Faculty of Science and Technology

Department of Computer Science

WEB TECHNOLOGIES (CSC3215)

Lecture Note 4

Week 4

Supta Richard Philip

Richard@aiub.edu

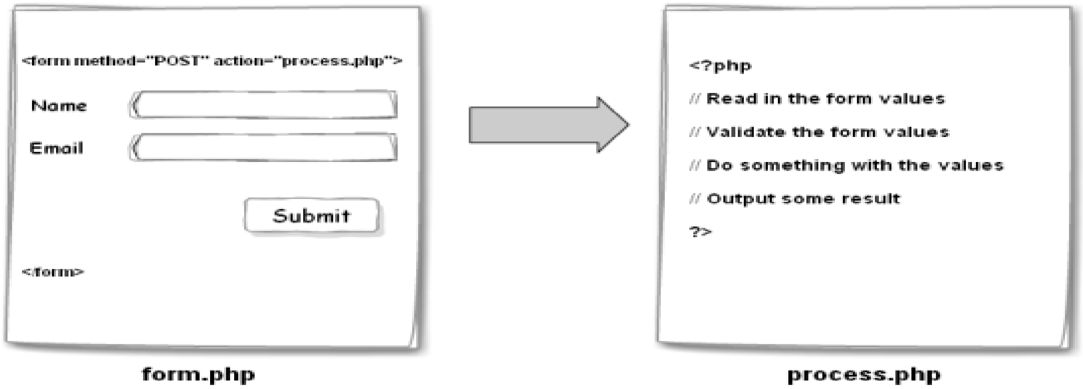
Contents

PHP File Handling

File handling is an important part of any web application. You often need to open and process a file for different tasks.

In this note, we will discuss “read the form data and write to the files” and read data from file. In between, we will assign data to the class variable (object) or Array.

Think about the following sequence:



```
<form method="POST" action="process.php">
<form>
```

```
<?php
// Read in the form values
// Validate the form values
// Do something with the values
// Output some result
?>
```

Data Read / Write in text file

Data Read from text file

The readfile() function reads a file and writes it to the output buffer.

```
<?php
echo readfile("webdictionary.txt");
?>
```

In the **webdictionary.txt** file have the following content:

```
      AJAX = Asynchronous JavaScript and XML
      CSS = Cascading Style Sheets
      HTML = Hyper Text Markup Language
      PHP = PHP Hypertext Preprocessor
```

A better method to open files is with the fopen() function. This function gives you more options than the readfile() function.

```
<?php
$myfile = fopen("webdictionary.txt", "r") or die("Unable to open file!");
echo fread($myfile,filesize("webdictionary.txt"));
fclose($myfile);
?>
```

The first parameter of **fopen()** contains the name of the file to be opened and the second parameter specifies in which mode the file should be opened.

The fclose() requires the name of the file (or a variable that holds the filename) we want to close.

The file may be opened in one of the following modes:

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

Write data in text file

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

```php
<?php
$myfile = fopen("newfile.txt", "w") or die("Unable to open file!");
$txt = "John Doe\n";
fwrite($myfile, $txt);
```

```
$txt = "Jane Doe\n";
fwrite($myfile, $txt);
fclose($myfile);
?>
```

Overwriting data in text file

For example, a file contains some data, if we want to write into that file, all the existing data will be ERASED when we open an existing file for writing. We can use different mode to handle different situation.

PHP SimpleXML

Read from xml File

- Assume we have an XML file called "note.xml", that looks like this:

```
<?xml version="1.0" encoding="UTF-8"?>
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

- simplexml_load_file() function to read XML data from file:

```
<?php
$xml=simplexml_load_file("note.xml") or die("Error: Cannot create
object");
print_r($xml);
?>
```

Get Node Values

```
<?php
$xml=simplexml_load_file("note.xml") or die("Error: Cannot create
object");
echo $xml->to . "<br>";
echo $xml->from . "<br>";
echo $xml->heading . "<br>";
echo $xml->body;
?>
```

Exercise

1. Get the books.xml from the link and print the entire book loop through.
https://www.w3schools.com/php/books.xml

Write in xml File

- The following code above is great and generates XML.
- The file_put_contents() function create a separate XML file called **books.xml**.

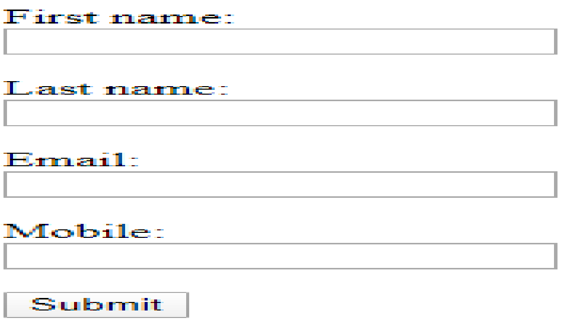
```
<?php
$simplexml= new SimpleXMLElement('<?xml
version="1.0"?><books/>');
$book1= $simplexml->addChild('book');
$book1->addChild("Booktitle", "The Wandering Oz");
$book1->addChild("PublicationDate", 2007);
$book2= $simplexml->addChild('book');
$book2->addChild("Booktitle", "The Roaming Fox");
$book2->addChild("PublicationDate", 2009);
file_put_contents('books.xml', $simplexml->asXML());
?>
```

Practice problem

Step 1: JSON objects are in name/value pairs separated by commas and hold objects by curly braces. Square brackets hold arrays. Save the following json data in mydata.json file.

