МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ПРИКАРПАТСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВАСИЛЯ СТЕФАНИКА

М. С. ДУТЧАК, І. М. ЛАЗАРОВИЧ

Лабораторний практикум із дисципліни “Програмування мовою PHP”

Івано-Франківськ
2023 р.

УДК 004.51
Д 84
Рекомендовано до друку Вченою радою факультету математики та інформатики Прикарпатського національного університету імені Василя Стефаника» (протокол №7 від 22.08.2023 р.)

Рецензенти:

Кузь М. В. – доктор технічних наук, професор кафедри інформаційних технологій факультету математики та інформатики Прикарпатського національного університету імені Василя Стефаника;

Ткачук В.М. – кандидат фізико-математичних наук, доцент кафедри інформаційних технологій факультету математики та інформатики Прикарпатського національного університету імені Василя Стефаника.

Дутчак М.С.

Д 84 Дутчак М.С, Лазарович І.М. Лабораторний практикум із дисципліни “Програмування мовою PHP”, Івано-Франківськ: ПНУ ім. В. Стефаника, 2023, 143 с.

У лабораторному практикумі наведено завдання та рекомендації щодо виконання лабораторних робіт з дисципліни «Програмування мовою PHP», що дозволяє здобувачам вищої освіти освоїти технології Web-програмування зі сторони сервера мовою програмування PHP.

Методичні вказівки рекомендовано для здобувачів вищої освіти спеціальності 121 «Інженерія програмного забезпечення» та інших спеціальностей ІТ-галузі.

УДК 004. 51
Д 84

Передмова

Враховуючи сучасні тенденції розвитку інформаційних технологій, зокрема дедалі ширше впровадження Web-сервісів та Web-технологій практично в усіх галузях діяльності людини, освоєння інструментальних засобів розробки серверної частини Web-ресурсів є актуальним завданням для сучасного фахівця з Web-програмування, яке передбачає процес проектування, розробки та супроводу веб-сайту або веб-додатку.

Даний лабораторний практикум орієнтований на надання студентам теоретичних відомостей та практичних завдань із детальним описом ходу їх виконання, в межах яких розглянуто архітектури типових веб-застосувань, описано програмування на стороні сервера з використанням мови PHP. Також у даній роботі описано загальний синтаксис мови PHP, основні мовні конструкції, способи визначення функцій, забезпечення роботи з базами даних на прикладі СУБД MySQL. Також наведено відомості по роботі із сесіями та файлами cookies. Значну увагу приділено обробці масивів та рядків за допомогою стандартних методів PHP, також розглянуто особливості синтаксису та застосування регулярних виразів для формування шаблонів пошуку чи верифікації необхідної інформації. Наведений теоретичний матеріал супроводжується значною кількістю прикладів. Кожна лабораторна робота супроводжується контрольними питаннями для самоперевірки.

Наведений матеріал призначений для студентів, які вивчають Web-програмування, може бути використаний ними для самостійної роботи, а також для підготовки до виконання лабораторних чи практичних робіт. Для розуміння та засвоєння поданих відомостей необхідно володіти навиками розробки веб-сторінок на основі HTML, CSS та Javascript.

Лабораторна робота №1. Програмування з боку сервера. Введення в PHP. Методи GET та POST.

Тривалість: 2 акад. години

Мета: набути практичних навичок з використання базових операторів PHP.

Завдання: ознайомитися із теоретичними відомостями та виконати завдання відповідно до ходу роботи.

1.1 Теоретичні відомості

1.1.1 Стилі PHP- дескрипторів

Існують чотири стилі PHP- дескрипторів (наведені нижче фрагменти коду еквівалентні):

– XML- стиль

<?php echo ‘<p>Hello world!</p>‘; ?>

Даний стиль найбільш поширений (у даних методичних вказівках використовується саме такий стиль), адміністратори серверів не мають можливості вимкнути його, тому він гарантовано доступний у всіх випадках, що важливо у випадку розробки програмного коду, виконання якого передбачається у різних середовищах.

– Скорочений стиль

<? echo ‘<p>Hello world!</p>‘; ?>

Це найпростіший стиль, що відповідає стилю інструкцій обробки мови SGML(StandardGeneralizedMarkupLanguage). Використання його не бажане, оскільки системні адміністратори часто його вимикають для запобігання конфліктів з XML-документами.

– SCRIPT-стиль

<script language=’php’>echo ‘ Hello world! ‘; </script>

SCRIPT- стиль часом застосовують у випадку виникнення проблем з іншими стилями у HTML-редакторах.

– ASP-стиль

<% echo ‘<p> Hello world!</p>‘; %>

Аналогічний стиль дескриптора використовується у технологіях ASP (ActiveServerPages). Такому стилеві надають перевагу при роботі з редактором, орієнтованим на ASP.NET.

1.1.2 Оператори PHP

Дії, котрі повинен виконати інтерпретатор PHP, задаються операторами PHP, розташованими між відкриваючими та закриваючими дескрипторами:

echo ‘ Hello world! ‘;

Оператор echo виконує виведення у вікно браузера переданого йому рядка. Для розділення операторів використовується символ-крапка з комою: “;”.

1.1.3 Пробіли

Порожні символи, такі як порожній рядок (повернення каретки), пробіли між словами та символи табуляції, утворюють категорію пробільних. Браузери ігнорують пробільні символи у HTML-коді. Аналогічно діє і механізм PHP. Пробіли між PHP- операторами не є необхідними, проте вони підвищують читабельність коду. Фрагменти

echo ‘Вітаємо’;
echo ‘на нашому сайті’;
echo ‘Вітаємо’; echo‘ на нашому сайті’;

еквівалентні, але перша версія візуально сприймається легше.

1.1.4 Коментарі

Зазвичай коментарями супроводжуються більшість PHP-сценаріїв, за виключенням найпростіших.

Інтерпретатор PHP ігнорує текст, розміщений у коментарі. Для синтаксичного аналізатора коментар рівнозначний пропускам.

PHP підтримує коментарі у стилі C, C++ і сценаріїв оболонки.

/*Приклад
багаторядкового коментаря С-стилю*/

Багаторядкові коментарі починаються з символів /*і закінчуються символами */.Коментарі не можуть бути вкладеними.

Також використовують однорядкові коментарі у стилі С++:

echo ‘ Замовлення виконано ’; // початок виведення замовлення

або у стилі сценаріїв оболонки

echo ‘ Замовлення виконано ’; # початок виведення замовлення.

1.1.5 Доступ до змінних форми

Сенс використання форми полягає в отриманні скриптом інформації, введеної користувачем у текстові поля форми. Розглянемо приклад:

```
<html>
<head>
<title>Приклад форми</title>
</head>
<body>
<form action="form.php" method="post">
<input type="text" name="formvariable">
<input type="submit" value="Додати">
</form>
</body>
</html>
```

Даний HTML-скрипт створює форму для введення даних і кнопку “Додати”, яка передає введене скриптові form.php

Доступ до вмісту поля PHP-скриптом отримують наступним чином (файл form.php):

```
<?php
$variable=$_POST['formvariable'];
```

```
echo $variable;
?>
```

У даному випадку передача даних реалізована за допомогою методу POST. У цьому випадкові відбувається неявна передача: користувач не помічає ніяких зовнішніх ознак процесу і не може здійснити припущення про імена змінних. Така передача зручна з точки зору безпеки.

Існує інший спосіб передачі даних – GET. При цьому дані передаються за допомогою сформованого додатку до назви файлу, що отримує дані. Зовні виглядає як лінк, при цьому користувач отримує доступ до імен змінних, що передаються, та їхніх значень.

```
<html>
<head>
<title>Приклад форми</title>
</head>
<body>
<form action="form.php" method="get">
<input type="text" name="formvariable">
<input type="submit" value="Додати">
</form>
</body>
</html>
```

Даний приклад відрізняється від попереднього методом передачі: замість POST використано GET.Нехай у поле введено значення “test”. Після натискання кнопки “Додати” буде особливим чином активізовано скрипт form.php: до адреси файлу буде додано символ “?”, назву змінної, переданої з форми, знак “=” та значення змінної:

http://localhost/form.php?formvariable=test

PHP-скрипт при цьому змінюють наступним чином:

```
<?php
$variable=$_GET['formvariable'];
echo $variable;
?>
```

Масиви $_POST та $_GET належать до категорії суперглобальних. Як у першому, так і у другому випадках, доступ до даних форми можна отримати через масив $_REQUEST.

```
<?php
$variable=$_REQUEST['formvariable'];
echo $variable;
?>
```

Детально використання масивів буде розглянуто у наступних лабораторних роботах.

Зазвичай блоки отримання даних з форми розташовують на початкові скрипта.

1.1.6 Конкатенація рядків

Операція конкатенації рядків використовується для об’єднання рядків (фрагментів тексту) у єдиний текст. Вона часто застосовується при виведенні даних у браузер за допомогою оператора echo. Реалізується за домового символу оператора конкатенації – крапки “.”.

```
<?php
$text1="Hello, ";
$text2="world!";
echo $text1.$text2;
?>
<?php
$text1="world!";
echo "Hello, ".$text1;
?>
```

Довільну змінну, відмінну від змінної масиву, можна помістити у подвійні лапки і застосувати до неї оператор echo. Зверніть увагу на результат виконання двох наступних скриптів:

```
<?php
$text1="world!";
```

```
echo "Hello, $text1";
?>
```

Буде виведено у вікно браузера: Hello, world!

```
<?php
$text1="world!";
echo 'Hello, $text1';
?>
```

Результат: Hello, $text1.

Якщо ім’я змінної знаходиться між подвійними лапками, то ім’я замінюється значенням змінної, якщо ім’я змінної чи довільний текст обмежені одинарними лапками, то вони передаються без змін.

1.1.7 Ідентифікатори

Ідентифікатори – це імена змінних, функцій та класів. Використання ідентифікаторів регламентується наступними правилами:

– ідентифікатори можуть мати довільну довжину і складатися з букв, цифр та символів підкреслювання “_”;

– ідентифікатори не можуть починатися з цифри;

– у PHP ідентифікатори чутливі до регістру символів. Змінні $formvariable $FormVariable– це різні змінні. Виключення складають вбудовані PHP-функції – їхні імена можуть бути представлені довільним регістром;

– змінні можуть мати імена, що співпадають з іменами вбудованих функції. Однак, це може призвести до плутанини, тому подібних ситуацій слід уникати. Не можна також створювати функції, чиї імена співпадають з іменами вбудованих функцій.

Імена змінних у PHP починаються знаком долара ($). Пропуск цього знаку – поширена помилка.

