

Лабораторна робота № 1

ЗНАЙОМСТВО З PHP

Мета роботи: отримання практичних навичок роботи з установки й роботі з одним з основних засобів розробки додатків на мові програмування PHP [1].

Досліджуваний матеріал: Мова програмування PHP є одним з найпопулярніших мов сценаріїв в світі. Для створення додатків використовують два найбільш популярних середовища розробки: Eclipse, StormPHP. Легковим середовищем, що не вимагає установки, є Eclipse PHP. Цю програму досить розпакувати і можна починати працювати.

1. Постановка задачі

Навчитися встановлювати і працювати з середовищем розробки додатків на мові сценаріїв PHP (Середовище на вибір студента).

2. Порядок виконання роботи

1. Прочитати теоретичний матеріал
2. Вивчити процес інсталювання EclipsePHP/PHPStorm або іншого середовища
3. Навчитися створювати проект в обраному середовищі
4. Навчитися запускати проект PHP
5. Навчитися користуватися коментарями.
6. Навчитися створювати прості програми мовою PHP
7. Зробити висновки по роботі.

3. Зміст звіту

1. Постановка завдання.
2. Встановити завдання, які вирішуються в лабораторній роботі
3. Описати особливості процесу створення і запуску додатка.
4. Коротко описати лексеми і висновок інформації на екран.
5. Висновки по роботі.

4. Теоритичні відомості

Для створення програм на мові програмування PHP нам буде потрібно текстовий редактор. Найбільш популярним на сьогоднішній день є програма Notepad ++.

Перейдемо до раніше створеного каталогу ~\localhost, який буде зберігати всі документи сайту. Створимо текстовий файл і назвемо його index.html. Відкриємо його в текстовому редакторі і додамо в нього наступний код:

```
<!DOCTYPE html>
<html>
<head>
<title>Перший сайт на PHP</title>
<meta charset="utf-8">
</head>
<body>
<h2>Введіть ваші дані:</h2>
<form action="display.php" method="POST">
<p>Введіть ім'я: <input type="text" name="firstname" /></p>
<p>Введіть прізвище: <input type="text" name="lastname" /></p>
<input type="submit" value="Відправити ">
</form>
</body>
</html>
```

Код html містить форму з двома текстовими полями. При натисканні на кнопку дані цієї форми відсилаються скрипту display.php, так як він вказаний в атрибуті action.

Тепер створимо цей скрипт, який буде обробляти дані. Додамо в папку ~\localhost новий текстовий файл. Перейменуємо його у display.php. За замовчуванням файли програм на php мають розширення .php. Отже, додамо в файл display.php наступний код:

```
<!DOCTYPE html>
<html>
<head>
<title>Перший сайт на PHP </title>
<meta charset="utf-8">
</head>
<body>
<?php
$name = $_POST["firstname"];
$surname = $_POST["lastname"];
echo "Ваше ім'я: <b>". $name . " " . $surname . "</b>";
?>
</body>
```

</html>

Тут вже в розмітці html йдуть вкраплення коду PHP. Для додавання виразів PHP на сторінку використовуються теги <? Php?>, Між якими йдуть інструкції на мові PHP. У коді php ми отримуємо дані форми і виводимо їх на сторінку.

Кожне окреме вираз PHP має завершуватися крапкою з комою. В даному випадку у нас три вирази. Два з них отримують передані дані форми, наприклад, \$ name = \$ _POST ["firstname"] ;.

\$ Name – це змінна, яка буде зберігати деяке значення. Всі змінні в PHP передують знаком \$. І так як форма на сторінці index.html використовує для відправки метод POST, то за допомогою виразу \$ _POST ["firstname"] ми можемо отримати значення, яке було введено в текстове поле з атрибутом name = "firstname". І це значення потрапляє в змінну \$ name.

За допомогою оператора echo можна вивести на сторінку будь-яке значення або текст, які йдуть після оператора. В даному випадку (echo "Ваше ім'я: ". \$ Name. "". \$ Surname. "</ B>") за допомогою знака точки текст в лапках з'єднується зі значеннями змінних \$ name і \$ surname і виводиться на сторінку .

Основи синтаксису

При створенні першої програми на PHP вже були порушені деякі основні принципи створення скриптів на мові PHP. Тепер, розглянемо їх більш детально.

Програма або скрипт на PHP, як правило, знаходиться в файлі розширенням .php. Хоча розробники можуть також вставляти код php і в файли з розширеннями .html / .htm.