```
[
    {
        "firstName": "Richard",
        "lastName": "Philip",
        "email": "richard@aiub.edu",
        "mobile": "12121"
    },
    {
        "firstName": "Aaron",
        "lastName": "Philip",
        "email": "aaron@gmail.com",
        "mobile": "44444"
    }
]
```

Step 2: Design the following HTML form to save data in json file.



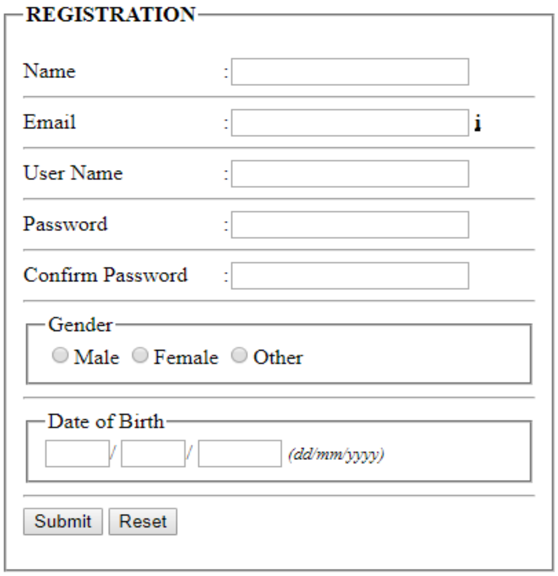
Step 3: Run the following PHP script to save Form data to json file.

```
$myFile = "mydata.json";
$arr_data = array(); // create empty array
//Get form data
$formdata = array(
               'firstName'=> $_POST['firstName'],
               'lastName'=> $_POST['lastName'],
               'email'=>$_POST['email'],
               'mobile'=> $_POST['mobile']
            );
//Get data from existing json file
$jsondata = file_get_contents($myFile);
// converts json data into array
$arr_data = json_decode($jsondata, true);
// Push user data to array
array_push($arr_data,$formdata);
//Convert updated array to JSON
$jsondata = json_encode($arr_data, JSON_PRETTY_PRINT);
//write json data into mydata.json file
file_put_contents($myFile, $jsondata)
```

Exercise

1. Design the following registration form and perform the following operations.

- Perform validation. [from previous week]
- Assign the data into the class objects.
- Write data into the text file, xml file and json file.



PHP Include Files

- The **include (or require)** statement takes all the text/ code/markup that exists in the specified file and copies it into the file that uses the include statement.
    - Including files is very useful when you want to include the same PHP, HTML, or text on multiple pages of a website.
    - It is possible to insert the content of one PHP file into another PHP file (before the server executes it), with the include or require statement.
- The **include and require** statements are identical, except upon failure:
    - require will produce a fatal error (E_COMPILE_ERROR) and stop the script.
    - include will only produce a warning (E_WARNING) and the script will continue

Example

Assume we have a standard footer file called "footer.php", that looks like this:

```
<?php
echo "<p>Copyright &copy; 1999-" . date("Y") . " W3Schools.com</p>";
?>
```

To include the footer file in a page, use the include statement:

```
<html>
<body>
<h1>Welcome to my home page!</h1>
<p>Some text.</p>
<p>Some more text.</p>
<?php include 'footer.php';?>
</body>
</html>
```

Exercise

1. Design a Registration form and perform the operations.

- Assign the data into the class objects. But class should be in separate PHP file/ directory. To do that use **include (or require)** function**.**

Upload File in PHP

- With PHP, it is easy to upload files to the server. First, ensure that PHP is configured to allow file uploads. In your "php.ini" file, search for the file_uploads directive, and set it to On. Remove comments from the line. `file_uploads = On`

- Create an HTML form that allow users to choose the any file they want to upload.
- The HTTP method should be POST and add enctype="multipart/form-data" attributes in the form tag.

Example

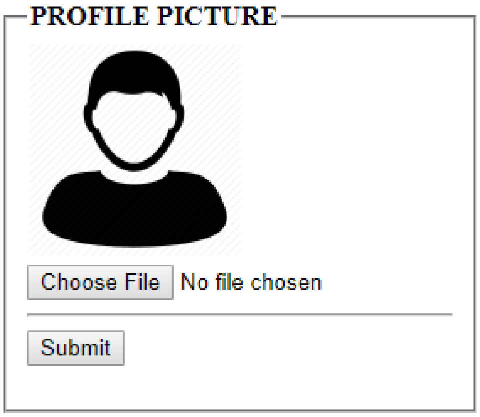
```
<!DOCTYPE html>
<html>
<body>
<form action="upload.php" method="post" enctype="multipart/form-data">
 Select image to upload:
 <input type="file" name="fileToUpload" id="fileToUpload">
 <input type="submit" value="Upload Image" name="submit">
</form>
</body>
</html>
```

The complete "upload.php" file now looks like this:

```
<?php
$target_dir = "uploads/";
$target_file = $target_dir .
basename($_FILES["fileToUpload"]["name"]);
//If you want to check file size, file already exists or allows
certain file formats, and you can do that.
if (move_uploaded_file($_FILES["fileToUpload"]["tmp_name"],
$target_file)) {
    echo "The file ".
basename( $_FILES["fileToUpload"]["name"]). " has been uploaded.";
  } else {
    echo "Sorry, there was an error uploading your file.";
  }
?>
```

Exercise

1. Design the following HTML form and perform following validations



Validation Rules

- Picture format must be in jpeg or jpg or png
- Picture size should not be more than 4MB

2. Design a form to upload CV.

Validation Rules

- File format must be in pdf or doc or docx
- File size should not be more than 4MB

References

[1] http://www.learningaboutelectronics.com/Articles/How-to-create-an-XML-document-with-PHP.php

[2] https://www.guru99.com/php-and-xml.html

[3] https://alexwebdevelop.com/xml-from-scratch/