

File Handling with PHP

Files and PHP

- File Handling:
 - refers to working with files that are stored on a hard disk
- Rationale:
 - Data Storage
 - Can store data in “flat” files. e.g. logs, dist_lists, xml
 - Note: flat file processing tends to be slower than a database data retrieval using SQL
 - Manipulating uploaded files
 - The files can be uploaded through form input of type “file”
 - Creating Files for download

To Open/Close a File

- A file is opened with `fopen()` as a “stream”, and PHP returns a ‘handle’ to the file that can be used to reference the open file in other functions.
- Each file is opened in a particular **mode**.
- A file is closed with `fclose()` or when your script ends.
- NOTE: the web server needs to have permission to read and write to a file
 - You should `chmod` a file to 0760, giving the file owner full permission and the group (www) read and write permissions.
 - The folder that the file exists in should be 0770, giving the file owner and the group (www) full permissions.
 - The “group” on the opentech server is “www” which is in reality the Apache web server (the software that is running PHP)

File Open Modes

<code>'r'</code>	Open for reading only. Start at beginning of file.
<code>'r+'</code>	Open for reading and writing. Start at beginning of file.
<code>'w'</code>	Open for writing only. Remove all previous content, if file doesn't exist, create it.
<code>'a'</code>	Open writing, but start at END of current content (create file if necessary)
<code>'a+'</code>	Open for reading and writing, start at END and create file if necessary.

File Open/Close Example

```
<?php
// open file to read
$storead = fopen('some/file.ext','r');
// open (possibly new) file to write
$towrite = fopen('some/file.ext','w');
// close both files
fclose($storead);
fclose($towrite);
?>
```

Once the File is Open

- If you open the file to write, there are pre-defined PHP provided functions to write into a file stream.
- If you open a file to read, there are pre-defined PHP provided functions to read content from a file stream

Writing Data

- To write data to a file use:
- `fwrite($handle, $data)`
 - Write \$data to the file to the open file stream named \$handle.

Data Writing Example

```
<?php
```

```
    $handle = fopen( 'contacts.txt', 'a' );
```

```
    fwrite( $handle, "\nJohn Doe:\tjdoe@gmail.com" );
```

```
    fclose( $handle );
```

```
?>
```


Data Writing Example

```
<?php
```

Open file to append data (mode 'a')

```
$handle = fopen( 'contacts.txt', 'a' );
```

```
fwrite( $handle, "\nJohn Doe: \tjdoe@gmail.com" );
```

Write new data (with a line break after previous data and a tab in between)

```
fclose( $handle );
```

```
?>
```

Other File Operations

- Delete file
 - `unlink('filename');`
- Rename (file or directory), aka move
 - `rename('old name' , 'new name');`
- Copy file
 - `copy('source' , 'destination');`
- There are more, for a complete listing:
 - www.php.net/manual/en/ref.filesystem.php

Reading Data

- There are two main functions to read data:
- **fgets** (\$handle , \$bytes)
 - Reads up to \$bytes of data, stops at newline or end of file (EOF)
- **fread** (\$handle , \$bytes)
 - Reads up to \$bytes of data, stops at EOF.

Reading Data

- **end-of-file**, commonly abbreviated **EOF**, is a condition in a computer operating system where no more data can be read from a data source.
- To read a file using PHP the EOF has to be checked for and handled
- **feof (\$handle)**
 - Whether the file has reached the EOF point. Returns true if have reached EOF.

Data Reading Example

```
<?php
```

```
    $handle = fopen('contacts.txt', 'r');
```

```
    while (!feof($handle)) {  
        echo fgets($handle, 1024);  
        echo '<br />';  
    }
```

```
    fclose($handle);
```

```
?>
```

Data Reading Example

```
<?php
```

Open the file and assign the resource to \$handle

```
$handle = fopen('contacts.txt', 'r');
```

```
while (!feof($handle)) {  
    echo fgets($handle, 1024);  
    echo '<br />';  
}
```

```
fclose($handle);
```

```
?>
```

Data Reading Example

```
<?php
```

```
$handle = fopen('contacts.txt', 'r');
```

```
while (!feof($handle)) {  
    echo fgets($handle, 1024);  
    echo '<br />';  
}
```

```
fclose($handle);
```

```
?>
```

While NOT at the end of the file, pointed to by \$handle, get and echo the data line by line up to 1024 bytes, separate lines with

Data Reading Example

```
<?php
```

```
$handle = fopen('contacts.txt', 'r');
```

```
while (!feof($handle)) {  
    echo fgets($handle, 1024);  
    echo '<br />';  
}
```

```
fclose($handle);
```

Close the file stream

```
?>
```


File Open Shortcuts

- There are two 'shortcut' functions that don't require a file to be opened:
- `$lines = file($filename)`
 - Reads entire file into an array with each line a separate entry in the array.
- `$str = file_get_contents($filename)`
 - Reads entire file into a single string.

Class Example -Write

Build a self-referring form around the example below as a practice. BE SURE TO CREATE THE FILE BASED ON TODAY'S DATE IN AN *attendance* SUB-FOLDER

```
if( $_SERVER["REQUEST_METHOD"] == "POST"){ //when the form is submitted it will be in "POST"
mode
//retrieve info from the form
$student_number = trim($_POST['number']);
$first_name = trim($_POST['first']);
$last_name = trim($_POST['last']);
$check = trim($_POST['secret']);

//set up time dependent stuff
$today = date("Ymd");
$now = date("Y-m-d G:i:s");

//here is the file stuff
$handle = fopen("./attendance/" . $today . ".txt", 'a');

fwrite($handle, $now . "-" . $student_number . " - " . $first_name . " " . $last_name . " " . $check . "\n");

fclose($handle);
$message .= $now . "-" . $student_number . " - " . $first_name . " " . $last_name . " was added to the
log<br/>";
$message .= "Click <a href=\"./file2.php\">here</a> to see who is in class"; //echo
$message on page
}
```

Class Example - Read

Build a page around the code below, and as long as the file exists you should get the file content out on the page

```
$message = "";
$today = date("Ymd");

//here is the file stuff
$handle = fopen("./attendance/" . $today . ".txt", 'r');

while (!feof($handle)) {
    $message .= fgets($handle, 1024);
    $message .= "<br />\n";
    //echo the message somewhere on the page
}
fclose($handle);
```