1.1.8 Типи змінних

Тип змінної характеризується видом даних, котрі вона зберігає. PHP підтримує наступні базові типи даних:

– Integer (цілий) – використовується для представлення цілих чисел;

– Float (ще їх називають double – подвійної точності) – використовується для представлення дійсних чисел;

– String (рядковий) – використовується для представлення символів;

– Boolean (булевий) – використовується для зберігання значень true та false;

– Array(масив) – використовується для зберігання декількох елементів даних;

– Object (об’єкт) – використовується для зберігання екземплярів класу.

Доступні також два спеціальних типи – NULL та resource (ресурс). Змінні, котрим не присвоєно конкретного значення, котрі не визначені або набувають значення NULL, належать до типу NULL. Деякі вбудовані функції (такі, як для роботи з базами даних) повертають змінну ресурсного типу (наприклад, з’єднання з базами даних).

Також PHP підтримує типи pdfdoc та pdfinfo, якщо встановлена підтримка обробки PDF-документів.

Мова PHP – слабо типізована. На відміну від мови C, тип змінної визначається типом присвоєного їй значення.

1.1.9 Перетворення типів

За допомогою механізму перетворення типів можна переводити змінну або конкретне значення до іншого типу. Перетворення відбувається так само, як і на мові C.

```
<?php
$a=2.34;      // Змінна $a має тип float
$a=(int)$a;   // Тепер змінна $a має тип integer
echo $a;
?>
```

1.1.10 Змінні змінних

Усі мови програмування дозволяють змінювати значення змінної, деякі – тип змінної і зовсім небагато мов дозволяють змінювати ім’я змінної.

В основі цієї можливості – ідея використання значення однієї змінної як імені іншої.

Нехай існує змінна $variable. Тоді:

```
<?php
$variable=NULL;
$varname="variable";
$$varname="Hello, world!";
echo $variable;
?>
```

Запис $$varname="Hello, world!" еквівалентний наступному: $variable="Hello, world!".

1.1.11 Константи

Разом із використанням змінних, PHP надає можливість оголошення констант. Як і змінна, константа містить значення, але її значення встановлюється одноразово і не може бути зміненим у сценарії.

Константи оголошуються за допомогою функції define

```
<?php
define('CONSVALUE',10);
echo CONSVALUE;
?>
```

Зверніть увагу на те, що імена констант записують символами у верхньому регістрі. Дана особливість перейшла з мови С. Дотримуватися її не обов’язково, проте вона значно спрощує читання і супровід коду.

Важливо: при звертанні до констант не використовується знак долара ($).

1.1.12 Область дії змінних

Термін “область дії” належить тим розділам сценаріїв, в яких можливий доступ до деякої конкретної змінної, іншими словами, це – область, з довільного місця якої видно дану змінну. У PHP використовують шість базових правил визначення області дії:

– вбудовані суперглобальні змінні видно з довільного місця сценарію;

– константи, щойно вони оголошені, видно глобально, тобто можуть використовуватися як зовні так і всередині функцій;

– глобальні змінні, оголошені у сценарії, видно у довільному місці сценарію, але не всередині функцій;

– змінні, що використовуються всередині функцій, що оголошені як глобальні, посилаються на глобальні змінні з тими самими іменами;

– змінні, що використовуються всередині функцій, що оголошені як статичні, невидимі поза межами функцій, проте вони зберігають свої значення поміж двома викликами цих функцій;

– змінні, створені всередині функцій, є локальними стосовно своєї функції і припиняють існування після завершення функції.

Вбудовані суперглобальні змінні у PHP версій від 4.1 і вище наступні:

– $_GLOBALS – масив глобальних змінних. Дає можливість доступу до глобальних змінних всередині функції, наприклад: $_GLOBALS[‘myvariable’];

– $_SERVER – масив змінних середовища сервера;

– $_GET – масив змінних, переданих у сценарій за допомогою методу GET;

– $_POST – масив змінних, переданих у сценарій за допомогою методу POST;

– $_COOKIE – масив cookie- змінних;

– $_FILES – масив змінних завантаження файлів;

– $_ENV – масив змінних оточення;

– $_REQUEST – масив, пов’язаний із введенням даних користувачем, включаючи $_GET, $_POST, $_COOKIE;

– $_SESSION – масив змінних сеансу.

1.1.13 Посилання

Операція посилання позначається як “&” і може використовуватися у поєднанні з операцією присвоювання.

Зазвичай, коли значення змінної $a присвоюється змінній $b, створюється копія змінної $a, яка зберігається у пам’яті. Якщо у майбутньому значення $a буде змінене, то $b зміни не торкнуться.

```
$a=5;
$b=$a;        // $b=5;
$a=6;         // як і раніше, $b=5
```

Створення копії можна уникнути, використавши операцію посилання &:

```
$a=5;
$b=&$a;      // $b дорівнює 5;
$a=6;        // зверніть увагу: $b дорівнює 6
```

$a та $b вказують на одну і ту ж ділянку пам’яті. Цей зв’язок можна розірвати, “скинувши” одну зі змінних:

```
unset($a);
```

При цьому значення $b продовжуватиме існувати.

1.1.14 Операції порівняння

Операції порівняння наведено у табл.1.1. Результатом виконання операції порівняння є true або false.

Таблиця 1.1 – Операції порівняння PHP

Операція	Назва	Використання
==	дорівнює	$a == $b
===	тотожно	$a === $b
!=	не дорівнює	$a != $b
!==	не тотожно	$a !== $b
<>	не дорівнює	$a <> $b
<	менше	$a < $b
>	більше	$a > $b
<=	менше або дорівнює	$a <= $b
>=	більше або дорівнює	$a >= $b

Операція перевірки тотожності повертає значення true тільки у тому випадкові, якщо обидва операнди рівні і мають однаковий тип. Наприклад:

```
$a=0;
$b='0';
echo($a==$b);      // Результат:  true
echo($a===$b);     // Результат: false
```

Логічні операції слугують для комбінування результатів логічних умов. Перелік логічних операцій разом з описом їхнього застосування наведено у табл. 1.2.

Таблиця 1.2 – Логічні операції PHP

Операція	Назва	Використання	Результат
!	НЕ	!$b	Повертається true, якщо значення $a дорівнює false та навпаки
&&	І	$a && $b	Повертається true, якщо обидві змінні $a та $b мають значення true, в іншому випадкові повертається значення false
||	АБО	$a || $b	Повертається true, якщо довільна зі змінних $a або $b має значення true, інакше повертається false
and	І	$a and $b	Те саме, що і &&, але з меншим пріоритетом
or	АБО	$a or $b	Те саме, що і ||, але з меншим пріоритетом

1.1.15 Оператори if

Для прийняття рішень використовується оператор if. Операторові необхідно задати умову. Якщо умова рівна true, то виконується блок коду, розташований після оператора. Умова в операторі if записується поміж круглими дужками “(…)”. Приклад:

```
<?php
$a=0;
$b=3;
if($a<$b) echo '$a < $b';
?>
```

1.1.16. Блоки коду

Часто всередині такого умовного оператора, як if, необхідно виконати більше одного оператора. У такому випадку відповідна послідовність операторів записується у вигляді блоку. Для оголошення блоку оператори необхідно оточити фігурними дужками:

```
if($a<$b)
{
echo '$a < $b';
```

```
$a=$b;
}
```

Таким чином при виконанні умови ($a<$b) буде виконано обидва оператори

```
echo '$a < $b';
$a=$b;
```

Якщо ж умова не виконуватиметься, то обидва оператори будуть проігнорованими.

1.1.17. Оператор else

Оператор else дозволяє визначити альтернативну дію, котра повинна виконатися, якщо значення умови в операторі if виявиться false. Наприклад:

```
<?php
$a=0;
$b=3;
echo '$a='.$a.'; $b='.$b.'<br />';
if($a<$b)
echo '$a < $b';
else
echo '$a < $b';
?>
```

Результат виконання скрипта:

```
$a=0; $b=3
$a< $b
```

Вкладання операторів if один в одного дозволяє будувати складні ланцюжки.

1.1.18 Оператор elseif

У багатьох випадках прийняття рішення передбачає вибір відповідного варіанту з деякої множини можливих варіантів. Послідовність цієї множити

можна створити за допомогою оператора комбінації операторів if – else. За наявності послідовності умов програма може перевіряти кожну з них до тих пір, поки не знайде таку, значення якої буде true. Наприклад:

```
<?php
$a=23;
if($a>0 && $a<10) echo 'Результат - у першому інтервалі';
elseif($a>=10 && $a<20) echo 'Результат - у другому інтервалі';
elseif($a>=20 && $a<30) echo 'Результат - у третьому інтервалі';
?>
```

Після виконання скрипта отримують: “Результат - у третьому інтервалі”

1.1.19 Оператор switch

Оператор switch працює аналогічно до оператора if, але надає можливість умовному виразові мати як результат більше двох значень. В операторі if умова набуває значення true або false. Натомість в операторі switch умова може набувати довільну кількість значень у тих випадках, коли результат його обчислення – простий тип (integer, string чи float). Для забезпечення реагування на кожне таке значення слід передбачити для нього відповідний оператор case, а також (не обов’язково) визначити дії, що виконуватимуться по замовчуванню, якщо виникне випадок, не передбачений оператором case. Наприклад:

```
<?php
$a=1;
switch ($a)
{
case 1:
echo '$a=1';
break;
case 2:
echo '$a=2';
break;
default:
echo '$a='.$a;
```

```
break;
}
?>
```

Результат виконання: $a=1.

1.1.20 Цикл while

Найпростішим циклом у PHP є цикл while. Різниця між оператором if та оператором while полягає у тому, що у випадку виконання умови оператор if виконує блок коду тільки один раз, а оператор while повторює його до тих пір, поки умова виконується.

Наприклад:

```
<?php
$var=1;
while($var<=5)
{
echo $var.'<br />';
$var++;
}
?>
```

1.1.21 Цикл for

Цикл типу while можна записати і у більш компактній формі за допомогою оператора for. Базова структура оператора for має вигляд:

```
for(вираз 1; умова; вираз 2)
вираз 3;
```

Вираз 1 виконується один раз на початкові циклу. Як правило, він встановлює початкове значення змінної циклу.

Вираз умова перевіряється перед кожною ітерацією. Якщо цей вираз повертає значення false – цикл зупиняється.

Вираз 2 виконується у кінці кожної ітерації. Зазвичай у ньому змінюється значення лічильника.

