ENVIRONMENT SETUP FOR CSIS 3280

Please adhere to the following hardware recommendation and instruction to setup the software environment required for your CSIS 3280 course. Note that Douglas College assumes no responsibility for any data loss or damage to your PC, software or files on your computer as a result of setting up the following software. CSIS 3280 is a third level course and some basic computer literacy is required.

HARDWARE

Since physical access to Douglas College's computing labs may not be available due to the spread of Covid-19 virus, **students registering for the Summer term need to ensure access to a computer and internet to be able to meet the requirements for instructional delivery*.** The following is the hardware requirements that each student needs to prepare in order to be able to successfully follow and complete the course loads and tasks:

- Windows operating system. See note below for those using Mac machine**
- Decent processor speed (computer with 6th generation or newer of Intel i5 or i7 core, or AMD Ryzen)
- RAM at least 8GB (16GB is better)
- SSD drive (to have significant speed boost as compared to the ones using HDD ones)
- Webcam
- Good headset or speaker and microphone (if it is not embedded with the webcam)
- **Good internet connection (ethernet or stable wifi)**
- **Optional second monitor*****

Note:

* Although phone and tablet can be used to join the online classroom platform that we are going to use, i.e., ZOOM, you will get a much better learning experience using a dedicated computer.

** All the software demo, the labs and the tests for the course are designed in a Windows-based operating system. If you have a MacOS machine, the instructor will not be able to provide any guidance for you in setting up the environment and on how to do the labs, assignments and tests.

*** We will have a lot of practical hands-on in the class. By connecting a second monitor to your laptop of desktop, you will be able to follow the video feed from the lecture in one monitor while follow along to code in your other monitor. Note: You can also use an ultrawide monitor that allows you to split the monitor screen into several parts of display.

SOFTWARE

We will be using ZOOM to conduct the class. We might also use Blackboard Collaborate Ultra for some of our class activities. You will need to use a browser to access the platform. **Mozilla Firefox** and **Chrome** are reported to work flawlessly with the Collaborate.

Please have **Zoom** ready for the exams and office hour sessions. More information about this will be announced in the first lecture.

The following guide will help you setup the software required to complete the tasks in the course. If you are worry about any possible data loss, please backup your data before proceeding with the installation.

1. WAMP server (http://www.wampserver.com/en/)
2. Visual Studio Code and its PHP extension
3. Xdebug extension of VS Code

WAMP SERVER

Installing the dependencies

Depending on your machine, you may install either the 32-bit or the 64-bit versions of the WAMP server. Before you start the installation of the WAMP server,

1. **Please uninstall any previous version of WAMP server**, in case you have it on your system, then
2. **Make sure that you have installed all the required dependencies**. Please download the following http://wampserver.aviatechno.net/files/vcpackages/all_vc_redist_x86.zip

Run through EVERY executable. If you are prompted for a 'repair' of a specific dependency, you may choose to cancel the installer.

Download the following file http://wampserver.aviatechno.net/files/tools/check_vcredist.exe and run the executable. If it gives you a list of additional dependencies, you need to make sure to download each dependency and install it. Note: if it gives you a list of some dependencies, most probably because your PC is a 64-bit machine. Please download the following file http://wampserver.aviatechno.net/files/vcpackages/all_vc_redist_x86_x64.zip and install the missing dependencies.

You may want to make sure that you have all the dependency by running the check_vcredist.exe file again.

