# Lab I: Basic Usage of PHP

## PART 1: Configure and test your local web server

Pages 2-6 are instructions for installing a web server.
The actual assignment starts on page 7.

## Part1:

### Alternative A: DDEV

1. Download and install Docker Desktop (as a student, you are eligible to use “Personal” license)
2. Follow the installation guidelines for your OS at https://ddev.readthedocs.io/en/latest/users/install/ddev-installation/
   - Windows: “Install directly on traditional Windows” will be **easier** to use (but is not recommended by the authors of DDEV). The installer can be found here: https://github.com/ddev/ddev/releases, look for ‘ddev_windows_<architecture>_installer.v1.24.2.exe’
   - MacOS: the “Install script” method works just fine!
3. Pick a folder for storing your web projects

   Suggestions:
   - Windows: c:\web\
   - Mac: /Users/YourUserName/web/
   - Linux: /home/YourUserName/web/
4. Run your command line (Windows CMD, Windows Powershell, Mac Terminal, Gnome Terminal…) and change into the web project root, then make a folder for Lab1

   ```
   cd c:\web\
   mkdir lab1
   ```
5. Change into the newly created directory and initialize a ddev project in it

   ```
   cd lab1
   ddev config
   ```

   In the config wizard, leave the defaults for “Project name” and “Document Location”. Choose ‘php’ as Project Type.

   Don’t close the window, you will need it in step 8!

6. Make sure you have Visual Studio Code installed (https://code.visualstudio.com/download).

   Add the following extensions to VS Code:

   DDEV Manager<br>
   PHP Intelephense
7. In Visual Studio Code, open the folder “Lab1” you created in step 4. Create a file ‘index.php’ in Lab folder. Set its contents to this:

   ```
   <?php echo "Hello from php"; ?>
   ```

8. In the command line, run the project:

   ```
     ddev start
   ```

   The command will take some time to execute.
   The last line of output will output the address where your page can be accessed, e.g.:

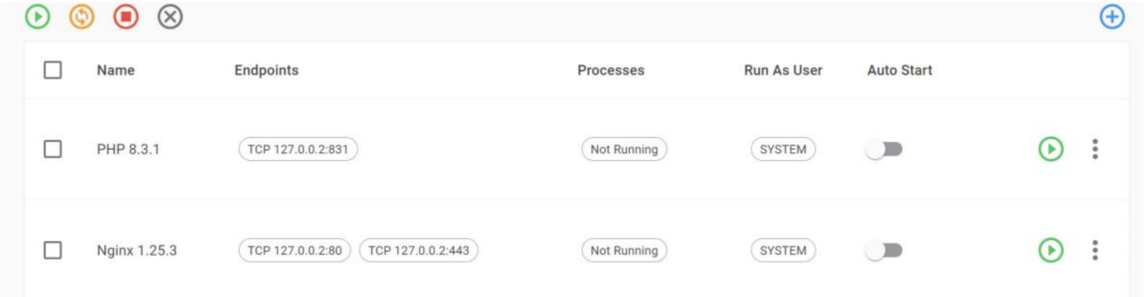
   ```
   Project can be reached at https://lab1.ddev.site
   https://127.0.0.1:51479
   ```

9. Using your web browser, visit the mentioned address and make sure you see the text “Hello from php” only.

## Part1:

### Alternative B: Wamp.Net  on Windows

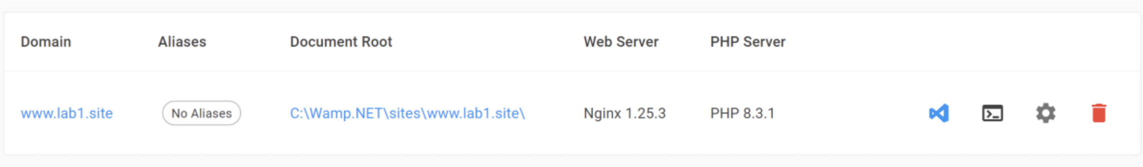
1. Download and install the latest version of Wamp.Net (https://wamp.net) Since the package is not signed, Microsoft Edge might complain that the file might be dangerous. Choose "Keep"
2. Launch Wamp.Net management panel
3. Using Servers -> Install Packages, add PHP 8.3.1 and Nginx 1.25.3. Leave the default parameter values of the said services, **except**, set PHP Time Zone to Europe/Riga
4. Start both services. The services page should look like this now:



| □ | Name | Endpoints | Processes | Run As User | Auto Start |
|---|---|---|---|---|---|
| □ | PHP 8.3.1 | TCP 127.0.0.2:831 | Not Running | SYSTEM | |
| □ | Nginx 1.25.3 | TCP 127.0.0.2:80<br>TCP 127.0.0.2:443 | Not Running | SYSTEM | |

5. Go to "Sites section" of the management panel.
6. Create a new site for Lab1 with the following properties:
   - a Domain name: lab1.dev
   - b Domain aliases www.lab1.site
   - c Web server: nginx 1.25.3
   - d PHP version: php 8.3.1