Вираз 3 виконується один раз під час кожної ітерації. Це, як правило, – блок коду, який містить тіло циклу. Наступний приклад виводить табличку з двох стовпчиків і п’яти рядків.

```
<?php
echo "<table>";
for($distance=50; $distance<=250; $distance+=50)
{
echo "<tr>\n <td align='right'> $distance </td>\n";
echo "<td align=right'> ".$distance." </td>\n</tr>\n";
}
echo "</table>";
?>
```

1.1.22 Цикл do…while

Загальні структура циклу do…while має вигляд:

```
do
вираз;
while(умова 1);
```

Цикл do…while відрізняється від циклу while тим, що в ньому умова перевіряється у кінці. Отже, у циклі do…while оператор або блок операторів всередині циклу завжди виконується, принаймі, один раз.

```
<?php
$a=10;
do
{
echo $a.'<br />';
$a-=1;
}
while($a>=0);
?>
```

Додаткові теоретичні відомості:

Синтаксис PHP

PHP Manual

1.2 Хід роботи

Примітки. Інструкцію для виконання перших двох завдань можете знати у додатках А-Г.

1. Зареєструйтеся на веб-хостингу (наприклад https://infinityfree.net/ (рекомендовано), https://www.000webhost.com/, https://www.zzz.com.ua, https://profreehost.com/ тощо. Встановіть з'єднання із веб-хостингом через FTP. Якщо Ви вже є зареєстровані і термін дії закінчується не швидше 30 червня поточного навчального року, то повторно реєструватися не потрібно.
2. Налаштуйте редактор коду (VS Code, Sublime, Notepad++, тощо) для роботи з FTP-сервером.
3. Скопіюйте на FTP-сервер папку PHP. Проаналізуйте запропоновані викладачем файлами. Перегляньте їх через браузер.
4. У файл config.php в межах блоку php додайте наступний код:

```
$LastModified_unix = strtotime(date("D, d M Y H:i:s", filectime($_SERVER['SCRIPT_FILENAME'])));
$LastModified = gmdate("D, d M Y H:i:s \G\M\T", $LastModified_unix);
echo "Last modified: $LastModified"."<br>";
```

Файл config.php повинен бути приєднаним до всіх файлів всіх лабораторних робіт і бути єдиним (цієї вимоги також потрібно дотримуватися щодо інших файлів і папок, які знаходяться безпосередньо в папці Web, окрім папок labN), тоді даний код буде виводити повідомлення, коли будь-який із файлів був останній раз змінений.

5. Реалізуйте і дослідіть функціонування операторів, дія яких розглядається у теоретичних відомостях. Посилання (назва повинна містити номер підпункту прикладу) на файл із реалізацією кожного із прикладів розмістити у файлі lab1.php.
6. Модифікуйте приклад в файлі example1_1_5_1.php наступним чином:
   - вбудуйте html-код в php-блок;
   - додайте ще одну форму із текстовим полем і кнопкою, яка б методом GET передавала в example1_1_5_2.php введене число. Одержане число у файлі example1_1_5_2.php помножте на 2, виведіть із вказівкою того, що вхідні дані одержані із файлу example1_1_5_1.php;

7. Дано: три цілих числа, що є значеннями температури та ціле число (зі значенням 1, 2, або 3), яке є номером завдання. Номер завдання вказує, які дії потрібно виконувати зі значеннями температури.

Знайти: максимальну, або мінімальну, або середню температуру в залежності від введеного номера завдання.

Напишіть код, який би виводив на сторінці такий текст:

“Введіть номер завдання:

1 - обчислення максимальної температури;

2 - обчислення мінімальної температури;

3 - обчислення середньої температури.”

Створіть 4 текстових поля для введення значення температури і номера завдання та кнопку для передачі даних.

При нажатті на кнопку повинно виконатися завдання, номер якого буде введено (використайте оператор Switch).

8. На Google Drive створіть папку з назвою предмету, для облікового запису mariia.dutchak@pnu.edu.ua надайте до неї доступ з правами коментування, для всіх решту встановити обмежений доступ. При наданні доступу зніміть відмітку біля сповістити.

Увага! Цю та всі наступні роботи потрібно розміщувати на вебхостингу, копії папок із виконаними роботами завантажуєте у папку дисципліни на Google Drive.

9. Для здачі робіт, у Classroom присилаєте посилання на відповідні папки з роботами, розміщені на Google Drive і знімок екрану, який демонструє розміщення файлів із виконаною роботою на вебхостингу.

10. Всі наступні роботи теж розміщуйте на вебхостингу, папки із окремими лабораторними роботами називайте labN_lastname, де N- номер роботи, lastname - Ваше прізвище. На стартовій сторінці дисципліни index.php робіть посилання на стартові сторінки відповідних робіт під назвою labN.php, N-номер роботи, в них створіть посилання на виконані завдання відповідної роботи, і зворотні посилання на відповідну сторінку labN.php. Файл labN.php має знаходитися у відповідній папці labN_lastname. Також файл index.php із папки дисципліни повинен містити Ваше ПІБ та номер варіанту (порядковий номер у списку групи).

11. Посилання на стартову сторінку дисципліни на вебхостингу і посилання на папку предмету на Google Drive вкажіть тут (форму заповнюєте тільки один раз, за винятком зміни адреси вебхостингу чи папки предмету на Google Drive).

12. Здайте і захистіть лабораторну роботу.

1.3 Контрольні запитання:

1. Які типи PHP- дескрипторів існують на даний час?
2. Поясніть суть оператора у PHP.
3. Які особливості використання пробілів?
4. Які типи коментарів використовують при програмуванні на PHP?
5. Яким чином реалізують доступ до змінних форми?
6. Що таке конкатенація рядків та як вона реалізується?
7. Поясніть зміст поняття “ідентифікатор” на PHP.
8. Які типи змінних підтримує PHP?
9. Поясніть суть поняття “змінна змінних”?
10. Яким чином оголошуються константи на PHP?
11. Поясніть суть поняття “область дії змінних”?
12. Яке призначення посилань?
13. Операції порівняння на PHP.
14. Яким чином реалізують розгалуження у програмі?
15. Які оператори для створення циклів надає PHP?

Лабораторна робота №2. PHP. Математичні функції. Оператори. Передача та отримання даних

Тривалість: 2 акад. години

Мета: набути практичних навичок з використання базових операторів PHP.

Завдання: виконайте завдання згідно ходу роботи.

2.1 Теоретичні відомості

Передача кількох змінних через посилання:

 Link text

Додаткові теоретичні відомості:

1. Теоретичні відомості Лабораторної роботи 1
2. Синтаксис PHP
3. Математичні функції
4. Передача значення змінних через посилання

2.2 Хід роботи

1. Створіть форму з кнопкою типу Submit та двома текстовими полями, в які користувач вводить два числа. Далі після натискання кнопки Submit викликається php-скрипт, який виводить три наступні варіанти дії між цими числами: наприклад 5-2 = 3; 5 * 2 = 10; 5%2 = 1.
2. Створіть форму з двома (трьома) текстовими полями і кнопкою типу Submit, в поля користувач вводить два (три) натуральні числа. Далі після натискання кнопки Submit викликається php-скрипт, який перевіряє чи числа натуральні, якщо так, то обчислює і виводить значення виразу згідно варіанту, а також введені числа:

№ п/п	Варіант завдання	№ п/п	Варіант завдання
1	$F = x + 2y$	16	$F = \frac{1}{x + y + z}$
2	$F = 2x - 3y^2$	17	$F = \frac{x - y}{2xz}$
3	$F = \frac{1 - x^2}{y^3}$	18	$F = 21xy - xz + xyz$
4	$F = 2x + 3y - z$	19	$F = x^3 + x^2 + x + xy^2$
5	$F = 3xy - y + 2x$	20	$F = x + y^2 + z^{0.5}$
6	$F = \frac{x}{y} + \frac{y^2}{1 - xy}$	21	$F = 2x + 3y - z$
7	$F = x - xy + y^2$	22	$F = x^6 + x^5 y + 0.5z + x^2$
8	$F = x + \sqrt{y}$	23	$F = \frac{xz}{y}$
9	$F = xy + xz - yz$	24	$F = \frac{\sqrt{x}}{y + z}$
10	$F = x^{y^z} + 2xz$	25	$F = \frac{x - y^{0.5}}{xz}$
11	$F = x^6 + x^5 y + 0.5z + x^2$	26	$F = -3xz + \sqrt{x^3}$
12	$F = \left(1 - \frac{x}{y}\right)^2 + z$	27	$F = x^{y^z} + 2xz$
13	$F = x^y + y^z + z^x$	28	$F = \frac{\sqrt{y}}{\sqrt[3]{xz}}$
14	$F = 1 - x - y - z$	29	$F = 2x + y - \frac{z}{xy}$
15	$F = x + y^2 + z^{0.5}$	30	$F = \frac{\sqrt{x}}{y + z}$

3. Створіть форму з двома текстовими полями і кнопкою типу Submit, в поля користувач вводить два цілі числа. Далі після натискання кнопки Submit викликається php-скрипт, який перевіряє чи число ціле, якщо так, то обчислює і виводить значення виразу згідно варіанту, а також введені числа:

№ п/п	Варіант завдання	№ п/п	Варіант завдання
1	$F=\begin{cases} x+y, xy>0 \\ x, x^2>100 \\ x^3+y, \text{інакше} \end{cases}$	16	$F=\begin{cases} x+y, xy\neq 0 \\ x, x^3>100 \\ x^3+y, \text{інакше} \end{cases}$
2	$F=\begin{cases} x+y, (x>0)\,ma\,(y>0) \\ x^2+y^2, (x<3)\,ma\,(y<4) \\ x^3, \text{інакше} \end{cases}$	17	$F=\begin{cases} 2x+y, (x>0)\,ma\,(y>0) \\ x^2+2y^2, (x<3)\,ma\,(y>4) \\ x^3, \text{інакше} \end{cases}$
3	$F=\begin{cases} x^2+y^2, (x<10)\,або\,(y>3) \\ x-y+x^2, x>2 \\ x^3+3xy, \text{інакше} \end{cases}$	18	$F=\begin{cases} 2x^2-3y^2, (x<10)\,або\,(y\neq 3) \\ 2x-3y/x+x^2, x\neq 2 \\ x^3+3xy, \text{інакше} \end{cases}$
4	$F=\begin{cases} x/y+xy^2, (x>0)\,ma\,(y>0) \\ y/x, (-x+y)/2>0 \\ x^2, \text{інакше} \end{cases}$	19	$F=\begin{cases} 2x/y+xy^2, (x>0)\,ma\,(y\neq 0) \\ y/2x, (2x+y)/2\neq 0 \\ x^2, \text{інакше} \end{cases}$
5	$F=\begin{cases} x-y^2, x-y>2 \\ x/y, x>0 \\ x^3-y^2, \text{інакше} \end{cases}$	20	$F=\begin{cases} x-2y^2, x-3y\neq 2 \\ x-y/y, x>0 \\ 2x^2-3y, \text{інакше} \end{cases}$
6	$F=\begin{cases} x/2y, xy<-100 \\ x^2-xy, xy>20 \\ x^3+2, \text{інакше} \end{cases}$	21	$F=\begin{cases} 2x/y, xy\neq -100 \\ x^3-2xy, xy\neq 20 \\ x^2+4, \text{інакше} \end{cases}$
7	$F=\begin{cases} 2xy-3, x/y>0 \\ xy/3, (x>0)\,ma\,(y<3) \\ 2x-3y, \text{інакше} \end{cases}$	22	$F=\begin{cases} xy-1, x/y>0 \\ 2xy/2, (x\neq 0)\,ma\,(y\neq 3) \\ x-y, \text{інакше} \end{cases}$