Installing WAMP Server

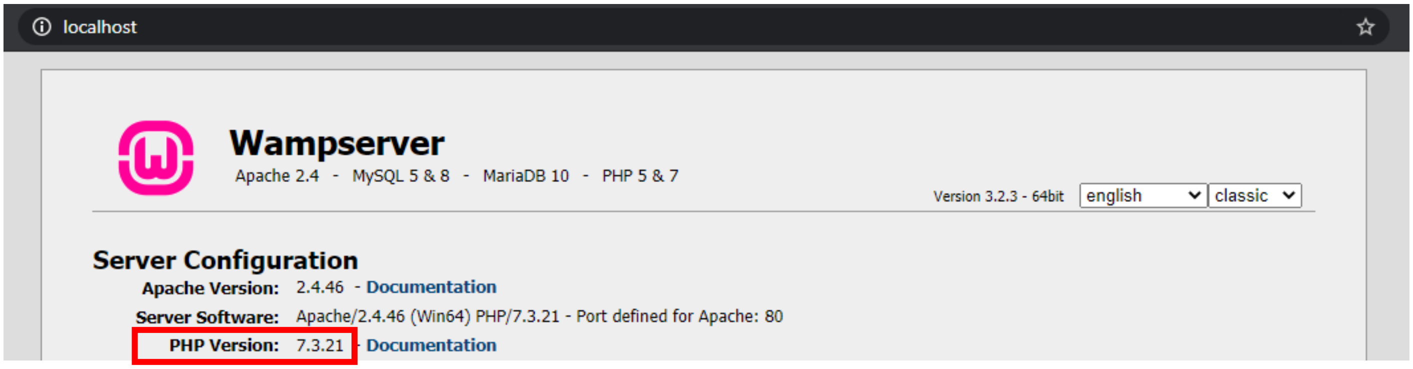
Download the correct version of WAMP server from http://www.wampserver.com/en/. If you have an IIS (Windows' web server) running, please make sure to shut it down before installing the WAMP server. In installing the WAMP server, please make sure that **MySQL 5.7** option is selected as well. Make sure that both PHP 5.X.X and PHP 7.3.X are also selected.

Note: Unless you want to use a different browser (Chrome) or text editor such as Notepad++, you can use the default configurations when installing the WAMP server.

To test the installation, click the WampServer icon (available at the Start Menu and Desktop). An icon should appear in the system tray and depending on the speed of your computer, the icon should turn green within a few seconds. If the icon is yellow or red, this means you probably did not install the appropriate dependencies. Uninstall the WAMP server and repeat the installation process.

Adding the PATH statement

Before we add the PATH for PHP and MySQL, you need to make sure the PHP version that is used by the Apache web server. Make sure that the WAMP server is started (notice the green WAMP server icon on the taskbar tray). Open your browser and type localhost. You should see the WAMP server configuration main page as shown in the following figure. Notice the PHP version. Your WAMP server installation may use different PHP version.



Server Configuration

Apache Version: 2.4.46 - Documentation
Server Software: Apache/2.4.46 (Win64) PHP/7.3.21 - Port defined for Apache: 80
PHP Version: 7.3.21 - Documentation

Then, perform the following steps:

1. Right click on “This PC” and select “Properties”
2. Click the “Advance system settings” on the left pane to bring a system properties pop-up window
3. Make sure that the “Advanced” tab is selected and click on “Environment Variables” button
4. Under the “User variables …”, select “Path” and click “Edit” button
5. Add the following paths by clicking "New" and entering the following paths. The following assume that you installed PHP 7.3.21 and MySQL 5.7.31. **Depending on the WAMP server being installed, please adjust the path accordingly**.

```
C:\wamp64\bin\php\php7.3.21
C:\wamp64\bin\mysql\mysql5.7.31\bin
```

6. Open a command prompt and test your path settings. Depending on the WAMP server being installed, you may see a different version of PHP 7 and MySQL 5.7

```
php -v
PHP 7.3.21 (cli) (built: Aug  4 2020 11:21:19) ( ZTS MSVC15 (Visual C++ 2017) x64 )
Copyright (c) 1997-2018 The PHP Group
Zend Engine v3.3.21, Copyright (c) 1998-2018 Zend Technologies

mysql -V
mysql  Ver 14.14 Distrib 5.7.31, for Win64 (x86_64)
```

VISUAL STUDIO CODE

1. Navigate to: https://code.visualstudio.com/Download
2. Click on 64-bit under "**System Installer**" under the Windows logo on the page.
3. Run the installer.
4. When prompted select all the additional tasks for VS Code including:
   - Add "Open with Code" action to Windows Explorer context Menu
   - Add "Open with Code" action to Windows Explorer directory context menu
   - Register Code as an editor for supported file types
   - **Add to PATH** (available after restart)
5. Continue the installer with the default values

