



Lab worksheet 1 - Getting started with PHP

Introduction to writing and running PHP scripts

This Lab's Objectives

1. Copy your P3T files, if necessary
 2. Create a PhpStorm project
 3. Write, debug and execute a simple PHP script
-

Expected Outcomes

After the laboratory session, students should be able to:

- a) Create a PhpStorm project.
 - b) Write and store a simple PHP script.
 - c) Execute and debug PHP scripts.
-

Workplan

I recommend that you buy a new hardback A4 laboratory notebook for PHP. Everything you do in the lab should be noted down for later reference. Alternatively, use an application such as CherryTree Notes to make comprehensive record of everything that you do.

All your PHP lab work will be stored on your H drive (DMU) account, in your directory:

H:\p3t\phpappfolder\public_php

NB remember to use the lab work wiki available to you on BB to share and communicate with your colleagues. The more time you spend in the labs, the better you will write PHP.

1. If necessary, turn off your PC and reboot into Windows.
2. Create a directory for the topic: *H:\ctec2712\labs*. Use this for storing the lab sheets (your PHP scripts are stored elsewhere).
3. If you have already copied your P3T files, skip the next section of instructions.

Preparing P3T

NB you only need to complete this section if you have not yet created the P3T directory hierarchy on your H drive.

4. Download the extra lab sheet *ctec2712_wad_lab-worksheet-00_setup-p3t.pdf* from the module's Blackboard shell (under *ctec2712_wad_lab-worksheet-00_setup-p3t*). Follow the instructions carefully in this sheet to copy your P3T development environment.



Lab worksheet 1 - Getting started with PHP

NB your initial PHP scripts will be stored in `H:\p3t\phpappfolder\public_php\`. When the web application architecture becomes more sophisticated, you will also use `H:\p3t\phpappfolder\includes\` for the secure part of your web applications.

Using P3T

5. You should have a folder entitled *P3T-controls* on your desktop. If not, go back to *ctec2712_wad_lab-worksheet-00_setup-p3t.pdf* and follow the appropriate steps.
6. Open a browser ready to execute PHP scripts.

To use P3T, at the start of each session in a lab, remember to run the *Install* script in the P3T folder.

The *install* script maps your current desktop to your P3T file system on the H drive.

7. Use the *apache startup* script to start the P3T Apache server.

When the install process has finished, run the Apache startup script, and check that all is well by opening a browser and pointing it at `http://localhost:6789/`

Configuring PhpStorm

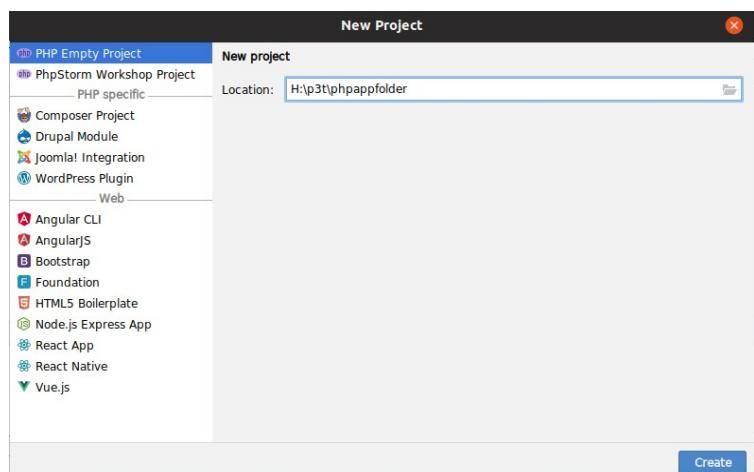
You are going to create your first PHP script using PhpStorm. PhpStorm is a dedicated PHP Integrated Development Environment (IDE) from JetBrains (<https://www.jetbrains.com/>).

PhpStorm is available in the labs via the *CloudPlayer* tool. It natively edits PHP, HTML, CSS, JS, etc. It features full debugging tools such as *step*, *watch* and *breakpoints* and integrates with Xdebug, the PHP debugging library (<https://xdebug.org/>) for local and remote debugging.

You may download and install a copy of PhpStorm, to use at home, from <http://www.jetbrains.com/phpstorm>. A full student licence is free whilst you have a DMU email account.