8	$F=\begin{cases}2x+3y, x<3y\\ xy+x/3y, (y^2>2xy)або(y>0\\ x^2+y, інакше\end{cases}$	23	$F=\begin{cases}10x+y, x\neq y\\ 2xy+4x/y, (y^2\neq 2xy)або(y<0)\\ x+y^2, інакше\end{cases}$
9	$F=\begin{cases}2\sqrt{x}\cdot y, (y>3)та(x>0)\\ x/y^2, (yx^2/2>3x)або(x^2>2y\\ x^2/y, інакше\end{cases}$	24	$F=\begin{cases}x^3, x^3\neq 3\\ xy, (x\neq 0)та(y<0)\\ 2x+xy-x/2y, інакше\end{cases}$
10	$F=\begin{cases}3xy-x, xy>0\\ 2x-3y/x, (7x>2y)або(y>2)\\ x-y, інакше\end{cases}$	25	$F=\begin{cases}xy-x, 2x/y\neq 0\\ (x-2y)/x, (14x\neq y)або(2y<-4\\ x-y, інакше\end{cases}$
11	$F=\begin{cases}3xy, x>0\\ 2x/y, (x<0)та(y>3/2)\\ x^2+y^3, інакше\end{cases}$	26	$F=\begin{cases}x/y, x\neq 0\\ 2x/y, (x\neq 0)або(y>3/2)\\ -x^3+2y^2, інакше\end{cases}$
12	$F=\begin{cases}x-y, xy>0\\ x^2-y/x, (y<0)та(x>3y)\\ 3xy-x^2y^2, інакше\end{cases}$	27	$F=\begin{cases}2x-y, 2xy\neq 0\\ x^2+2y/x, (y\neq 0)або(x\neq 3y)\\ x/y-xy, інакше\end{cases}$
13	$F=\begin{cases}2\sqrt{x}\cdot y, (y>3)та(x>0)\\ x/y^2, (yx^2/2>3x)або(x^2>2y\\ x^2/y, інакше\end{cases}$	28	$F=\begin{cases}(3/5)\sqrt{x}\cdot y, (y\neq 5)та(x\neq 0)\\ x/y^2, (yx^2\neq 3x)або(x^3\neq 5xy)\\ x/y^2, інакше\end{cases}$
14	$F=\begin{cases}-3x+y^3, (x>0)та(y<0)\\ x+y-x^2y, xy-x<3\\ x^2+y^2, інакше\end{cases}$	29	$F=\begin{cases}x+y^2, (x\neq 0)та(y\neq 0)\\ x-y+x^3y^2, xy-x>=3\\ x^2-y^2, інакше\end{cases}$

15	$F=\begin{cases}3xy-x, xy>0\\ 2x-3y/x, (7x>2y) або (y>2)\\ x-y, інакше\end{cases}$	30	$F=\begin{cases}2x-y, 2xy\neq 0\\ x^2+2y/x, (y\neq 0) або (x\neq 3y)\\ x/y-xy, інакше\end{cases}$

4. Створіть веб-сторінку, яка показує текстове повідомлення «Виберіть, будь ласка, зображення <рандомна із 4-х назва згідно теми>» та відповідні зображення як варіанти відповіді. Після натискання на малюнок скрипт PHP перевіряє правильність відповіді та інформує про результати перевірки.

Теми :

1. Птахи
2. Їжа
3. Звірі
4. Домашні тварини
5. Прапори
6. Гриби
7. Транспорт
8. Дерева
9. Фрукти
10. Овочі
11. Ягоди
12. Косметика
13. Кондитерські вироби
14. Двоколісний транспорт
15. Меблі
16. Комп’ютерна техніка
17. Жіночий одяг
18. Чоловічий одяг
19. Дитячий одяг
20. Взуття
21. Солодощі
22. Спортивні аксесуари
23. Музичні інструменти
24. Транспорт
25. Професії
26. Гроші
27. Кухонна техніка
28. Побутова техніка
29. Автомобілі
30. Комп’ютерна техніка

Приклад виконання.

Запитання: Натисніть, будь ласка, на зображення кота




Оцінка правильності:

Ви вибрали равлика

Це неправильно

5. Модифікуйте попереднє завдання так, щоб випадковим чином відображалися 4-ри картинки із доступних 6.

Увага! Текстове повідомлення «Виберіть, будь ласка, зображення <рандомна із 4-х назва згідно теми >» має залишатися.

6. Здайте і захистіть лабораторну роботу.

2.3 Контрольні запитання:

1. Які типи PHP- дескрипторів існують на даний час?
2. Поясніть суть оператора у PHP.
3. Які особливості використання пробілів?
4. Які типи коментарів використовують при програмуванні на PHP?
5. Яким чином реалізують доступ до змінних форми?
6. Що таке конкатенація рядків та як вона реалізується?
7. Поясніть зміст поняття “ідентифікатор” на PHP.
8. Які типи змінних підтримує PHP?
9. Поясніть суть поняття “змінна змінних”?
10. Яким чином оголошуються константи на PHP?
11. Поясніть суть поняття “область дії змінних”?
12. Яке призначення посилань?
13. Операції порівняння на PHP.
14. Яким чином реалізують розгалуження у програмі?
15. Які оператори для створення циклів надає PHP?

Лабораторна робота №3. Масиви у PHP. Багатократне використання коду. Створення функцій.

Тривалість: 2 акад. години

Мета: набути практичних навичок з використання масивів та багатократного використання програмного коду.

Завдання: дослідіть використання масивів, створення функцій користувача та можливості включення у програмний код раніше розроблених файлів. Виконайте завдання згідно ходу роботи.

3.1 Теоретичні відомості

3.1.1. Використання масивів у PHP

У більшості мов програмування масиви мають числові індекси, котрі, як правило, починаються з нуля чи одиниці.

PHP дозволяє використовувати як індекси числа або рядки.

Масив створюється наступним чином:

```
$a = array('data1', 'data2', 'data3');
$b = array(1, 2, 3);
```

Відповідно, доступ до елементів масиву, отримують так:

```
echo '$a[0]='.$a[0].'<br/>';
echo '$a[1]='.$a[1].'<br/>';
echo '$a[2]='.$a[2].'<br/>';
echo '$b[0]='.$b[0].'<br/>';
echo '$b[1]='.$b[1].'<br/>';
echo '$b[2]='.$b[2].'<br/>';
```

Якщо у масиві необхідно зберегти зростаючу послідовність чисел, використовують функцію range(). Аналогічно функція утворює послідовність символів:

```
<?
$numbers = range(1, 10);
$symbols=range('a', 'j');
```

```
for($i=0; $i<10; $i++)
echo $numbers[$i].'---'.$symbols[$i].'<br/>';
?>
```

Даний приклад генерує масив із десяти чисел у діапазоні від 1 до 10 та послідовність символів від “a” до “j”.

Для доступу до елементів масиву зручно використовувати цикл foreach, який призначений, власне, для роботи з масивами.

```
<?php
$arr = range(1, 20);
foreach ($arr as $current)
echo $current.'<br/>';
?>
```

Наведений код почергово зберігає кожний елемент масиву у змінній $current і потім виводить її значення у вікно браузера.

При створенні масивів у наведених прикладах PHP було надано можливість присвоїти кожному елементу масиву індекс по замовчуванню, тобто перший доданий елемент став 0-м, другий – 1-м і так далі. PHP також підтримує масиви, у яких з кожним елементом можна пов’язати (асоціювати) довільний ключ (індекс).

У наступному прикладі символьні назви слугують як ключі, а числа – як значення.

```
<?php
$arr = array('the_first'=>1, 'the_second'=>2, 'the_third'=>3);
echo $arr['the_first'].'<br/>';
echo $arr['the_second'].'<br/>';
echo $arr['the_third'].'<br/>';
?>
```

Створити масив з багатьма елементами можна також, оголосивши масив з одним елементом і додаючи до нього пізніше решту потрібних елементів:

```
<?php
```

```
$arr = array('the_first'=>1);
$arr['the_second']=2;
$arr['the_third']=3;
foreach ($arr as $current)
        echo $current.'<br/>';
?>
```

Оскільки в асоціативних масивах індекси не є числами, то для роботи з такими масивами неможливо скористатися лічильником у циклі for. У цьому випадкові слід застосувати цикл foreach або конструкції list()та each().

```
<?php
$arr = array('the_first'=>1, 'the_second'=>2, 'the_third'=>3);
foreach ($arr as $current)
echo $current.'<br/>';
?>
```

Якщо необхідно отримати доступ до індексів масиву, використовують конструкцію:

```
<?php
$arr = array('the_first'=>1, 'the_second'=>2, 'the_third'=>3);
foreach ($arr as $key => $value)
echo '$arr['.$key.']='.$value.'<br/>';
?>
```

Результат:
$arr[the_first]=1
$arr[the_second]=2
$arr[the_third]=3.

Наведений нижче код виводить вміст масиву за допомогою конструкції each():

```
<?php
$arr = array('the_first'=>1, 'the_second'=>2, 'the_third'=>3);
while( $element = each($arr)){
```

```
echo '$arr['.$element['key'].']='.$element['value'].'<br/>';
}
?>
```

У цьому прикладі змінна $element– це масив. При викликові функції each() вона повертає масив з чотирма значеннями і чотирма індексами. Комірки key та 0 містять ключ поточного елементу, value та 1 – його значення.

Важливо: при використанні функції each() слід пам’ятати, що масив відстежує поточний елемент. Якщо в одному і тому ж сценарії необхідно двічі скористатися значеннями одного і того ж масиву, то потрібно за допомогою функції reset() встановити поточний елемент на початок масиву.

Наведемо ряд корисних функцій для роботи з масивами.

sort($array) – впорядковує елементи за алфавітом чи порядком зростання;

asort($array) – те саме, що і sort(), але для масивів з описовими індексами;

ksort($array) – впорядковує ключі у масиві з описовими індексами за зростанням;

rsort($array) – впорядковує елементи за алфавітом чи порядком спадання;

rasort($array) – те саме, що і sort(), але для масивів з описовими індексами;

rksort($array) – впорядковує ключі у масиві з описовими індексами за спаданням.