INSTALLING THE PHP PACK

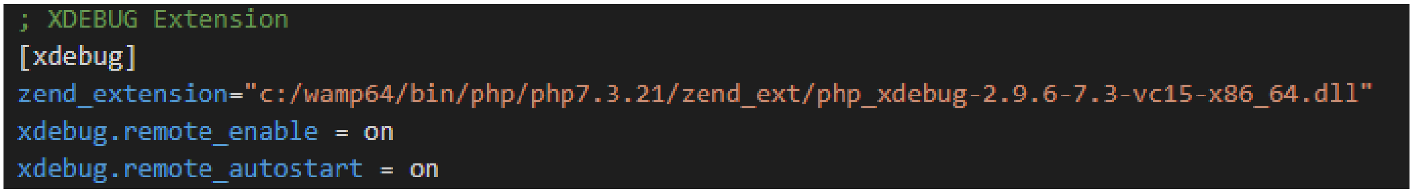
Execute Visual Studio Code, click on the "Extensions" button in the sidebar (Ctrl + Shift + X). Search for "PHP pack" and click the install next to the "**PHP Extension Pack**" you want the one by **Felix Becker.**

XDEBUG INSTALLATION

The new version of WAMP server comes with XDebug installed, we just have to modify two files to get them to work.

1. Navigate to the PHP folder path that was set when we install the WAMP server. Then open the phpForApache.ini file that contains the php setting for the Apache web server. If you are installing the same version of WAMP server of this document, the complete path of the file is **C:\wamp64\bin\php\php7.3.21\phpForApache.ini**

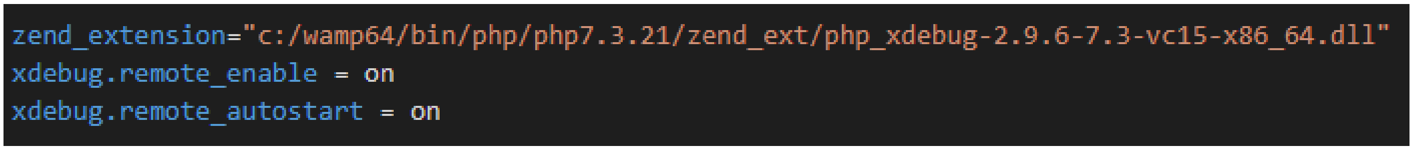
2. Locate the [xdebug] section (towards the bottom of the file). We need to turn on the remote debugging automatically. Ensure the top part of the [xdebug] section is as follow



```
; XDEBUG Extension
[xdebug]
zend_extension="c:/wamp64/bin/php/php7.3.21/zend_ext/php_xdebug-2.9.6-7.3-vc15-x86_64.dll"
xdebug.remote_enable = on
xdebug.remote_autostart = on
```

Note: make sure the path and filename are correct. **Depending on your WAMP installation, you need to adjust the path and filename accordingly**.

3. Open the php.ini file that is used by command line php. Navigate towards the bottom of the file and add the xdebug setting



```
zend_extension="c:/wamp64/bin/php/php7.3.21/zend_ext/php_xdebug-2.9.6-7.3-vc15-x86_64.dll"
xdebug.remote_enable = on
xdebug.remote_autostart = on
```

Note: make sure the path and filename are correct. **Depending on your WAMP installation, you need to adjust the path and filename accordingly**.

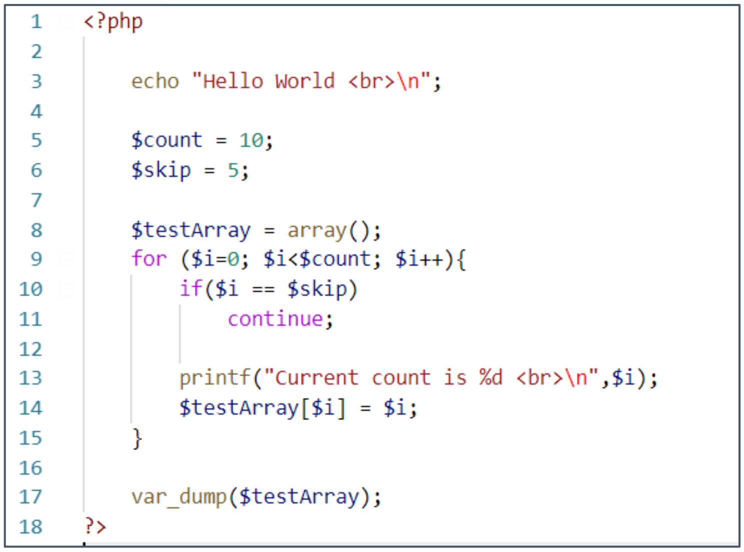
4. Restart the WAMP server
5. Use the command line to check php as follow. You should see a new line stating that XDebug support is enabled in php as shown below.

```
php -v
PHP 7.3.21 (cli) (built: Aug  4 2020 11:21:19) ( ZTS MSVC15 (Visual C++ 2017) x64 )
Copyright (c) 1997-2018 The PHP Group
Zend Engine v3.3.21, Copyright (c) 1998-2018 Zend Technologies
    with Xdebug v2.9.6, Copyright (c) 2002-2020, by Derick Rethans
```