Коли користувач звертається до скрипту в адресному рядку браузера, набираючи, наприклад, http: // localhost: 8080 / display.php, то веб-сервер передає його інтерпретатора PHP. Потім інтерпретатор обробляє код і генерує на його основі html-розмітку. І потім згенерований html-код відправляється користувачеві.

Документ PHP може містити як розмітку html, так і код на мові php. Для переходу від розмітки html до коду php використовуються теги <?php та ?>, Між якими йде код php. Дані теги служать зазначенням інтерпретатора, що їх вміст треба інтерпретувати як код php, а не розмітку html.

Також можна використовувати коротку версію тегів: <? і ?>. Для цього в файлі php.ini треба змінити значення short_open_tag = Off на short_open_tag = On.

Розглянемо найпростіший скрипт на php:

```
<html>
<head>
<title>Веб-сайт</title>
```

```

</head>
<body>
<?php
echo "<p>Привет мир!</p>";
echo "2 + 2 = " . (2+2);
?>
</body>
</html>

```

Після обробки файлу інтерпретатор сформує наступну розмітку:

```

<html>
<head>
<title>Веб-сайт</title>
</head>
<body>
<p>Привет мир!</p>
2 + 2 = 4
</body>
</html>

```

Тут використані дві інструкції echo "<p> Привіт світ! </ p>" і echo "2 + 2 =". (2 + 2), який виводять певне значення на сторінку. Кожна окрема інструкція в PHP завершується крапкою з комою.

Коментарі

При створенні веб-сайту ми можемо використовувати коментарі. Наприклад, ми можемо прокоментувати будь-яку дію, щоб згодом мати уявлення, що цей код робить:

```

<?php
echo "<p>Привет мир!</p>"; // вывод сообщения
// echo "2 + 2 = " . (2+2);
?>

```

Знак // передує однорядковий коментар, і все що йде після цього знака на одному рядку, буде вважатися коментарем і не буде виконуватися інтерпретатором. При обробці інтерпретатор просто буде пропускати коментарі.

Якщо нам треба закомментувати кілька рядків, то ми можемо використовувати багаторядковий коментар / * текст коментаря * /:

```

<? php
echo "<p> Привіт світ! </ p>"; // виведення повідомлення
/*
багаторядковий коментарвисновок результату арифметичного виразу
echo "2 + 2 =". (2 + 2);
* /
?>

```

Всі рядки всередині коментаря також не оброблятися інтерпретатором.

Змінні

Як і в багатьох мовах програмування, в PHP є змінні. Змінні зберігають окремі значення, які можна використовувати у виразах на PHP. Для позначення змінних використовується знак долара \$. Наприклад:

```
<?php
$a = 10;
echo $a;
?>
```

Тут визначена змінна, яка буде зберігати число 10. Присвоєння значення відбувається за допомогою знака рівності =.

Можна привласнювати значення іншій змінній:

```
$a = 10;
$b=$a;
echo $b;
```

PHP є чутливою до регістру мовою, а значить, змінні \$ counter і \$ Counter представлятимуть дві різні змінні.

Також при найменуванні змінних нам треба враховувати наступні правила:

- Імена змінних повинні починатися з алфавітного символу або з підкреслення
- Імена змінних можуть містити тільки символи: a-z, A-Z, 0-9, і знак підкреслення
- Імена змінних не повинні включати в себе прогалини

Перевірка існування змінної. оператор isset

Якщо змінна оголошена, але їй спочатку не присвоєно ніякого значення (інакше кажучи вона не ініціалізована), то нам буде проблематично її використовувати. наприклад:

```
<?php
$a;
echo $a;
?>
```

При спробі вивести значення змінної ми отримаємо діагностичне повідомлення про те, що змінна не визначена: Notice: Undefined variable: a in C: \localhost \ echo.php on line 3.

Оператор `isset ()` дозволяє визначити, ініціалізована змінна чи ні. Якщо змінна визначена, то `isset ()` повертає значення `true`. Якщо змінна не визначена, то `isset ()` повертає `false`. Наприклад:

```
<?php
$a;
if(isset($a))
    echo $a;
else
```

```
echo "переменная а не определена";
?>
```

Для перевірки змінної використовувалася конструкція `if ... else`, яка визначає істинність виразу. І якщо вираз істинний, тоді виконується вираз після блоку `if`. Якщо ж вираз помилково (тобто одно `false`), виконується вираз після блоку `else`.