У PHP відсутня можливість порівняння двох масивів, тому для сортування багатовимірного масиву необхідно створювати деякий ручний метод.

shuffle($array) – розташовує елементи масиву випадковим чином;

array_push($array, $data) – додає елемент $data у кінець масиву $array;

$array = array_reverse($array) – змінює порядок розташування елементів на зворотній;

next($array)або each($array) – переміщують показник на елемент вперед на один елемент; next – спочатку змінює покажчик, потім – повертає значення, each–навпаки;

end($array) – встановлює покажчик на елемент у кінець масиву;

prev($array) – встановлює покажчик на елемент на одну позицію ближче до початку масиву;

current ($array) — повертає поточний елемент масиву

reset($array) – встановлює покажчик масиву на його початок.

sizeof($array) – підраховує кількість елементів у масиві.

3.1.2 Багатократне використання коду

Одне із завдань, які зазвичай ставлять перед собою розробники програмного забезпечення – повторне використання існуючого програмного коду замість написання нового. Таким чином вдається знизити вартість розробки, підвищити надійність та забезпечити певну уніфікацію.

PHP надає два оператори require() та include() для забезпечення повторного використання програмного коду.

Оператор require('шлях до файлу') слугує для включення повного коду файлу-параметру оператора у тому місці файла реципієнта, в якому знаходиться сам оператор.

При цьому розширення імені файла, який включається, ігнорується, тобто він може бути названий довільним чином. Файл, що включається, повинен мати PHP-дескриптори. Інакше вміст файла буде розглянуто як простий текст.

Оператор require()часто застосовується для створення шаблонів сторінок.

Оператор include()практично не відрізняється від оператора require() за винятком того, що require() у випадку невдалого виконання видає повідомлення про непоправну помилку, а include() – тільки попередження.

3.1.3 Створення функцій

Функцією називають незалежний модуль коду, який встановлює інтерфейс введення, виконує певну задачу і за необхідності повертає результат.

Імена функцій у PHP не чутливі до регістру.

Оголошення функції створює нову функцію. Оголошення починається ключовим словом function, воно присвоює функції ім'я, задає необхідні параметри і містить код, котрий виконується при кожному викликові функції.

Оголошення функції має вигляд:

```
function my_function(){
Echo 'викликано функцію';
}
```

Виклик функції реалізується за допомогою оператора my_function().

Вбудовані функції PHP доступні для усіх сценаріїв, а функції користувача – тільки у тому сценарії, у якому вони оголошені. Є сенс додати усі функції, що часто використовуються, в окремий файл і за допомогою оператора require()надати доступ до них.

Імена функцій користувача вибираються, враховуючи обмеження:

– функція не може мати ім’я таке саме, як у вже існуючої функції;

– ім’я функції може складатися тільки з букв, цифр та символів підкреслювання;

– ім’я функції не може починатися з цифри.

Для нормального функціонування більшість функцій вимагають передачі у них параметрів. Наприклад (виводиться таблиця з 3 рядків і 1 стовпчика):

```
<?php
function create_table($data){
echo '<table border=1>';
reset($data); // вказуємо початок
$value=current($data);
while($value){
        echo "<tr><td>$value</td></tr>\n";
        $value = next($data);
}
echo '</table>';
}
$my_array = array('Рядок 1', 'Рядок 2', 'Рядок 3');
create_table($my_array);
?>
```

Як і вбудовані функції, функції користувача можуть мати обов’язкові та необов’язкові параметри. Модифікуємо попередній приклад:

```
<?php
function create_table2($data, $border=1, $cellpadding=4, $cellspacing=4){
echo "<table border=$border cellpadding=$cellpadding"
                "cellspacing=$cellspacing>\n";
reset($data); // вказуємо початок
```

```
$value=current($data);
while($value){
        echo "<tr><td>$value</td></tr>\n";
        $value = next($data);
}
echo '</table>';
}
$my_array = array('Рядок 1', 'Рядок 2', 'Рядок 3');
create_table2($my_array, 3, 8, 8);
?>
```

Перший параметр функції create_table2() як і раніше – обов'язковий. Наступні три параметри – не обов'язкові, оскільки для них визначені значення по замовчуванню. Необов'язкові параметри можна передавати не всі, а тільки деякі з них. Параметри присвоюються зліва-направо.

Слід пам'ятати, що пропуск необов'язкового параметру і задавання наступного після нього параметру не допускається. У наведеному прикладі це означає, що за необхідності задати $cellpadding пропустити $border не можна.

Існує можливість дізнатися, скільки та які саме параметри передано.

```
<?php
function create_table2($data, $border=1, $cellpadding=4, $cellspacing=4){
echo "<table border=$border cellpadding=$cellpadding cellspacing=$cellspacing>\n";
reset($data); // вказуємо початок
$value=current($data);
while($value){
        echo "<tr><td>$value</td></tr>\n";
        $value = next($data);
}
echo '</table>';
echo 'Кількість параметрів: ';
echo func_num_args().'<br/>';
$args = func_get_args();
foreach ($args as $arg)
        echo $arg.'<br/>';
}
```

```
$my_array = array('Рядок 1', 'Рядок 2', 'Рядок 3');
create_table2($my_array, 3, 8, 8);
?>
```

Функція func_num_args() визначає, скільки аргументів було передано функції користувача, а func_get_args() повертає масив, який містить ці аргументи.

Звичний спосіб передачі параметрів називається передачею за значенням. При цьому зміна значення глобального параметра функції з тіла функції неможлива. Для вирішення проблеми використовується так звана передача за посиланням, яка реалізується шляхом додавання символу амперсанда (&) перед іменем параметра у визначенні функції.

Ключове слово return завершує виконання функції і передає виконання наступному операторові після виклику функції. Також функція завершує свою роботу, коли усі оператори виконані.

Найпоширеніша причина використання оператора return з метою передчасного завершення функції – умова виникнення помилки. Також return використовують, коли функція повинна повертати результат.

PHP підтримує рекурсивні функції. Рекурсивною функцією називають функцію, котра викликає сама себе. Такі функції особливо корисні для переміщення по динамічних структурах даних, таких як зв’язані списки та дерева.

Слід пам’ятати, що рекурсивна функція створює у пам’яті копії самої себе і, відповідно, веде до непродуктивних витрат ресурсів.

Детальнішу інформацію шукайте на сайті https://www.php.net чи будь-якому іншому.

Додаткові теоретичні відомості:

Масиви

Функції

3.2 Хід роботи

1. Ознайомтеся із теоретичними відомостями і наведеними в них прикладами. Код функції create_table2 скопіюйте в файл function.php, а виклик функції здійсніть з файлу labN.php.

2. Значення масу із 10 елементів згенеруйте випадковим чином з діапазону він 1 до a+10 (функція mt_rand), виведіть на веб-сторінку індекси і відповідні значення згенерованого масиву, а також індекси та значення мінімального, максимального та середнє значення елементів.

3. У масив внесіть 5 довільних чисел. Далі використовуючи конструкцію foreach виведіть їх і їх квадрати у вигляді: 4^2=16 2^2=4 5^2=25 і т.д.

4. Виконайте три завдання: одне завдання згідно номеру варіанту і два наступних (тобто, якщо номер в списку групи 3, то потрібно виконати завдання 3, 4, 5), дані зберігайте у масивах, значення елементів виводьте на вебсторінку:

№	Завдання
1	Знайти та роздрукувати цілі числа, кратні трьом, та їх кількість в діапазоні від 30 до 60.
2	Знайти методом підбору корінь рівняння 4*x+5=25. Пошук почати з x=1, збільшувати кожне наступне значення x на 1.
3	Знайти та роздрукувати перші 8 чисел з ряду Фібоначчі та їх суму.
4	Знайти та роздрукувати усі парні числа ряду від 0 до 20.
5	Годинник б'є стільки разів, яка зараз година. Знайти та роздрукувати кількість ударів годинника від 00 год. до 15 год. включно.
6	Знайти та роздрукувати ряд Фібоначчі починаючи з 5-го члену ряду і закінчуючи 20-м. Підрахувати їх кількість.
7	Знайти та роздрукувати кожне п'яте ціле число в ряді -1 до 51.
8	Спортсмен у перший день подолав 10 км дистанції. Кожного наступного дня він пробігав на 10% більше, ніж попереднього. За скільки днів спортсмен подолає 50 км?
9	Знайти та роздрукувати суму всіх парних та добуток непарних цілих чисел в діапазоні від 0 до 20.
10	Знайти та вивести на екран у вигляді таблиці значення функції y=2*x при x=[0..4] з кроком 0.5.
11	Вивести на екран лінію, яка буде складатися з 20 символів «*», замінюючи кожен третій символ на знак «?».
12	Знайти та вивести на екран у вигляді таблиці значення квадратних коренів цілих чисел в діапазоні [1..10]
13	Знайти та роздрукувати середнє арифметичне квадратів усіх цілих чисел від a до b. Прийняти a=3, b=10.

14	Дано натуральне число N=10. Знайти суму та кількість усіх членів ряду S=1+1/2+1/3+...+1/N.
15	Вивести в таблицю квадрати й куби таких чисел: 1, 3, 5, 9, 17.
16	Одноклітинна амеба кожні 3 години ділиться на 2 клітини, вивести в таблицю, скільки буде амеб після 3, 6, 9...24 год.
17	Щомісячна стипендія студента складає 1500 грн., а загальні витрати на місяць 2000 грн., визначте, яку суму студент візьме в борг за 10 міс., якщо загальні витрати зростають на 2% кожного місяця.
18	Знайти та вивести на екран суму та кількість елементів ряду y=1*3*5*...* (2*n-1) при n=20.
19	Знайти та вивести таблицю зведення числа 2 до ступеню 1..10.
20	Знайти та вивести на екран суму натуральних чисел, кратних 5, в діапазоні від 500 до 1000.
21	Дано натуральне число N=200. Одержати найбільше число $4^n<N$, де n - натуральне число.
22	Кожного дня спортсмен пробігає 5 км. Який шлях пробіжить спортсмен за 30 днів, якщо кожного наступного дня він пробігає на 0, 5 км більше за попередній.
23	Вивести на екран таблицю 10 рядків на 10 стовпчиків (усього 100 комірок), і в кожній комірці вивести наступне число (від 0 до 99).
24	Яка сума накопичиться на депозитному рахунку через 5 років, якщо вносити на рахунок кожен місяць на 20% більше, ніж було внесено попереднього місяця без урахування капіталізованих відсотків. Відсотки по внеску складають 12% річних, кожного місяця відсотки капіталізуються. Початкова сума - 200 грн.
25	Знайти суму та кількість парних чисел в діапазоні від 50 до 100.
26	Визначте число n, при якому сума квадратів натурального ряду 1..n буде менша за встановлене число N, де N=50.
27	Знайти та роздрукувати в стовпчик усі непарні числа ряду від 1 до 100.
28	Знайти та роздрукувати в таблиці цілі числа, кратні п'яти, та їх кількість в діапазоні від 100 до 300.
29	Дано натуральне число n=10. Знайти кількість натуральних чисел, більше n та менше 100, які не діляться на 2 та на 3. Виведіть ці числа.
30	Знайти і вивести на екран таблицю квадратів перших п'яти цілих додатних непарних чисел.