ADDING THE XDEBUG SUPPORT IN VSCODE

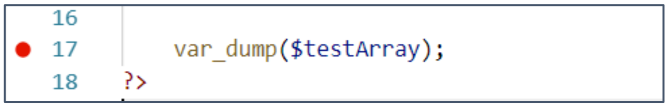
In order to add Xdebug support in VSCode, you need to open project/folder in VSCode. You need to open the **C:\wamp64\www** folder (if you installed the 64-bit version of the WAMP server)

1. Create a simple php file, named ***test.php*** (see below for an example php file)



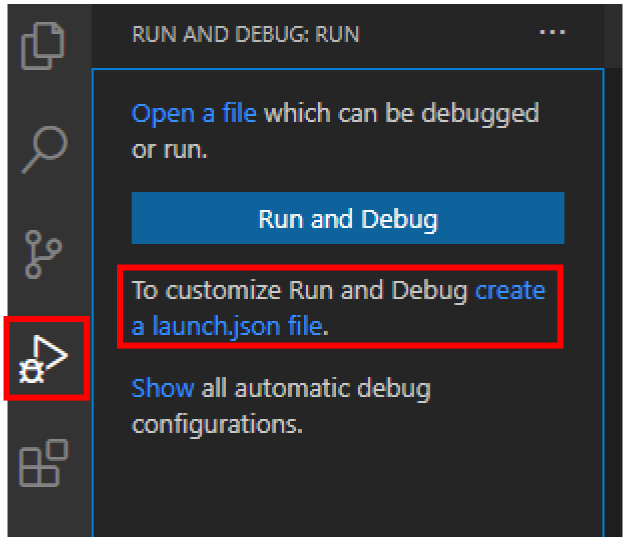
```php
<?php

    echo "Hello World <br>\n";

    $count = 10;
    $skip = 5;

    $testArray = array();
    for ($i=0; $i<$count; $i++){
        if($i == $skip)
            continue;

        printf("Current count is %d <br>\n",$i);
        $testArray[$i] = $i;
    }

    var_dump($testArray);
?>
```

2. Set a "Break Point" but clicking to the left of the line number you want to break at. A red dot should appear when you click there.



```php

●   var_dump($testArray);
?>
```

3. Click "Debug", then click “Create a launch.json file” link

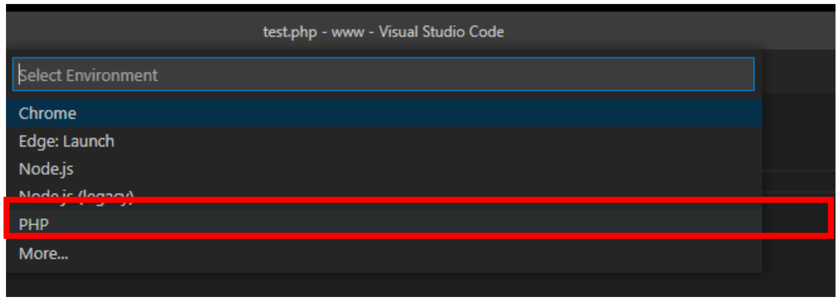


Open a file which can be debugged or run.

To customize Run and Debug create a launch.json file.

Show all automatic debug configurations.