NB If you really prefer, you could use MS Notepad or similar to write code, although the level of PHP-specific and debugging support is somewhat limited. On Linux, *nano*, *gedit* or *gPHPEdit* offer more support than *vi* or *vim*.

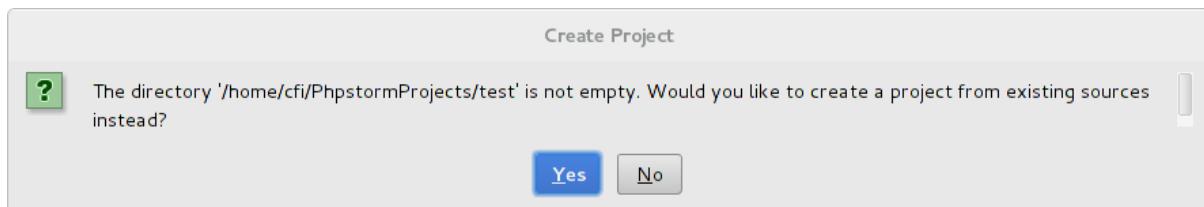
8. In PhpStorm, create a new project
 - o File => New Project
 - o Select `H:\p3t\phpappfolder` as the path





Lab worksheet 1 - Getting started with PHP

9. Select Yes in the next popup window:



Your new project is now created and opened ready for use.

PhpStorm Settings

PhpStorm has many configuration settings that allow you to fine-tune its performance. Two useful configuration settings are *line numbers* and *method separators*. These are available under *File => Settings*.

10. Select *File => Settings => Editor => General => Appearance*
11. Select *Show line numbers*
12. Select *Show method separators*
13. Select *Apply/OK*

PHP Configuration

PHP uses the configuration file *php.ini* for all configuration settings. You have two copies of this file. The working copy is in:

C:\user\php

and your personal copy is in

H:\p3t\php

Each time you run the P3T install script, the H: version is copied and overwrites the C: version. Therefore, if you wish, you can make changes to the C: version, and test them, but you will need to copy the C: version file back to H: if you want to keep the configuration changes between lab/editing sessions.

14. Open *C:\user\php\php.ini* for inspection.
15. Locate the sections for Date/Time zones, include_paths, error logging and display, and installed extension libraries.
16. In the browser, run the *phpinfo.php* script.
17. Compare the installed libraries (in *php.ini*) with the display in *phpinfo.php*. What information is displayed about each library?
 - **NB are there any security issues with the information displayed here?**



Your first PHP script

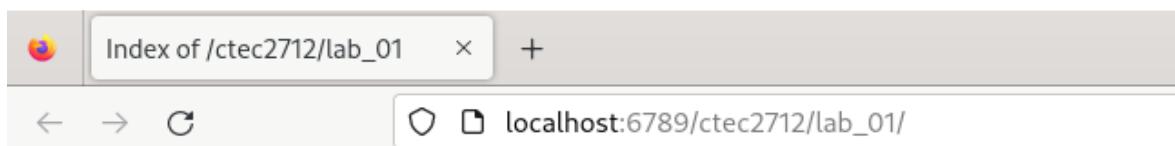
Create a new directory for each week's lab work. This is essential as some files/directories may have the same name each week.

18. Use PhpStorm to create a directory *lab_01* under *phpappfolder\public_php\ctec2712*
19. Again with PhpStorm, create a new PHP file called *hello-world.php* in the *lab_01* directory (how?)
20. Enter the following (very traditional) PHP script into your newly created file:
 - can you copy and paste?

```
<?php
echo 'Hello World!';
```

21. PhpStorm automatically saves your text as you work. However, this can be a little slow over the DMU network.
22. Open a browser (unless one is already open), and enter this URL (assuming that you have followed all previous instructions exactly):

http://localhost:6789/ctec2712/lab_01



Index of /ctec2712/lab_01

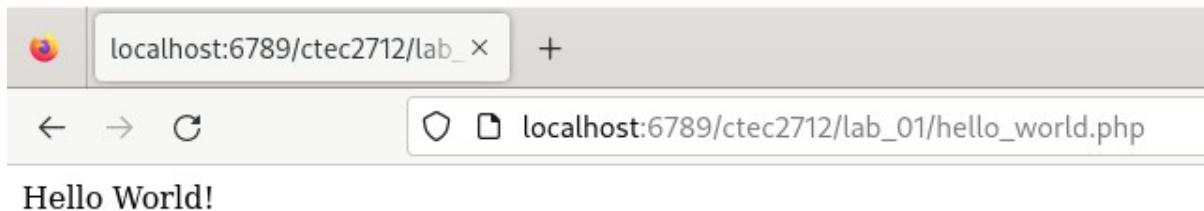
Name	Last modified	Size	Description
Parent Directory		-	
hello_world.php	2021-10-05 20:42	30	