5. Реалізуйте наступні функції (функції реалізовуйте в файлі function.php, а необхідні масиви і виклики функції задайте у файлах розміщених в папці labN):

5.1. Функція, що виводить заданий масив чисел разом із індексами в заданому і оберненому порядку. Задану послідовність вказайте при виклику функції.

5.2. Для двовимірного масиву NxN, де N=(a%10+1)*2, значення згенеруйте випадковим чином з діапазону від 1 до 100. Створіть функцію, що виводить переданий в неї двовимірний масив у вигляді таблиці і два одновимірні масиви: елементами першого є мінімальні значення рядків вхідного масиву, а другого — числа, які знаходяться в останньому стовпці.

6. Створіть файл task6.php, в якому розмістіть форму, яка містить однорядкове текстове поле для введення числа із підписом «Введіть натуральне число » і кнопку для передачі даних. При умові успішної передачі даних (перевірка на те чи введено і передано натуральне число), викликати функцію, розміщену у файлі function.php, яка приймає введене у текстове поле число N, формує і виводить масив, елементами якого є перші N натуральних чисел, піднесених до квадрату і відповідні індекси.

Приклад виклику функції при кліку на кнопку форми:

У файл task6.php записуєте:

```
<?php
require("../config.php");
include_once("function.php");
?>
<html>
<H2>PHP. Робота з функціями</H2>
<?php
$variable=$_GET['formvariable'];
if ($variable==2){
echo_fun();
}
// інший варіант: if ($_GET['formvariable']<>""), умова залежить від поставленого завдання
?>
```

```
<form action="task6.php" method="get">
<input type="text" name="formvariable">
<input type="submit" value="Додати">
</form>
</html>
```

У файлі function.php:

```
function echo_fun(){
echo "<br>text from function echo_fun<br>";
}
```

7. Створити файл task7.php. Даний файл підключіть в кінець файлу task6.php з допомогою include_once. У файл task7.php у блок PHP додайте посилання на файл task7.php (тобто посилання на самого себе). У посиланні передайте число згенероване випадковим чином від 1 до 6. З допомогою оператора switch відповідно до переданого посиланням числа виведіть повідомлення «Викликати функцію func1», якщо передане значення 1, func2 при 2, func3 при 3, якщо не спрацює жоден із варіантів, вивести «Некоректні дані».

8. Здайте і захистіть лабораторну роботу.

3.3. Контрольні запитання

1. Яким чином створюють масиви. Ініціалізація елементів масивів.
2. Яким чином організувати доступ до елементів масиву за допомогою циклів на PHP.
3. Асоціативні масиви. Призначення. Доступ до елементів.
4. Сортування елементів масивів.
5. Призначення операторів require() та include().
6. Яким чином створюють функції на PHP?

Лабораторна робота №4. Асоціативні масиви в PHP. Опрацювання параметрів форми

Тривалість: 2 акад. години

Мета: ознайомитися з прийомами застосування масивів при розв’язуванні задач та опрацюванні параметрів форми.

Завдання: ознайомитися з описом масивів та функціями опрацювання масивів у PHP та виконати всі пункти згідно ходу роботи.

4.1 Теоретичні відомості

4.1.1 Асоціативні масиви в PHP

В PHP індексом масиву може бути не тільки число, а й рядок. Причому на такий рядок не накладаються ніякі обмежень: він може містити пробіли, довжина такого рядка може бути будь-яка.

Асоціативні масиви особливо зручні в ситуаціях, коли елементи масиву зручніше пов'язувати зі словами, а не з числами.

Отже, масиви, індексами яких є рядки, називаються асоціативними масивами.

У наступному прикладу символьні назви слугують як ключі, а числа – як значення.

```
<?php
$arr=array('the_first'=>1, 'the_second'=>2, 'the_third'=>3);
echo $arr['the_first'].'<br/>';
echo $arr['the_second'].'<br/>';
echo $arr['the_third'].'<br/>';
?>
```

Створити масив з багатьма елементами можна також, оголосивши масив з одним елементом і додаючи до нього пізніше решту потрібних елементів:

```
<?php
$arr=array('the_first'=>1);
$arr['the_second']=2;
$arr['the_third']=3;
```

```
foreach($arr as $current)
        echo $current.'<br/>';
?>
```

Оскільки в асоціативних масивах індекси не є числами, то для роботи з такими масивами неможливо скористатися лічильником у циклі for. У цьому випадкові слід застосувати цикл foreach або конструкції list() та each().

```
<?php
$arr=array('the_first'=>1, 'the_second'=>2, 'the_third'=>3);
foreach($arr as $current)
echo $current.'<br/>';
?>
```

Якщо необхідно отримати доступ до індексів масиву, використовують конструкцію:

```
<?php
$arr=array('the_first'=>1, 'the_second'=>2, 'the_third'=>3);
foreach($arr as $key => $value)
echo '$arr['.$key.']='.$value.'<br/>';
?>
```

Результат:

$arr[the_first]=1
$arr[the_second]=2
$arr[the_third]=3.

Наведений нижче код виводить вміст масиву за допомогою конструкції each():

```
<?php
$arr=array('the_first'=>1, 'the_second'=>2, 'the_third'=>3);
while($element=each($arr))
{
echo '$arr['.$element['key'].']='.$element['value'].'<br/>';
```

```
}
?>
```

У цьому прикладі змінна $element– це масив. При викликові функції each() вона повертає масив з чотирма значеннями і чотирма індексами. Комірки key та 0 містять ключ поточного елементу, value та 1 – його значення.

Важливо: при використанні функції each() слід пам’ятати, що масив відстежує поточний елемент. Якщо в одному і тому ж сценарії необхідно двічі скористатися значеннями одного і того ж масиву, то потрібно за допомогою функції reset() встановити поточний елемент на початок масиву.

4.1.2 Багатовимірні асоціативні масиви

Багатовимірні асоціативні масиви можуть містити кілька ключів, відповідних конкретному індексу асоціативного масиву. Розглянемо приклад багатовимірного асоціативного масиву:

```
<?php
//Багатовимірний масив
$A["Ivaniv"]=array("name"=>"Іванів І.І.", "age"=>"25",
"email"=>"ivaniv@pnu.edu.ua");
$A["Petriv"]=array("name"=>"Петрів П.П.", "age"=>"34",
"email"=>"petriv@pnu.edu.ua");
$A["Pavliv"]=array("name"=>"Павлів С.С.", "age"=>"47",
"email"=>"pavliv@pnu.edu.ua");
?>
```

Доступ до елементів багатовимірного асоціативного масиву здійснюється таким чином:

```
echo $A["Ivaniv"]["name"]; //Виводить Іванів І.І.
echo $A["Petriv"]["email"]; //Виводить petriv@pnu.edu.ua
```

Асоціативні багатовимірні масиви можна створювати і класичним способом, хоча це не так зручно:

```
<?php
//Багатовимірний асоціативний масив
$A["Ivaniv"]["name"]="Іванів І.І.";
```

```
$A["Ivaniv"]["age"]="25";
$A["Ivaniv"]["email"]="ivaniv@pnu.edu.ua";

$A["Petriv"]["name"]="Петрів П.П.";
$A["Petriv"]["age"]="34";
$A["Petriv"]["email"]="petriv@pnu.edu.ua";

$A["Pavliv"]["name"]="Павлів С.С.";
$A["Pavliv"]["age"]="47";
$A["Pavliv"]["email"]="pavliv@pnu.edu.ua";

//Одержуємо доступ до багатовимірного асоціативного масиву
Echo $A["Ivaniv"]["name"]."<br>"; //Виводить Іванів І.І.
Echo $A["Pavliv"]["age"]."<br>"; //Виводить 47
Echo $A["Petriv"]["email"]."<br>"; //Виводить petriv@pnu.edu.ua
?>
```

Видалити окремий елемент масиву можна за допомогою функції **unset(),** а перевірити існування масиву можна за допомогою функції **isset().** Визначимо масив з 10 елементів і знищимо кожен парний елемент.

```
<?php
$arr=array(9, 8, 7, 6, 5, 4, 3, 2, 1, 0);
unset($arr[0], $arr[2], $arr[4], $arr[6], $arr[8]);
//Перевіряємо чи існують елементи масиву
for($i=0 ; $i<10 ; $i++)
{
if(isset($arr[$i])) echo "Елемент $arr[$i] визначений <br/>" ;
else echo "Елемент $arr[$i] не визначений <br/>" ;
}
?>
```

Результатом роботи скрипта будуть наступні рядки:

```
елемент $arr[0] не визначений
елемент $arr[1] визначений
елемент $arr[2] не визначений
```

елемент $arr[3] визначений
елемент $arr[4] не визначений
елемент $arr[5] визначений
елемент $arr[6] не визначений
елемент $arr[7] визначений
елемент $arr[8] не визначений
елемент $arr[9] визначений

За допомогою функції **unset()** можна знищити весь масив відразу.

```
<?php
$arr=array(9, 8, 7, 6, 5, 4, 3, 2, 1, 0);
unset($arr);
if(isset($arr)) echo "Массив визначений" ;
else echo "Массив не визначений" ;
?>
```

До цього масиви виводилися за допомогою циклу, проте в PHP передбачена спеціальна функція для виведення вмісту масиву **print_r().** Функція орієнтована на вивід в консольний потік, тому при виведенні результатів у вікно браузера краще оформити її тегами <pre> і </pre>:

```
<?php
$arr[]="345";
$arr[]="mail@pnu.edu.ua";
$arr[]="http://www.softtime.ua";
$arr[]="login";
$arr[]="password";
echo "<pre>";
print_r($arr);
echo "</pre>";
?>
```

Результат роботи скрипта виглядає наступним чином:

```
Array
(
[0] => 345
[1] => mail@pnu.edu.ua
[2] => http://www.softtime.ua
[3] => login
[4] => password
)
```

4.1.3 Операції над масивами

4.1.3.1 Визначення числа елементів в масиві **count():**

Створимо масив $name:

```
<?php
$name=array('Boss', 'Lentin', 'NAV', 'Endless', 'Dragons', 'SiLeNT', 'Doctor', 'Lynx');
?>
```

Щоб визначити число елементів в масиві можна поступити наступним чином:

```
<?php
echo 'Число елементів в масиві - ' . count($name);
?>
```

Результат:

Число елементів в масиві – 8

4.1.3.2 Об'єднання масивів

а) Створимо два асоціативних масиву $a і $b:

```
<?php
$a=array(“a” => ”aa”, “b” => “bb”);
$b=array(“c” => ”cc”, “d” => “dd”);
?>
```