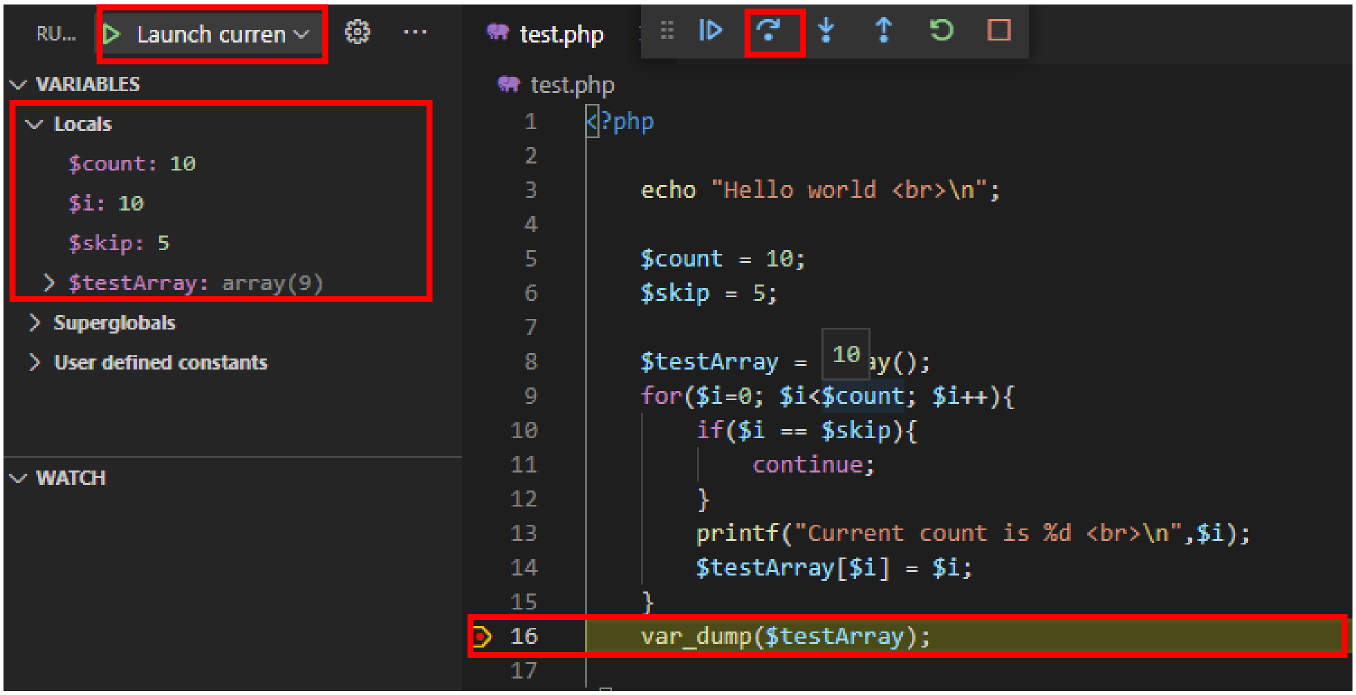
4. You will be prompted for a language. Select PHP.