Apache/2.4.57 (Debian) Server at localhost Port 6789



Lab worksheet 1 - Getting started with PHP

23. Click on the *hello-world.php* file name. The text *Hello World!* will appear in the browser. This is your first web application!



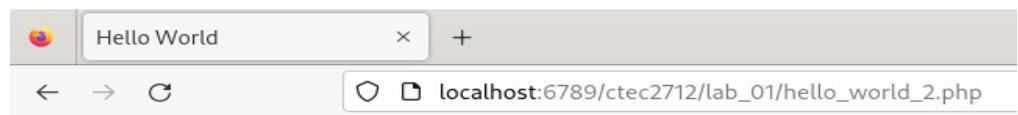
24. Returning to the PhpStorm window, in the same directory, create another new PHP script called **hello-world-2.php**
25. Enter the PHP script below, as before:
NB if you copy/paste the code, be aware that some characters may not copy correctly.

```
<?php

// Hello World the right way ...

$eol = "\n";
$output = '<!DOCTYPE html>' . $eol;
$output .= '<html lang="en">' . $eol;
$output .= '<head>' . $eol;
$output .= '<meta http-equiv="Content-Type" content="text/html;
charset=utf-8">' . $eol;
$output .= '<meta name="Author" content="Clinton Ingams">' . $eol;
$output .= '<title>Hello World</title>' . $eol;
$output .= '</head>' . $eol;
$output .= '<body>' . $eol;
$output .= '<h2>Hello World</h2>' . $eol;
$output .= '<p>Today\'s date is ' . gmdate("M d Y") . '</p>' . $eol;
$output .= '</body>' . $eol;
$output .= '</html>' . $eol;
echo $output;
```

26. The code will be explained in detail in a future lecture.
- 27.
28. Execute this file in your browser as before – you should see output something similar to:



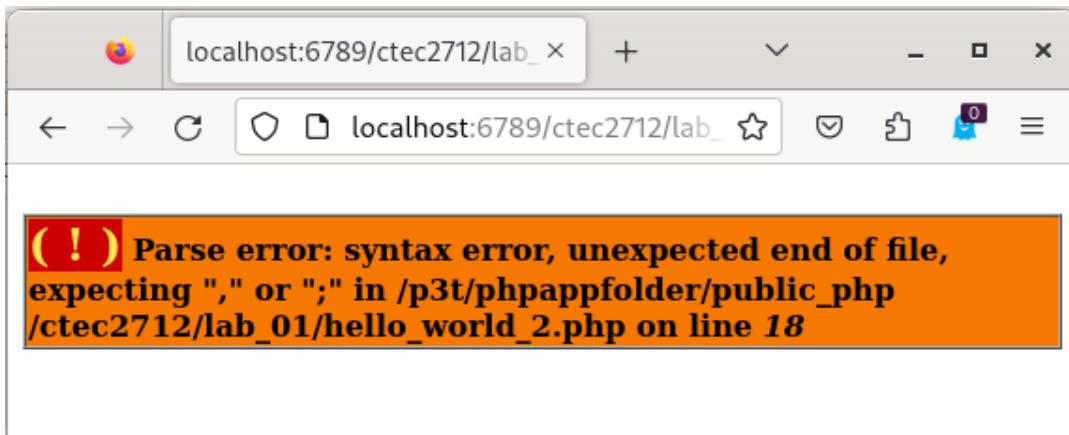
Hello World

Today's date is Mar 03 2024



Lab worksheet 1 - Getting started with PHP

If you don't see this result, you have an error in your code (such as the screenshot below). Locate it and correct it, then re-execute the script. This may become a multi-iterative process.



NB This error message is supplied by a debugging library called [xdebug](#).