Нехай необхідно створити масив $c, які буде містити як елементи масиву $a так і масиву $b:

```
<?php
$a=array("a" => "aa", "x" => "xx");
$b=array("c" => "cc", "d" => "dd");
$c=$a + $b ;
echo "<pre>" ;
print_r($c);
echo "</pre>" ;
?>
```

Результат:

```
Array
(
[a] => aa
[x] => xx
[c] => cc
[d] => dd
)
```

Створимо два числових масиву $a і $b:

```
<?php
$a=array(10, 20);
$b=array(100, 200, 300, 400, 500);
?>
```

Їх вже не вийде об'єднати за допомогою конструкції $c=$a + $b ;. Для їх об'єднання буде потрібно скористатися функцією **array_merge():**

```
<?php
$c=array_merge($a, $b);
?>
```

4.1.3.3 Сортування масиву

Скористаємося масивом $name:

```
<?php
$name=array('Boss', 'Lentin', 'NAV', 'Endless', 'Dragons', 'SiLeNT', 'Doctor', 'Lynx');
?>
```

Нехай потрібно впорядкувати масив в алфавітному порядку, для цього можна скористатися наступним кодом:

```
<?php
sort($name);
echo "<pre>" ;
print_r($name);
echo "</pre>" ;
?>
```

Результат:

```
Array
(
[0] => Boss
[1] => Doctor
[2] => Dragons
[3] => Endless
[4] => Lentin
[5] => Lynx
[6] => NAV
[7] => SiLeNT
)
```

Нехай необхідно з масиву $name вибрати найкоротший елемент(у якого найменша кількість символів), в цьому випадку можна скористатися кодом:

```
<?php
$name=array('Boss', 'Lentin', 'NAV', 'Endless', 'Dragons', 'SiLeNT', 'Doctor',
'Lynx');
$min=strlen($name[0]);
$nam=$name[0];
for($i=1 ; $i<count($name); $i++)
{
$len=strlen($name[$i]);
if($len<$min)
{
$nam=$name[$i];
$min=strlen($nam);
} }
echo 'Найменша довжина - ' . $nam ;
?>
```

4.1.3.4 Переміщення всередині масиву

Створимо масив $num:

```
<?php
$num=array(1, 2, 3, 4, 5, 6, 7, 8, 9, 10);
?>
```

Нехай потрібно відобразити елементи масиву в зворотному порядку, в цьому випадку можна скористатися кодом:

```
<?php
$end=end($num);
While($end)
{
echo $end . ' - ' ;
$end=prev($num);
}
?>
```

Результат:

10 - 9 - 8 - 7 - 6 - 5 - 4 - 3 - 2 – 1

Наведений вище код можна модифікувати:

```
<?php
$num=range(1, 10);
print_r(array_reverse($num));
?>
```

Функція **range(1, 10)** створює масив(1, 2, 3, 4, 5, 6, 7, 8, 9, 10) з випадковим розташуванням елементів. Функція **array_reverse()** приймає масив і повертає елементи в зворотному порядку(10, 9, 8, 7, 6, 5, 4, 3, 2, 1)

Функція **reset()** повертає покажчик на перший елемент в масиві. Скористаємося масивом $num:

```
<?php
$num=array(1, 2, 3, 4, 5, 6, 7, 8, 9, 10);
?>
```

Нехай необхідно вивести всі елементи по порядку, і на останньому елементі масиву повернути покажчик на перший елемент масиву. Цю операцію можна здійснити за допомогою наступного коду:

```
<?php
$i=0 ; //Індекс 1 елемента
while($i<count($num)) {
echo $num[$i]. ' ' ;
$i++;
//Перевірка якщо $i рівний числу елементів в масиві
//тоді виводимо останній елемент і повертаємо вказівник
if($num[$i] == count($num))
{
echo $num[$i];
reset($num);
echo '<br/>' . "Кінець масиву" ;
exit();
} }
?>
```

Результат:
1 2 3 4 5 6 7 8 9 10

Кінець масиву

4.1.3.5 Перемішування елементів в масиві **shuffle()**

Функція **shuffle()** перемішує значення в масиві, і якщо масив асоціативний то повертає його як список:

```
<?php
$a=array(43, 'PHP', 4, 57, 'Boss', 90);
shuffle($a);
foreach($a as $n) echo "$n " ;
?>
```

4.1.3.6 Випадковий елемент масиву

Якщо є готовий масив, з якого необхідно вивести один випадковий елемент, для цього необов'язково перемішувати весь масив за допомогою функції **shuffle(),** досить згенерувати випадковий індекс масиву:

```
<?php
$arr=array(1, 2, 3, 4, 5, 6, 7, 8, 9, 10);
$index=rand(0, count($arr) - 1);
echo $arr[$index];
?>
```

4.1.3.7 Отримання частини масиву **array_slice()**

Створимо масив $a

```
<?php
$a=array('a', 'b', '3', '5', 'f');
?>
```

Отримати частину масиву можна за допомогою наступного коду:

```
<?php
```

```
$b=array_slice($a, 2) //вивід 3, 5, f
$b=array_slice($a, 0, 3) //a, b, 3
?>
```

4.1.3.8 Серіалізация масиву

Функції **serialize()** і **unserialize()** дозволяють здійснювати упакування і розпакування, відповідно, масивів і об'єктів.

Зауваження. Серіалізація вперше з'явилася в об'єктно-орієнтованих бібліотеках, (першої з яких була MFC), потім серіалізація стала з'являтися в об'єктно-орієнтованих мовах (Java). Ідея серіалізації полягає в тому, що об'єкти і масиви дуже складні за своєю структурою і на збереження їх шляхом перебору кожного елемента потрібен значний обсяг коду - найпростішим рішенням є збереження таких структур у вигляді єдиної закодованої послідовності - байт-коді. У PHP функції серіалізації упаковують дані не у вигляді байт-коду, а у вигляді рядка.

```
<?php
$poll[0]=23 ;
$poll[1]=45 ;
$poll[2]=34 ;
$poll[3]=2 ;
$poll[4]=12 ;
//Упаковуємо масив в рядок
$str=serialize($poll);
echo $str . "<br/>" ;
//Розпаковуємо масив із рядка
$arr=unserialize($str);
print_r($arr);
?>
```

Результат:

```
a:5:{i:0;i:23;i:1;i:45;i:2;i:34;i:3;i:2;i:4;i:12;}
```

```
Array
(
[0] => 23
[1] => 45
```

```
[2] => 34
[3] => 2
[4] => 12
)
```

Ви можете ознайомитися з усіма функціями для роботи з масивами у PHP в довіднику функцій.

Передавання параметрів форми

Спочатку напишемо HTML-документ, який буде містити практично всі елементи HTML-форми. Параметри форми будемо передавати скрипту для наступної обробки. Отже, лістинг HTML-документа send.html:

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html"
charset="windows-1251">
<title> Test </title>
</head>
<body>
<h3> Тестова форма </h3>
<form name="form1" method="post" action="script.php">
<p> <span> Текстове поле: </span>
<input type="text" name="textfield">
</p>
<p> Поле введення пароля:
<input type="password" name="pswfield">
</p>
<p> Приховане поле hidden
<input name="hidden" type="hidden" id="hidden"
value="Приховане_значення">
</p>
<hr size="1">
<p> Незалежні перемикачі(checkbox): </p>
<p>
<input type="checkbox" name="checkbox1" value="1">
Варіант перший
```

```
<input type="checkbox" name="checkbox2" value="1">
Варіант другий
<input type="checkbox" name="checkbox3" value="1" checked>
Варіант третій(за замовчуванням) </p>
<hr size="1">
<p> Залежні перемикачі(radio): </p>
<p>
<input name="radio_button" type="radio" value="yes">
Так
<input name="radio_button" type="radio" value="no">
Ні </p>
<hr size="1">
<p> Багаторядкове текстове поле(textarea): </p>
<p>
<textarea name="textarea" cols="40" rows="10"> Текст за умовчанням </textarea>
</p>
<hr size="1">
<p> Список з єдиним вибором: </p>
<p>
<select name=day_s size=1>
<option value=1> понеділок </option>
<option value=2> вівторок </option>
<option value=3 selected> середа </option>
<option value=4> четвер </option>
<option value=5> п'ятниця </option>
<option value=6> субота </option>
<option value=7> неділя </option>
</select>
</p>
<p> Список із множинним вибором(multiple): </p>
<p>
<select name=day_m[] size=7 multiple>
<option value=1 selected> понеділок </option>
<option value=2> вівторок </option>
<option value=3> середа </option>
<option value=4> четвер </option>
```

```
<option value=5> п'ятниця </option>
<option value=6> субота </option>
<option value=7> неділя </option>
</select>
</p>
<hr size="1">
<p>
<input type="submit" value="Надіслати форму">
<input type="reset" value="Очистити форму">
</p>
</form>
</body>
</html>
```

Коли користувач натискає кнопку "Надіслати форму", браузер передасть скрипту наступні параметри:

• textfield - значення текстового поля;

• pswfield - значення поля введення пароля;

• hidden - значення прихованого поля;

• параметри checkbox: checkbox1, checkbox2 і checkbox3 будуть передані тільки в тому випадку, якщо відповідні їм незалежні перемикачі активні;

• radio_button - значення групи radio(буде передано одне із значень: Yes або No);

• textarea - вміст багаторядкової текстової області;

• day_s - значення списку з єдиним вибором;

• day_m - значення списку із множинним вибором.

Тепер перед нами стоїть завдання обробки всіх параметрів переданої форми за допомогою PHP скрипта.

Параметри textfield, pswfield і textarea обробляються досить просто. Наприклад, для відображення значення параметра textfield достатньо написати в обробному скрипті: echo $_POST['textfield'];

З параметрами checkbox1, checkbox2, checkbox3, і radio_button справа дещо складніша. Якщо перемикач не активний, то перелічені параметри взагалі не будуть передані на сервер, ніби їх взагалі не було.

Отже, при спробі звернутися в скрипті до цих параметрів, ми отримаємо повідомлення, що змінна не існує. Тому просто написати echo

$_POST['checkbox1']; ми не можемо, нам необхідно спочатку перевірити існування цих параметрів в запиті.

Перевірка існування параметра здійснюється за допомогою функції isset(), яка служить для перевірки існування змінних.

