PHP: Part 6
Working with External Files

Writing to files

- Writing to an external files requires three steps:
  - open the file for writing
  - write to the file
  - close the file

fopen() Modes

- The modes that we will use to open the files:
  - a to append to a file – to add data to a file and create it if necessary
  - w to write to a file; this will delete a current file by the same name
  - r to read only, beginning at the start of the file.

Sample code to write to a file:

```
$fp = fopen('museums.txt',w);
fwrite($fp,”MoMA”);
fclose($fp);
```

- Or use variables:

```
$museum1 = "Metropolitan Museum of Art\n";
$fp = fopen('museums.txt',w);
fwrite($fp,$museum1);
fclose($fp);
```

PHP: Reading from a file vs. include() or require()

- One can use the file() and related functions in PHP to read the contents of a file for the purpose of further processing, printing, etc.
    - For example, one could use the file() function to open a file of stock prices and use the data to print and/or calculate statistical results such as the high, low, average, etc.
- This is different from using include() or require() in which the contents of the file are copied into the PHP code of the page that is currently being created.

To read data from a file

- Use file() to open a file’ s contents into an array:

```
$tobe = file('ShakespeareHamlet.txt');
print "<p>";
for ($n=0; $n<count($tobe); $n++) {
    print trim($tobe[$n])."<br />\n";
}
```

Using PHP to create a .php file

- It is common to write (or append) to an external file in PHP for the purpose of using include() with that file later on. For example:
  - create a .php file with functions
  - create a .php file with “global” or “local” variables that come from a source outside of the current script
  - create a .php file to include JavaScript
  - create a file to save the contents of $_SESSION
  - … and many other reasons.

Creating a .php file to include()

- Note that PHP parsing stops when a file is used with include() as PHP assumes the contents to be HTML.
- Therefore, be sure to use <?php … ?> at the beginning and end of a file that you plan to use with include().

Sample code

- Notice that the <?php … ?> is needed here:

```
$fp2 = fopen('museum_variables.php',w);
$m1  = "\$museum1 = \"".$museum1 . "\";\n";
fwrite($fp2,"<?php \n");
fwrite($fp2,$m1);
fwrite($fp2, "?> \n");
fclose($fp2);
```

Using include() in this context

- Then, using the include() function will allow the PHP parser to interpret the contents as php code and not as HTML or text:

```
include 'museum_variables.php';
```

Working with data files

- PHP provides a function specifically for .csv files. This is a format readily available on the web and from other sources. (One can also use *FILE / SAVE AS* in MS-Excel or any spreadsheet application to create a .csv file)

  fgetcsv($file, $size, $delimiter)

Additional file handling functions

- is_readable() – to determine if PHP can read the file before you proceed
- is_writable() – to determine if the server will allow you to write to the file.
- use the following to catch errors:

  ```
  if ($fp = fopen( ‘x.txt’ ,’ a’ ) { … }
  else { … error message … }
  ```

- stream_encoding($fp, ‘utf-8’) to set the encoding