29. View the HTML source code for the web page (View Page Source – varies with the browser you are using, or CTRL+U).

There is only really one technique to learn how to program:

- Write lots and lots and lots of code
- Make lots of mistakes in your code, and then fix the errors
 - actually, you'll find that writing mistakes comes naturally ...
- then write lots more code

30. Make some changes (including errors) in the above script and re-execute your program. Note the help that PhpStorm and xdebug (not so much yet) give you. NB check the PHP error log (where is it???) for a record of your mistakes.
31. Find the manual page for the PHP function *gdate()* on <http://php.net/>. Change the formatting to show different information. Repeat this for different formats.
32. What is the intention/effect of *\$eol* in the above code? How can you find out?
 - NB try replacing “\n” with *PHP_EOL*.



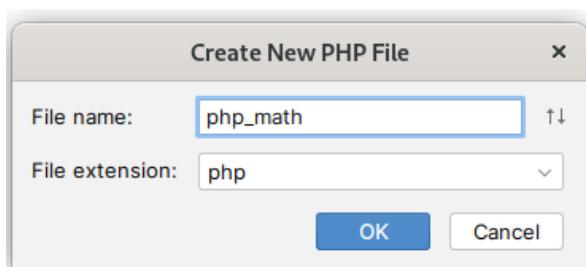
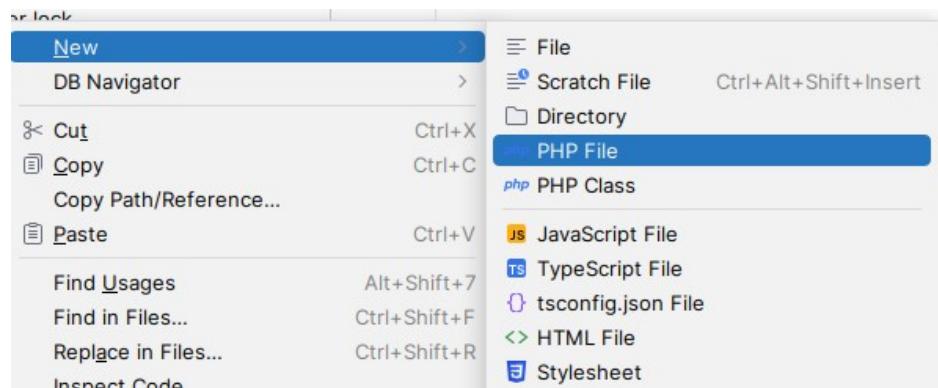
Lab worksheet 1 - Getting started with PHP

PHP Pandas

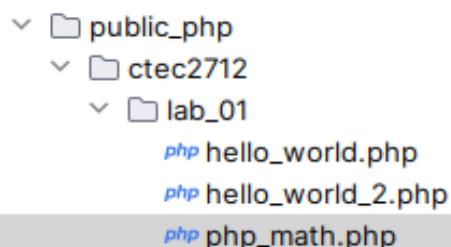
There are many beginners books/tutorials/web sites/videos, etc that will attempt to teach you PHP. Unfortunately, many of these are very out of date, and in some cases contain code examples that will introduce vulnerabilities into your web applications. Be careful to only use resources that reference PHP 8. A good indicator is if the example code uses the closing PHP tag: ?>. This is redundant, and if used can cause problems in your applications. Any examples using these should be avoided.

However, the PHP Pandas book is still one of the best introductions to the basic syntax and grammar of the PHP language. The PHP Pandas book is to be found [here](#).

33. Read chapters 3, 4 and 5 of the PHP Pandas [ebook](#) mentioned above. I suggest you make notes as you go,
34. To complete the exercises in chapter 5, first use PHPStorm to create a PHP script in your lab_01 directory.
35. To do this, right click on your lab_01 directory, then:



36. Your directory list in the PhpStorm directories pane should be:



NB if you don't follow the suggested directory structure, you may have problems in future labs.



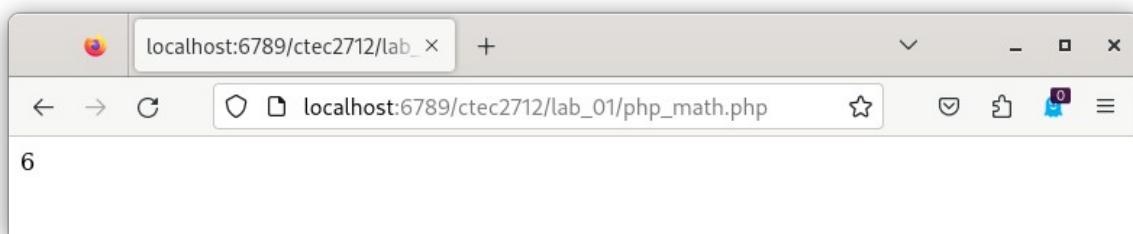
Lab worksheet 1 - Getting started with PHP

37. Work through the exercises in Chapter 5. Execute the file in your browser as before. You can put multiple exercises in the same file.