Тут змінна не ініціалізована, тому оператор `isset ($ a)` буде повертати значення `false`, і, отже, буде спрацьовувати блок `else`. Якби ми присвоїли змінної деяке початкове значення, наприклад, `$ a = 20`, то оператор `isset` повернув би значення `true`, і браузер вивів би її значення.

За допомогою оператора `unset ()` ми можемо знищити змінну:

```
<?php
$a=20;
echo $a; // 20
unset ($a);
echo $a; // ошибка, переменная не определена
?>
```

Інтерполяція змінних в PHP

Значення змінної може бути відображено наприклад так:

```
<?php
$capital = "Paris";
```

```
echo "The capital of France is", $capital,"<br />";
?>
```

Але є спосіб зробити це простіше. Якщо ім'я змінної укладено в подвійні (НЕ одинарні) лапки, то змінна інтерполюється. наприклад:

```
<?php
$capital = "Paris";
```

```
echo "The capital of France is $capital <br />";
?>
```

Також існує одна тонкість при використанні інтерполяції змінних. наприклад:

```
<?php
$ Text = "news";
echo "Where's the $ textraper <br />";
?>
```

Очікується, що ім'я змінної `$ text` буде замінено на її значення. Але в результаті буде виведено повідомлення про помилку. Щоб цього не траплялося потрібно вказати яку частину слова `textraper` ми маємо ввести за ім'я змінної. Для цього потрібно взяти ту частину яка нам потрібна в фігурні дужки:

```
<?php
$text = "news";
echo "Where's the {$text}paper <br />";
?>
```

Змінні, що містять імена інших змінних

У PHP можливо розміщувати в значення змінних не тільки звичайні значення, а й імена інших змінних.

```
<?php
$apples = 5; // Создаем переменную $apples
$fruit = "apples"; /* Создаем переменную $fruit, которая содержит имя
переменной $apples */
// Сейчас мы можем вывести $apples, как $$fruit
echo "Число яблок - ", $$fruit;
?>
```

Для коректного відображення подібних змінних в строкових константах, взятих в подвійні лапки, слід також використовувати фігурні дужки: ``${$fruit}`. наприклад:

```
<?php
echo "Число яблок - `${$fruit}`";
?>
```

Типи даних

PHP є мовою з динамічною типізацією. Це означає, що тип даних змінної виводиться під час виконання, і на відміну від ряду інших мов програмування в PHP не треба вказувати перед змінної тип даних.

PHP підтримує вісім простих типу даних:

- boolean (логічний тип)
- integer (цілі числа)
- double (дробові числа)
- string (рядки)
- array (масиви)
- object (об'єкти)
- resource (ресурси)
- NULL

Integer (цілочисельний тип)

Являє ціле число зі знаком розміром в 32 біта (від -2 147 483 648 до 2 147 483 647).

```
$int = -100;
```

```
echo $int;
```

Тут змінна \$ int представляє цілочисельний тип, так як їй присвоюється цілочисельне значення.

Крім десяткових цілих чисел PHP має можливість використовувати також виконавчі, восьмеричні і шістнадцяткові числа. Шаблони чисел для інших систем:

- шістнадцятиричні: 0 [xX] [0-9a-fA-F]
- восьмеричні: 0 [0-7]
- виконавчі: 0b [01]

Наприклад:

```
<?php
// Все числа в десятичной системе имеют значение 28
$int_10 = 28; // десятичное число
$int_2 = 0b11100; // двоичное число
$int_8 = 034; // восьмеричное число
$int_16 = 0x1C; // шестнадцатеричное число
echo "int_10 = $int_10 <br>";
echo "int_2 = $int_2 <br>";
echo "int_8 = $int_8 <br>";
echo "int_16 = $int_16";
?>
```

Тип double (числа з плаваючою точкою)

Розмір числа з плаваючою точкою залежить від платформи. Максимально можливе значення, як правило, становить $\sim 1.8e308$ з точністю близько 14 десяткових цифр. Наприклад:

```
<?php
$a1 = 1.5;
$a2 = 1.3e4; // 1.3 * 10^4
$a3 = 6E-8; // 0.00000006
echo $a1 . " | " . $a2 . " | " . $a3;
?>
```

Тип boolean (логічний тип)

Змінні логічного типу можуть приймати два значення: true і false або інакше кажучи істина і брехня. Найчастіше логічні значення використовуються в умовних конструкціях:

```
<?php
$foo = true;
$a=10;
$b=5;
echo "foo = true <br>";
if($foo)
    echo $a+$b;
else
    echo $a-$b;
```



```
$foo = false;  
echo "<br> foo = false <br>";  
if($foo)  
    echo $a+$b;  
else  
    echo $a-$b;  
?>
```

Вираз `if ()` перевіряє істинність виразу. В даному випадку перевіряється значення змінної `$foo`. Або воно істинно або дорівнює `true`, то виконується наступне за оператором `if` вираз. А якщо змінна або вираз в операторі `if` одно `false`, то виконується вираз після оператора `else`.