5. Tab will be opened with file contents called "launch.json". You may then close the launch.json file.

DEBUGGING IN THE VSCODE

1. Change the debug configuration into “Launch currently open script” and press the play button. You will see the HUD.



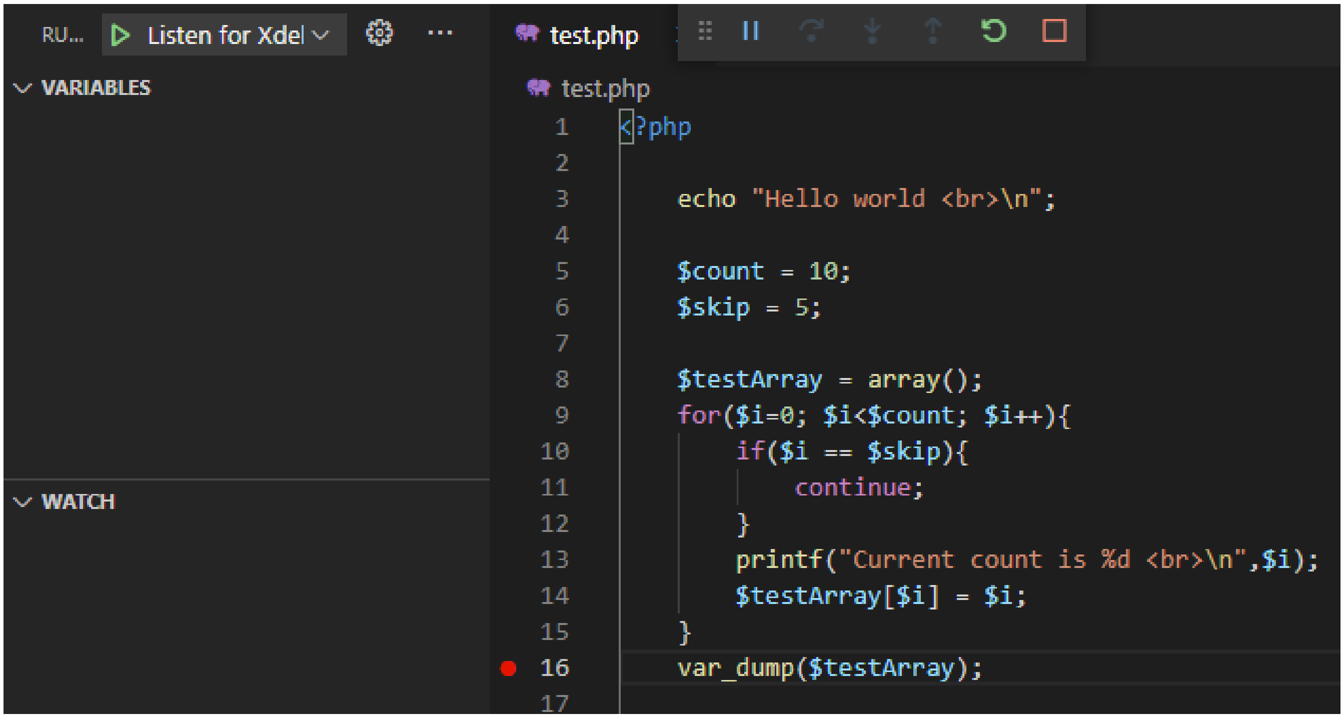
```
$count: 10
$i: 10
$skip: 5
$testArray: array(9)
```

```php
<?php

    echo "Hello world <br>\n";

    $count = 10;
    $skip = 5;

    $testArray = 10 ay();
    for($i=0; $i<$count; $i++){
        if($i == $skip){
            continue;
        }
        printf("Current count is %d <br>\n",$i);
        $testArray[$i] = $i;
    }
    var_dump($testArray);
```

2. Please take a loot at the highlighted part. In our test case, we put the breakpoint at var_dump function call. Notice the local variables observed.
3. Press the “Step over” or “Continue” button to exit.

DEBUGGING WITH THE WINDOWS COMMAND LINE

1. Change the debug configuration into “Listen to XDebug” and press the play button.

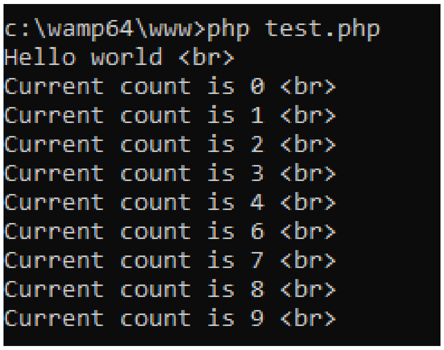


```php
<?php

    echo "Hello world <br>\n";

    $count = 10;
    $skip = 5;

    $testArray = array();
    for($i=0; $i<$count; $i++){
        if($i == $skip){
            continue;
        }
        printf("Current count is %d <br>\n",$i);
        $testArray[$i] = $i;
    }
    var_dump($testArray);
```

2. Open a windows command line, go to the folder of your php file and execute the test.php file

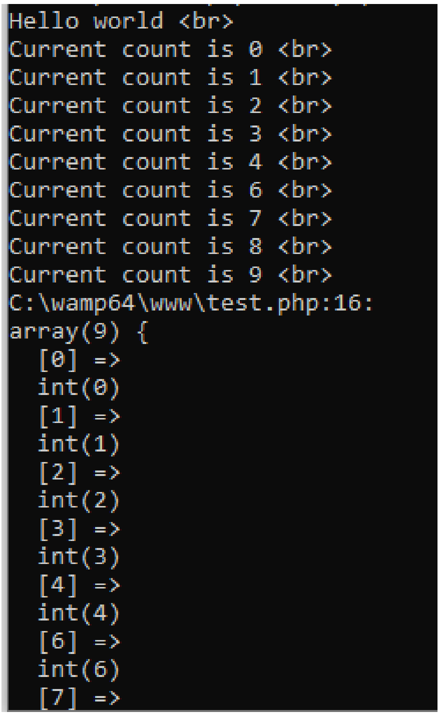
```
C:\Users\douglas>cd YOUR_FOLDER_PATH
C:\YOUR_FOLDER_PATH>php test.php
```