Executing the first very simple script:

```
php php_math.php ×  
1 <?php  
2  
3 echo 3 + 3;
```

will give this in the browser:



38. When you have completed chapter 5 exercises, experiment by creating many more mathematical examples using all of the operators mentioned

eg

- find the modulus after summing a series of numbers
- experiment with changing the order of processing with parentheses
- etc

Using Variables

39. Create a new PHP script (give it a semantically relevant name) in your lab_01 directory,

Remember that in PHP, identifier names (such as variables) are case sensitive, ie \$Var, \$vAr, \$VaR, \$var, \$VAR, \$Var, \$vAR are different variables and can store different values. This can become very confusing very quickly, so it is best to follow the coding standards.

Also, PHP variables are by default *weakly typed*. This means that if you assign a string value to a variable, the variable is now of data type *string*, and so on with different value types. More on this later.



Lab worksheet 1 - Getting started with PHP

40. Work through the exercises in Chapter 6 of the PHP Pandas ebook. Execute the script as before

Don't forget to experiment by making mistakes and correcting them as you go.

NB note particularly that variable names are camelCased. This is also mentioned in the [PSR-1 Basic Coding Standards document](#).

41. When you have finished chapter 6, enter, debug and execute the following code.

Are there any other restrictions on characters that can be used for variable names (how can you find out)?

```
<?php
$var = 'Bob';
$Var = 'Joe';
echo "$var, $Var";      // outputs "Bob, Joe"

$4site = 'not yet';    // invalid; starts with a number
$_4site = 'not yet';   // valid; starts with an underscore
$täyte = 'mansikka';  // valid; 'ä' is (Extended) ASCII 228
$this = 'no!';         // invalid: $this is a reserved variable
```

42. Create a new PHP script, then enter, debug and execute the following code. What effect does the '&' character have?

```
<?php
$foo = 'Bob';           // Assign the value 'Bob' to $foo
$bar = &$foo;            // Reference $foo via $bar.
$bar = "My name is $bar"; // Alter $bar...
echo $bar;
echo $foo;              // $foo is altered too.
```

Working with Strings

Assigning, displaying and concatenating strings (sequences of characters) are important topics, and can be confusing at times.

43. Enter, execute and debug the exercises in Chapter 7 (Strings) of the PHP Pandas ebook. Make sure that you fully understand the accompanying notes for each exercise.



Lab worksheet 1 - Getting started with PHP

44. Go back and look at the *hello-world-2.php* example above. You should now understand how the strings are being processed.

NB Dayle's book suggests that you use braces - {} - around values that you wish to be parsed within strings. This is not strictly necessary ...

45. Review all the work you have done in this lab and make many experiments!

ALL EXERCISES MUST BE COMPLETED BEFORE THE START OF LAB 2!

Exercises

46. Add a *docblock* page level comment to all of your scripts above.
- page level *docblock* comments give an explanation of the code for future developers. They are inserted at the top of the script (below the `<?php` opening tag), and are written like this:

```
<?php
/**
 * multi-line
 * comments here
 */
```

NB see [here](#) for more details about docblocks,

Learning and Using PHP at home

I have prepared a virtual machine for you configured with the same directory structure as P3T. Differences are:

- a) The user password to login to the VM is *password*
- b) The operating system is Linux (licensing restrictions prevent the distribution of a Windows VM).
- c) The AMP stack software is pre-installed and starts by default. The *P3T-controls* folder is not required.
- d) All scripts are stored within the VM, not on the H drive

To use the VM:

- e) Download the VM as a zip file from L:\CFI\ctec2712..., after which I suggest you copy it to a USB drive to transfer it to your laptop.
- f) You will need VMWare Player to open it for use – make sure the file is fully unzipped before attempting to use it.
- g) Your laptop should have at least 8Gb of RAM (memory), although 16Gb is better.