PHP

Day 02

Prepared by:
Noha Shehab

Open source dept.

ITI

nshehab@iti.gov.eg

Agenda

- Files
  - Open file
  - Read from file
  - Write to file
  - Close the file
- Arrays
  - Simple arrays
  - Associative arrays
- Forms

Files

- You can store data in two basic ways: in flat files or in a database.
- A flat file can have many formats, but in general, when we refer to a flat file, we mean a simple text file.

PROCESSING FILES

- Writing data to a file requires three steps:
  1. Open the file. If the file doesn't already exist, you need to create it.
  2. Write the data to the file.
  3. Close the file.
- Similarly, reading data from a file takes three steps:
  1. Open the file. If you cannot open the file (for example, if it doesn't exist), you need to recognize this and exit gracefully.
  2. Read data from the file.
  3. Close the file.

Open your first file

- To open a file in PHP, you use the fopen() function. When you open the file, you need to specify how you intend to use it. This is known as the *file mode*.

File modes

You need to make three choices when opening a file:

- You might want to open a file for reading only, for writing only, or for both reading and writing.
- If writing to a file, you might want to overwrite any existing contents of a file or append new data to the end of the file. You also might like to terminate your program gracefully instead of overwriting a file if the file already exists.
- If you are trying to write to a file on a system that differentiates between binary and text files, you might need to specify this fact.

Fopen()

- The fopen() function supports combinations of these three options.

```
$file = fopen( $filename, "r" );
```

Fopen() modes

Mode	Mode Name	Result
r	Read	Open the file for reading, beginning from the start of the file.
r+	Read	Open the file for reading and writing, beginning from the start of the file.
w	Write	Open the file for writing, beginning from the start of the file. If the file already exists, delete the existing contents. If it does not exist, try to create it.
w+	Write	Open the file for writing and reading, beginning from the start of the file. If the file already exists, delete the existing contents. If it does not exist, try to create it.

Fopen() modes

Mode	Mode Name	Result
x	Cautious write	Open the file for writing, beginning from the start of the file. If the file already exists, it will not be opened, fopen() will return false, and PHP will generate a warning.
x+	Cautious write	Open the file for writing and reading, beginning from the start of the file. If the file already exists, it will not be opened, fopen() will return false, and PHP will generate a warning.
a	Append	Open the file for appending (writing) only, starting from the end of the existing contents, if any. If it does not exist, try to create it.

Fopen() modes

Mode	Mode Name	Result
a+	Append	Open the file for appending (writing) and reading, starting from the end of the existing contents, if any. If it does not exist, try to create it.
b	Binary	Used in conjunction with one of the other modes. You might want to use this mode if your file system differentiates between binary and text files. Windows systems differentiate; Unix systems do not. The PHP developers recommend you always use this option for maxi-mum portability. It is the default mode.
t	Text	Used in conjunction with one of the other modes. This mode is an option only in Windows systems. It is not recommended except before you have ported your code to work with the b option.

Open file on read mode

```
$filename = "os.txt";
        $file = fopen( $filename, "r" );
        if( $file == false ) {
           echo ( "Error in opening file" );
           exit(); // from php script
        }
```

Open file on write mode

```
$filename = "os.txt";
        $file = fopen( $filename, “w" );
        if( $file == false ) {
           echo ( "Error in opening file" );
           exit(); // from php script
        }
```

Read from file

- You open the file by using fopen()
- Check the file size
- Open file using fread()
- Close file

```php
// functions that help in opening files
// returns with the file size
$filesize = filesize( $filename );
// reads the content of files
$filetext = fread( $file, $filesize );
fclose( $file );  // close the file
```

Read files functions

- Feof($fp) is used to test for the end-of-file on a file pointer.
- The fgets() function in PHP is an inbuilt function which is used to return a line from an opened file from a file pointer and it stops returning at a specified length.
  - fgets(file, length)
- **What about fgetss?**

Read files functions

- fgetcsv(): This function breaks up lines of files when you have used a delimiting character, such as the tab character or a comma. This function can be used with excel files.
- Reads the file into an array according the specified delimeter
- fgetcsv(file_pointer, length, delimiter, enclosure, escape)

```
$csvfile=fopen('testcsv.xlsx','rb');
$text =fgetcsv ($csvfile , 10 , "\t");
```

Write to a file

- You open the file by using fopen() in write mode
- Use fwrite=== fputs===file_put_contents()
- to write to a file fwrite(file, string, length)
  - length, is the maximum number of bytes to write. If this parameter is supplied, will write string until it reaches the end of string or has written length bytes, whichever comes first.
- Close file

Closing a file

- fclose($fp);

- This function returns true if the file was successfully closed or false if it wasn’t. This process is much less.