3. You will see that the output was printed, however the execution of the script was halted at the var_dump function call.



```
c:\wamp64\www>php test.php
Hello world <br>
Current count is 0 <br>
Current count is 1 <br>
Current count is 2 <br>
Current count is 3 <br>
Current count is 4 <br>
Current count is 6 <br>
Current count is 7 <br>
Current count is 8 <br>
Current count is 9 <br>
```

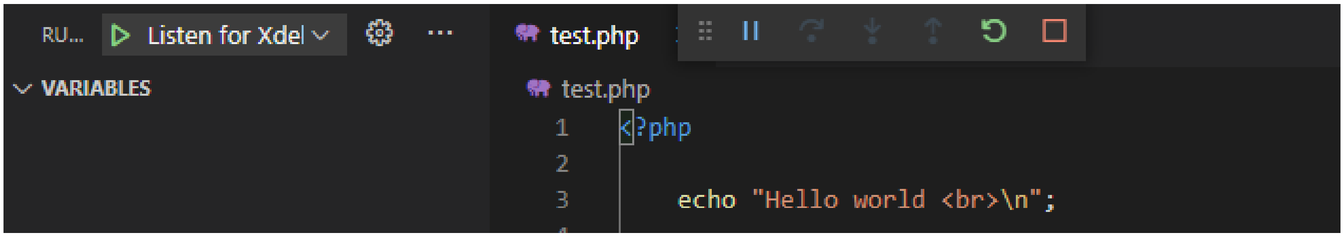
4. Go to the VSCode. You will see a similar debug condition as with the debugging from within the VSCode. Press the “Step over” or “Continue” button to continue the execution.
5. You will see in the command line windows the var_dump function call was executed.



```
Hello world <br>
Current count is 0 <br>
Current count is 1 <br>
Current count is 2 <br>
Current count is 3 <br>
Current count is 4 <br>
Current count is 6 <br>
Current count is 7 <br>
Current count is 8 <br>
Current count is 9 <br>
C:\wamp64\www\test.php:16:
array(9) {
  [0] =>
  int(0)
  [1] =>
  int(1)
  [2] =>
  int(2)
  [3] =>
  int(3)
  [4] =>
  int(4)
  [6] =>
  int(6)
  [7] =>
```