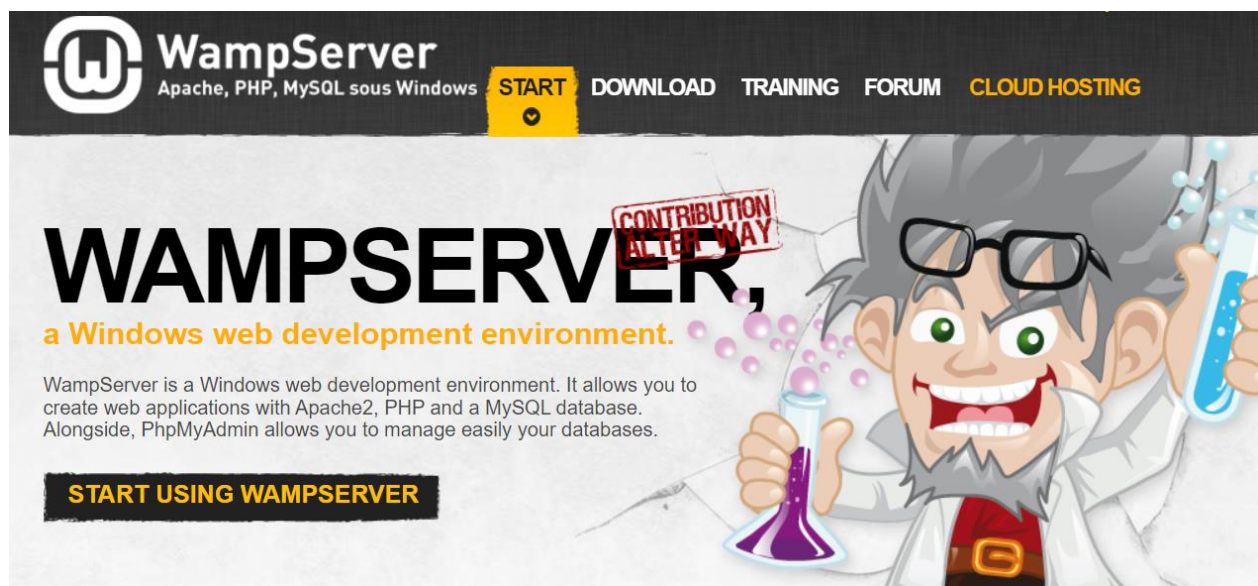
Завдання

Завдання 1. *Встановлення та налаштування програмного середовища для веб-розробки*

WAMP (Windows Apache MySQL PHP) — комплекс програмного забезпечення для Windows систем, який дає змогу швидко розгорнути сервер для розробки або надання веб-сервісів.

Для користувачів Linux існує аналогічне програмне забезпечення – LAMP. Тепер перейдемо власне до інсталяції.

Крок1. Переходимо на сайт www.wampserver.com/en/



Натискаємо в меню **Download**, побачимо кілька версій програми для завантаження. Вибираємо необхідну нам (в залежності від вашої Windows та версій програм, які включені в збірку)



DOWNLOADS

WampServer is available for free (under GPML license) in two distinct versions : 32 and 64 bits. Wampserver 2.5 is not compatible with Windows XP, neither with SP3, nor Windows Server 2003. Older WampServer versions are available on [SourceForge](#).

WAMPSERVER

WAMPSERVER 64 BITS (X64) 3.3.5

– WAMPSERVER 64 BITS (x64) 3.3.5 – Apache 2.4.59 – PHP 7.4.33/8.0.30/8.1.28/8.2.18/8.3.6 – MySQL 8.3.0 – MariaDB 11.3.2 – PhpMyAdmin 5.2.1 – Adminer 4.8.1 – PhpSysInfo 3.4.3

[changelog](#)

WAMPSERVER 32 BITS (X86) 3.3.0

– WAMPSERVER 32BITS (86) 3.3.0 – Wampserver 3.3.0 32 bit x86 – Apache 2.4.54.2 – PHP 7.4.33/8.0.26/8.1.13/8.2.0 – MySQL 5.7.40 – MariaDB 10.6.5 – PhpMyAdmin 5.2.0 – Adminer 4.8.1 – PhpSysInfo 3.4.2

[changelog](#)

Встановлюється WAMP, як і всі інші програми на Windows. Вас можуть попросити встановити відповідні пакети для роботи.

