of 5)

- Folder
  - Write permission means you can ADD a new file to the folder.
  - Execute permission means you can “go” into the folder.

- File
  - Write permission means you can CHANGE the contents of the file.
  - Execute permission means you can execute the file from the command line.

Permissions (4 of 5)

- It is not necessarily easy to change file permissions on your local computer.
  - Windows has its own concept of file permissions that is unique.
  - On Mac, you mostly need to use the Unix command chmod from the command line.
  - The book suggests you use the PHP function chmod() to set file permissions, but this is tedious.
- You can easily change file permissions on the Mission College PHP server using FileZilla.
- Chmod is available from within PHP, but it doesn't always make sense to use it there.

Permissions (5 of 5)

- Your permissions may not be transferred when you upload files using FileZilla or other FTP programs.

- You may have to set the file and directory permissions again after uploading.

File Upload

```
<form  method='post'
       action='upload.php'
       enctype='multipart/form-data'>

<input    type='hidden'
          name="MAX_FILE_SIZE"
          value='100000' />
```

$_FILES

- name = file name without path
- type = MIME type
- tmp_name = path to the file in its temporary upload location
- error = 0 (no error) or an error number
- size = file size in bytes

- move_uploaded_file ($temp, $perm);

- http://php.net/manual/en/reserved.variables.files.php
- http://php.net/manual/en/features.file-upload.errors.php