The sites collection should look like this



| Domain | Aliases | Document Root | Web Server | PHP Server |
|---|---|---|---|---|
| www.lab1.site | No Aliases | C:\Wamp.NET\sites\www.lab1.site\ | Nginx 1.25.3 | PHP 8.3.1 |

7. C:\Wamp.NET\sites\www.lab1.site is now your *web root*.
8. In Visual Studio Code, open your web root folder. Create a file 'index.php' in the root folder. Set its contents to this:

```
<?php echo "Hello from php"; ?>
```

9. Using your web browser, visit the address https://www.lab1.site and make sure you see the text "Hello from php" only.

## Part1:

### Alternative C: Ubuntu Linux 22.04 native or in Windows Subsystem for Linux

1) Install apache webserver, following the guidelines at https://www.digitalocean.com/community/tutorials/how-to-install-nginx-on-ubuntu-22-04

2) Follow guidelines at https://www.linuxtuto.com/how-to-install-php-8-3-on-ubuntu-22-04/ to install PHP 8.3.1, which will be useful in next lab assignments.

3) Your web root is /var/www/html by default

4) Create a sub-folder "lab1" in it and store all the files in this lab assignment in that folder.

5) In Visual Studio code, edit the file "index.php"

```
<?php
echo "Hello from PHP";
?>
```

6) Using your web browser, visit the address https://localhost/lab1/ and make sure you see the text "Hello from php" only.

## Part1:

### Alternative D: XAMPP

1) Download and install the latest version of XAMPP package https://www.apachefriends.org/

2) Launch XAMPP control panel.
3) Start Apache server, it must become green.

4) Create a folder for this lab in the web root C:\xampp\htdocs\lab1

5) In Visual Studio Code, open your web root folder. Create a file 'index.php' in the root folder. Set its contents to this:

```
<?php echo "Hello from php"; ?>
```

6) Using your web browser, visit the address http://localhost/lab1/ and make sure you see the text "Hello from php" only.

## Lab I: Basic Usage of PHP

### **PART 2:** PHP basics

1. Using Visual Studio Code, create a file “part2.php” in your lab1 folder (the same location where you just created “index.php”)

2. Within that file, do the following:

3. Change the content type of your script output to text/plain (using header function)

4. Using loop constructs, write a PHP script that takes an integer as input and outputs a pyramid of numbers based on a user-defined number of rows. Each row should display numbers incrementing from 1 up to the current row number. For example, if the input number of rows is 5, the output should be:

```
    1
   1 2
  1 2 3
 1 2 3 4
1 2 3 4 5
```

Ensure the pyramid is centered correctly by adding appropriate spaces before the numbers in each row.

## PART 3: PHP basics

1. Using Visual Studio Code, create a file "part3.php" un your lab1 folder (the same location where you just created "index.php")

   Write a PHP *function* that takes 3 parameters: loan amount, annual interest rate and loan term in years and calculates a monthly payment for a loan using a formula:

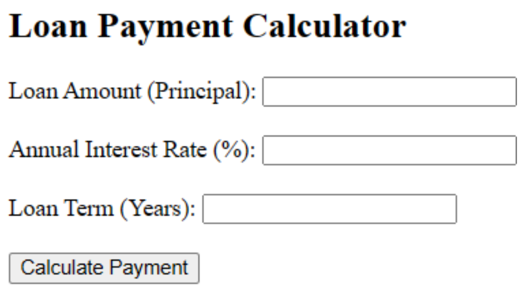
   $$Monthly\ Payment = Loan\ Amount \times \frac{i \times (1+i)^n}{(1+i)^n - 1}$$

   Where
   $i =$ monthly interest rate (annual interest rate divided by 12)
   $n =$ total number of payments (loan term multiplied by 12)

   In the main body of the script after the function definition, create some tests that run your function with several parameter values.

## **PART 4:** Forms

1. Create a new HTML file under the directory “lab1”, name the file “part4_input.html”
2. Create the following form in your HTML page. You can use POST or GET method.



**Loan Payment Calculator**

Loan Amount (Principal):

Annual Interest Rate (%):

Loan Term (Years):

3. When clicking the “Calculate Payment” button, all values should be submitted to a PHP file, say "part4_result.php".
4. Create the PHP file "part4_result.php". Within the script, read all input numbers and parse them as numbers.
5. If the data is wrong (values are not numbers, term is not integer, negative numbers or zero…) output an error message and stop execution of script.
6. Otherwise, calculate *monthly payment* of a loan using the function you created in Part 3.
7. Generate and print a monthly payment schedule. The schedule should include the following for each month:
    - *Month number*
    - *Remaining loan amount* which is equal to *loan amount* in the first month and is calculated as *remaining loan amount* in the previous month minus *principal payment* in the previous month
    - *Interest payment* calculated as *remaining loan amount* multiplied by *monthly interest rate*
    - *Principal payment* calculated as *monthly payment* minus *interest payment*
8. Test your program with different inputs.