В кінці інсталяції програма спитає ваш браузер по-замовчуванню, для того, щоб синхронізуватись з ним та редактор в якому ви плануєте працювати. Вибираємо свій браузер і рухаємось далі. Після встановлення і запуску програми, перейшовши за посиланням <http://localhost/> ви маєте побачити відповідну сторінку, якщо все працює:

The screenshot shows the WampServer control panel. At the top, there is a logo and the text "Wampserver" followed by "Apache 2.4 - MySQL 5 & 8 - MariaDB 10 - PHP 5, 7 & 8 Version 3.3.2 - 64bit". There are dropdown menus for "english" and "classic". Below this is the "Server Configuration" section with details for Apache, PHP, MySQL, and MariaDB, including version numbers and links to documentation. At the bottom, there are three sections: "Tools" with links to phpinfo(), xdebug_info(), and PhpSysInfo 3.4.3; "Your Projects (0)" with a message about creating a new project; and "Your Aliases (2)" listing adminer 4.8.1 and PhpMyAdmin 5.2.1. There is also a "Your VirtualHost (1)" section listing localhost.

На стартовій сторінці серверу вказані підключені модулі, є лінки на **phpinfo()** та **phpMyAdmin**, а внизу подані **Your Projects**, це всі проекти, які наявні на сервері. До слова, всі проекти (а це ніщо інше, як папки з файлами) повинні бути розміщені в папці **wamp\www**

Завдання 2. · Створити текстовий документ з і'м'ям **index.php** · Створити першу програму, яка виводить такий текст:
Полину в мріях в купель океану, Відчую **шовковистість** глибини,
Чарівні мушлі з дна собі дістану,
Щоб **взимку** тішили
мене вони...

Завдання по варіантам:

Варіант 1. Завдання 1. (Сума грн. задається програмно, не вводиться з клавіатури) Написати програму, що переводить гривні в долари. В результаті повинно з'явитися наступне:

Наприклад:

1500 грн. можна обміняти на 51 долар

Варіант 2. Завдання 1 (Номер задається програмно, не вводиться з клавіатури) Визначення сезону за номером місяця: Напишіть програму, яка приймає номер місяця (1-12) та виводить відповідний сезон пори року. Використати конструкцію **if-else**

Варіант 3. Завдання 1 (Символ задається програмно, не вводиться з клавіатури) Напишіть програму, яка приймає символ (букву) і визначає, чи є він голосним чи приголосним. Використати конструкцію **switch**

Варіант 1 Завдання 2

(Число задається програмно, не вводиться з клавіатури)

Дано випадкове тризначне число. Знайдіть:

1. Суму його цифр.
2. Знайдіть число, отримане виписуванням в зворотному порядку цифр даного тризначного натурального числа.
3. Переставте цифри так, щоб нове число виявилось найбільшим з можливих. *Примітка:* щоб отримати випадкове число, є функція `mt_rand()`. Якщо вам потрібно випадкове число від 100 до 300 (включно), наприклад, використовуйте `mt_rand(100, 300)`.

Варіант 2. Завдання 2

Робота з циклами

1) Написати функцію, яка приймає кількість рядків та стовпців таблиці, і виводить на екран таблицю, що складається з $n \times n$ комірок різного кольору.

Варіант 3. Завдання 2.

Написати функцію, яка приймає число n - кількість квадратів. Функція має вивести на чорному тлі n червоних квадратів випадкового розміру в випадковій позиції в браузері.

Заключення

Отримано практичні навички роботи з установки і роботи з одним з основних засобів розробки додатків на мові програмування PHP [1] в середовищі розробки Eclipse [2]

5. Контрольні питання

1. Що таке EclipsePHP?
2. Яка версія середовища розробки сьогодні актуальна?
3. Які коментарі ви знаєте?
4. Як створити проект в Eclipse?
5. Як вивести інформацію з програми?

6. Список рекомендованої літератури

- 1.PHP [Електронний ресурс]. – Режим доступу до ресурсу: <https://www.php.net/docs.php>
- 2.Eclipse [Електронний ресурс]. – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/Eclipse>
3. PHP введение [Електронний ресурс]. – Режим доступу до ресурсу: <https://www.php.net/manual/ru/introduction.php>