Useful File functions

rewind()

- function resets the file pointer to the beginning of the file.

ftell ()

- function reports how far into the file the pointe is in bytes. You can use the function

fseek ()

- to set the file pointer to some point within the file.

Filetype()

- String filetype(string $filepath)
- Check type of file (file, folder)

file_exists():

- Checking Whether a File Is there.

unlink (file)

- Deleting a File

Useful File functions

Problems with files

There are a number of problems in working with flat files:

- When a file grows large, working with it can be very slow.
- Searching for a particular record or group of records in a flat file is difficult. If the records are in order.
- Beyond the limits offered by file permissions, there is no easy way of enforcing different levels of access to data.
- Dealing with concurrent access can become problematic. You can lock files, but it can also cause a bottleneck.
- All of these file processing operations are sequential processing; you start from the beginning of the file and read through to the end.


Arrays

Arrays

- Indexed Arrays
- Associative arrays

Indexed arrays

- The indices of numerically indexed arrays in PHP, start at zero by default, although you can alter this value.
  - $array = array(‘text1’, ‘text2’,‘text3’);
  - $arr=[1,2,3,4]
- If you want an ascending sequence of numbers stored in an array, you can use the range()function to automatically create the array.
  - $numbers = range(1,10);

Range function

- range() function has an optional third parameter that allows you to set the step size between values.
  - $odds = range(1, 10, 2);
- The range() function can also be used with characters, as in this example:
  - $letters = range(‘a’, ‘z’);

Displaying array elements

- Because the array is indexed by a sequence of numbers, you can use a for loop to more easily display its contents:

```
for ($i= 0; $i<count($array); $i++) {
        echo $array[$i]." ";
}
```

Looping across array (foreach)

- foreach ($*array* as $*value*) {
  code to be executed;
  }

Associative arrays

- Associative arrays are key indexed arrays, (Key, Value)
- $alphabets = array(‘a’=>1, ‘b’=>2, ‘c’=>3);
- $ alphabets = array(‘a’=>1);
- $ alphabets[‘b’] = 2;
- $ alphabets[‘c’] = 3;

Create array from variables

- Variables:
  - $username = ‘Noha';
  - $email = ‘nshehab@iti.gov.eg';
- $variables = compact('username', 'email’);
  - // $variables is now ['username' => ‘Noha', 'email' => ‘nshehab@iti.gov.eg']

Looping across associative array

- foreach ($*array* as $*key*->$*value*) {
  echo $key." –".$value."
";
  }

Array operators

- One set of special operators applies only to arrays. Most of them have an analogue in the scalar operators.

Operator	Name	Use	Result
+	Union	$a+$b	Returns an array containing everything in $a and $b
==	Equality	$a == $b	Returns true if $a and $b have the same key and pairs
===	Identity	$a === $b	Returns true if $a and $b have the key and value pairs the same order
!=	Inequality	$a and $b	Returns true if $a and $b are not equal
<>	Inequality	$a or $b	Returns true if $a and $b are not equal
!==	Non-identity	$a x or $b	Returns true if $a and $b are not identical

Multi-dim Arrays

- Arrays do not have to be a simple list of keys and values; each location in the array can hold another array. This way, you can create a two-dimensional array.

Example:

```
$alphabets=
array( array( ‘1st’, ‘a’, 1 ),
array( ‘2nd’, ‘b’, 2 ),
array( ‘3rd’, ‘c’, 3 ) )
```

Sorting Arrays

- $names = array(‘Noha’, ‘Fatma’, ‘Dina’);
- **sort**($names);
- function is case sensitive. All capital letters come before all lowercase letters. So A is less than Z, but Z is less than a.
- The function also has an optional second parameter. You may pass one of the constants SORT_REGULAR(the default), SORT_NUMERIC, or SORT_STRING.

Sorting Associative arrays

- $prices = array(‘meat’=>100, ‘sugar’=>10, ‘tea’=>8);
  - asort($prices);
- The function asort() orders the array according to **the value of each element.**
- If, instead of sorting by value, you want to sort by key, you can use ksort()
  - ksort($prices);

Reverse sort

- rsort()
- arsort()
- Krsort()

REORDERING ARRAYS

shuffle()

- randomly reorders the elements of your array.

array_reverse()

- gives you a copy of your array with all the elements in reverse order.

array_push ()

- to add one new element to the end of an array

array_pop()

- removes and returns one element from the end of an array.

Array flip

- **`array_flip()`** function will exchange all keys with its elements.

```
$colors = array(
'one' => 'red',
'two' => 'blue',
'three' => 'yellow',
);
```

- `array_flip($colors); //will output`

```
array(
'red' => 'one',
'blue' => 'two',
'yellow' => 'three'
```


Dealing with forms

Thank you