DEBUGGING WITH THE BROWSER

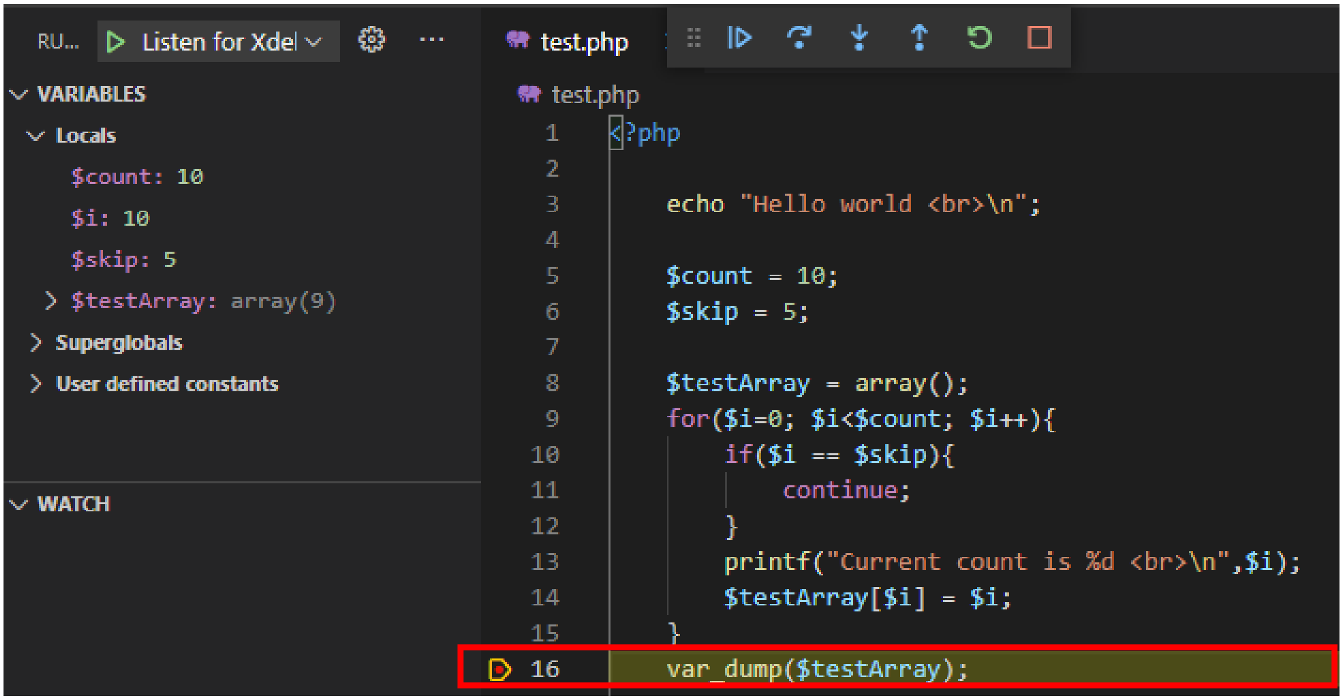
1. Change the debug configuration into “Listen to XDebug” and press the play button.



```
<?php

    echo "Hello world <br>\n";
```

2. Use a browser to open the php page, e.g., http://localhost/test.php. Navigate to the Web Page on the web server, you will notice that the page is in the "fetching state" for the browser (waiting for response from….)

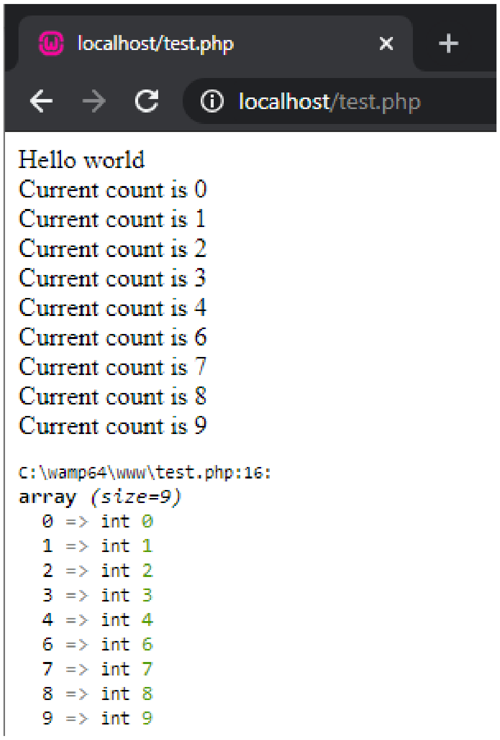
3. In our case the Breakpoint we set on var_dump has been caught so the server is waiting to hear back from XDebug which is waiting to hear back from VSCode by pressing the continue button. We can inspect the runtime of our php script while its executing inside the server.



```
$count: 10
$i: 10
$skip: 5
$testArray: array(9)
```

```php
<?php

    echo "Hello world <br>\n";

    $count = 10;
    $skip = 5;

    $testArray = array();
    for($i=0; $i<$count; $i++){
        if($i == $skip){
            continue;
        }
        printf("Current count is %d <br>\n",$i);
        $testArray[$i] = $i;
    }
    var_dump($testArray);
```

4. Go to the VSCode. You will see a similar debug condition as with the debugging from within the VSCode. Press the "Step over" or "Continue" button to continue the execution. You will see that the browser will now display the output.



```
Hello world
Current count is 0
Current count is 1
Current count is 2
Current count is 3
Current count is 4
Current count is 6
Current count is 7
Current count is 8
Current count is 9

C:\wamp64\www\test.php:16:
array (size=9)
  0 => int 0
  1 => int 1
  2 => int 2
  3 => int 3
  4 => int 4
  6 => int 6
  7 => int 7
  8 => int 8
  9 => int 9
```