

# Data Access using PHP

Course Code: **CSC 3215**

Course Title: WEB TECHNOLOGIES



**Dept. of Computer Science**  
**Faculty of Science and Technology**

<b>Lecturer No:</b>	<b>4</b>	<b>Week No:</b>	<b>4</b>	<b>Semester:</b>	
<b>Lecturer:</b>	<i>Supta Richard Philip &amp; richard@aiub.edu</i>				

# Data Access using PHP

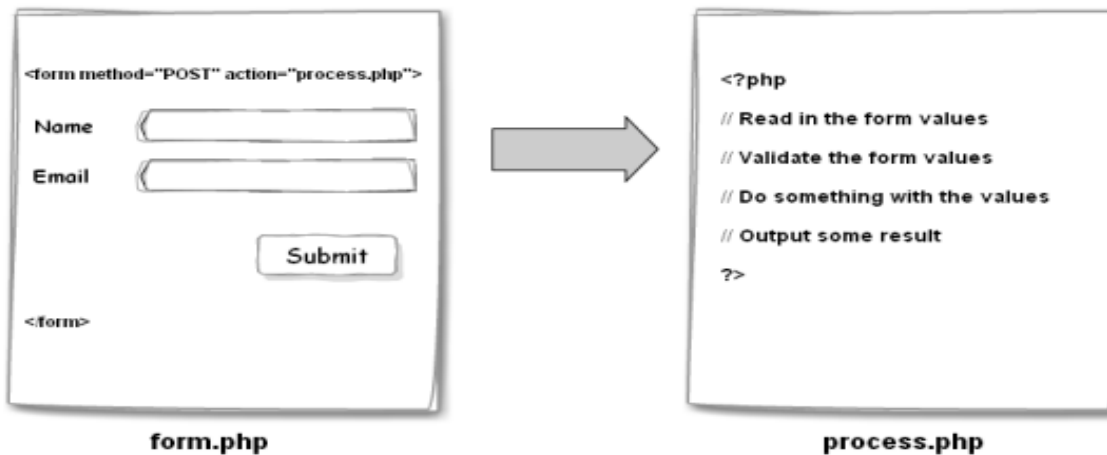


- 1. Learning outcome**
- 2. Data Access using PHP**
  - I. Data Read / Write in text file**
  - II. PHP SimpleXML - Read /write in xml File**
- 3. PHP Include Files**
- 4. Upload File in PHP**
- 5. Books and References**

# Learning outcome



- In this lecture, we will learn “read the form data and write to the files”.
- In between, we will assign data to the class variable(object) or Array.



# Data Access using PHP

Data Read / Write in 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

# Data Access using PHP

Data Read / Write in text file



- 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);`  
`?>`
- “r” means 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



## Data Read / Write in text file

- 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);`  
`?>`
- “r” means 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



## PHP Write to File - fwrite()

- 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
```

```
$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);
```

```
?>
```

# Data Access using PHP

PHP SimpleXML - Read /write in 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);
```

```
?>
```





# Creating a XML Document

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

le>  
tle>

```
<?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());
Echo $xml[0]->book[0]->Booktitle;
?>
```

# 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.



## Include 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>
```

# Upload File in PHP



- Before upload file in the server ensure that PHP is configured to allow file uploads and configure the "php.ini"
- In "php.ini" file, search for the file\_uploads, if the file\_uploads is directive, and set it to On: file\_uploads = On
- Create The HTML Form as following and the form needs the following attribute: enctype="multipart/form-data".

```
<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>
```



## upload.php

- 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 Allow
certain file formats, you can do that. We will see in the Lab
class.
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.";
}
?>
```



# Books

- W3Schools Online Web Tutorials; URL: <http://www.w3schools.com>
- PHP Documentation; URL: <http://www.php.net/docs.php>
- Sams Teach Yourself Ajax JavaScript and PHP All in One; Phil Ballard and Michael Moncur; Sams Publishing; 2010
- JavaScript Phrasebook; Christian Wenz; Sams Publishing; 2007
- PHP and MySQL Web Development, 4/E; Luke Welling and Laura Thomson; Addison-Wesley Professional; 2009
- JavaScript for Programmers Paul J. Deitel and Harvey M. Deitel; Prentice Hall; 2009
- Beginning PHP5, Apache, and MySQL Web Development; Elizabeth Naramore, Jason Gerner, Yann Le Scouarnec, Jeremy Stolz and Michael K. Glass; Wiley Publishing; 2005
- XML in a Nutshell, 3/E; Elliotte Rusty Harold and W. Scott Means; O'Reilly Media; 2004



# References

1. <http://www.kodecrash.com/javascript/read-write-json-file-using-php/>
2. <https://gist.github.com/recked/7329961>
3. <https://www.kodingmadesimple.com/2016/05/how-to-write-json-to-file-in-php.html>
4. <https://code.tutsplus.com/tutorials/how-to-upload-a-file-in-php-with-example--cms-31763>