```
if (isset($_POST['checkbox1'])) echo $_POST['checkbox1'];
if (isset($_POST['radio_button'])) echo $_POST['radio_button'];
```

Тільки після перевірки існування перерахованих параметрів форми можна починати роботу з змінними.

Складніше обробляти параметри списку із множинним вибором, оскільки в цьому випадку параметри передаються так:

```
day_m=01 & day_m=03 & day_m=07 ...
```

Ми опинилися в ситуації, коли один параметр має кілька значень. Це нагадує нам масив даних. В цьому випадку множинний список можна представити у вигляді масиву, а обробити його елементи за допомогою циклу foreach. Навіть не обов'язково знати кількість елементів множини списку. Потрібно лише попередньо дати зрозуміти транслятору PHP, що ми будемо передавати масив:

```
<select name="day_m[]" size=7 multiple>
```

Квадратні дужки[] - це ознака масиву. Циклічна обробка масиву здійснюється так:

```
foreach($_POST['day_m'] as $key => $value)
echo "$key=$value ";
```

А тепер наведемо остаточний лістинг PHP скрипта, обробника нашої тестової форми:

```
<? php
//Виводимо HTML-заголовки:
echo '<html>';
echo '<head>';
```

```
        echo '<meta http-equiv="Content-Type" content="text/html;
charset=windows-1251">';
        echo '<title> Test </title>';
        echo '</head>';
        echo '<body>';
        echo '<h3> Тестова форма </h3>';
        echo "<p> Передане значення текстового поля: <b>". $_POST['textfield'].
"</b> </p>";
        echo "<p> Передане значення поля пароля: <b>". $_POST['pswfield'].
"</b> </p>";
        echo "<p> Передане значення прихованого поля hidden: <b>".
$_POST['hidden']. "</b> </p>";
        echo '<hr size="1">';
        echo '<p> Були включені наступні незалежні перемикачі: </p>';
        if(isset($_POST['checkbox1'])) echo "<p> <b> Перший </b> </p>";
        if(isset($_POST['checkbox2'])) echo "<p> <b> Другий </b> </p>";
        if(isset($_POST['checkbox3'])) echo "<p> <b> Третій </b> </p>";
        echo '<hr size="1">';
        if(isset($_POST['radio_button']))
        {
        echo '<p> Був обраний незалежний перемикач з наступним значенням:';
        if($_POST['radio_button'] === "yes") echo "<b> Yes </b>";
        if($_POST['radio_button'] === "no") echo "<b> No </b>";
        echo '</p>';
        }
        else echo '<p> Жоден з незалежних перемикачів не був обраний </p>';
        echo '<hr size="1">';
        echo '<p> Значення багаторядкового текстового поля: </p>';
        echo "<p> <b>".$_POST['textarea']. "</b> </p>";
        echo '<hr size="1">';
        echo "<p> Значення списку з єдиним вибором: <b>". $_POST['day_s'].
"</b> </p>";
        echo '<hr size="1">';
        echo '<p> Значення списку із множинним вибором: </p>';
        foreach($_POST['day_m'] as $keys => $values)
         echo "<b> $values </b> <br>";
        echo '<hr size="1">';
```

```
echo '</body>';
echo '</html>';
?>
```

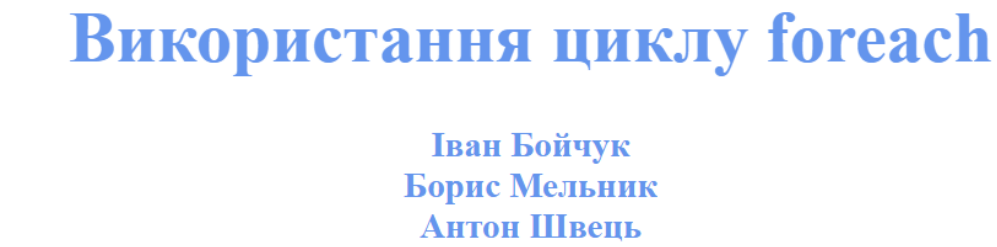
4.2 Хід роботи

1. Заповніть в циклах перший масив квадратами чисел від 10 до 20, а другий - кубами чисел від 1 до 10. Далі об'єднайте ці масиви і виведіть об'єднаний масив.

2. Розгляньте та реалізуйте використання циклу foreach для перебору та виведення елементів асоціативного масиву:

```
<html>
<body>
<h1> Використання циклу foreach </h1> <?php
$names["Бойчук"]="Іван";
$names["Мельник"]="Борис";
$names["Швець"]="Антон";
foreach ($names as $key => $value) {
echo "<b>$value $key</b><br>";}
?>
</body>
</html>
```

Приклад виводу:



4. В окремому файлі напишіть код створення масиву $my_topic (згідно теми із Лабораторної роботи 2) використовуючи конструкцію

$ім'я_масиву=array (індекс1 => значення1, індекс2 => значення2, ...).

Індексація елементів масиву повинна починатися з цифри 2. Масив повинен містити назви чотирьох елементів згідно Вашої теми. Виведіть індекси та елементи масиву. Поміняйте місцями індекси елементів масиву та

їх значення. Організуйте вивід даних таким чином, щоб спочатку виводився індекс масиву, а потім член масиву.

Створіть і заповніть багатовимірний асоціативний масив, який містить наступні дані по трьох країнах: назва, столиця, населення (дані придумайте самостійно).

Приклад реалізації одного із елементів багатовимірний асоційований масив:

$country["Ukraine"]=array("name"=>"Україна", "capital"=>"Київ", "popul"=>"45");

Використовуючи конструкцію **foreach** (для багатовимірного масиву одну конструкцію **foreach** вкладену в іншу)**,** виведіть з масиву:

- таблицю, яка містить три стовпці. Заголовки стовпців: «Назва», «Столиця», «Населення, млн.ч.». Кожен із стовпців містить відповідні дані по заданих країнах; К-сть рядків залежить від кількості заданих країн.
- три речення, типу «Столиця України — Київ, населення —45 млн. ч.;
- ключі першого та другого рівнів і їх значення, попередньо посортувавши елементи масиву $country:

Ukraine:
name=>Україна
capital=>Київ
popul=>45;

- виведіть даний масив у вікно браузера, використовуючи функцію **print_r().**

5. Використовуючи асоціативні масиви виконайте завдання згідно варіанту.

№	Завдання
1	● У масиві з 10 елементів задана вага спортсменок двох різних команди (в кілограмах). Знайти кількість спортсменок в кожній команді, чия вага перевищує 50 кг, але не більше 57 кг. Якщо таких спортсменок немає - видати повідомлення. ● Визначте масив, який буде зберігати інформацію про три населених пункти (одне з них Ваше місце народження), три вулиці в них та кількість мешканців на кожній вулиці. Виведіть на екран дану

	інформацію у вигляді таблиці. Інформацію про вулицю з мінімальною кількістю мешканців позначте червоним кольором. Визначте і виведіть кількість мешканців у кожному населеному пункті.
2	● Сформуйте масив з трьох речень. Поміняйте місцями перше і третє речення місцями. Виведіть на екран вихідний та змінений масив. ● Визначте масив, який буде зберігати інформацію про три області (одна з них Ваше місце народження), три міста в них та кількість мешканців на кожному місті. Виведіть на екран дану інформацію у вигляді таблиці. Інформацію про місто з максимальною кількістю мешканців позначте синім кольором. Визначте і виведіть кількість мешканців у кожній області.
3	● Визначте масив, який буде зберігати інформацію про 5 десертів з різною вартістю та кількістю калорій. Дайте відповідь на питання, чи є найдорожчий десерт самим калорійним? ● Визначте масив, який буде зберігати інформацію про три населених пункти (одне з них Ваше місце народження), три вулиці в них та кількість мешканців на кожній вулиці. Виведіть на екран дану інформацію у вигляді таблиці. Інформацію про вулицю з максимальною кількістю мешканців позначте зеленим кольором. Визначте і виведіть кількість мешканців у кожному населеному пункті.
4	● Визначте масив, що містить інформацію про авторів (прізвище, ім'я) та кількість книг, яку кожен опублікував. Результат виведіть у вигляді таблиці. Дайте відповідь на питання, скільки авторів опублікувало більше двох книг. ● Визначте масив, який буде зберігати інформацію про три області (одна з них Ваше місце народження), три міста в них та кількість мешканців на кожному місті. Виведіть на екран дану інформацію у вигляді таблиці. Інформацію про місто з мінімальною кількістю мешканців позначте синім кольором. Визначте і виведіть кількість мешканців у кожній області.
5	● У масиві з 10 елементів задана вага спортсменок двох різних команди (в кілограмах). Знайти кількість спортсменок в кожній команді, чия вага перевищує 50 кг, але не більше 57 кг. Якщо таких спортсменок немає - видати повідомлення. ● Визначте масив, що містить інформацію про трьох студентів (прізвище) та бали з 4-х предметів, які кожен вивчив. Результат виведіть

	у вигляді таблиці. Інформацію про студента (-тів) з найвищою оцінкою за предмет позначте синім кольором. Визначте і виведіть середній бал кожного студента.
6	● У масиві з 10 елементів задана вага спортсменок двох різних команди (в кілограмах). Знайти кількість спортсменок в кожній команді, чия вага перевищує 50 кг, але не більше 57 кг. Якщо таких спортсменок немає - видати повідомлення. ● Визначте масив, який буде зберігати інформацію про три населених пункти (одне з них Ваше місце народження), три вулиці в них та кількість мешканців на кожній вулиці. Виведіть на екран дану інформацію у вигляді таблиці. Інформацію про вулицю з максимальною кількістю мешканців позначте зеленим кольором. Визначте і виведіть кількість мешканців у кожному населеному пункті.
7	● Сформуйте масив з трьох речень. Поміняйте місцями перше і третє речення місцями. Виведіть на екран вихідний та змінений масив. ● Визначте масив, що містить інформацію про трьох студентів (прізвище) та бали з 4-х предметів, які кожен вивчив. Результат виведіть у вигляді таблиці. Інформацію про студента (-тів) з найнижчою оцінкою за предмет позначте червоним кольором. Визначте і виведіть сумарний бал кожного студента. .
8	● Визначте масив, який буде зберігати інформацію про 5 десертів з різною вартістю та кількістю калорій. Дайте відповідь на питання, чи є найдорожчий десерт самим калорійним? ● Визначте масив, який буде зберігати інформацію про три області (одна з них Ваше місце народження), три міста в них та кількість мешканців у кожному місті. Виведіть на екран дану інформацію у вигляді таблиці. Інформацію про місто з максимальною кількістю мешканців позначте синім кольором. Визначте і виведіть кількість мешканців у кожній області.
9	● Визначте масив, що містить інформацію про авторів (прізвище, ім'я) та кількість книг, яку кожен опублікував. Результат виведіть у вигляді таблиці. Дайте відповідь на питання, скільки авторів опублікувало більше двох книг. ● Визначте масив, який буде зберігати інформацію про три населених пункти (одне з них Ваше місце народження), три вулиці в них та кількість мешканців на кожній вулиці. Виведіть на екран дану

	інформацію у вигляді таблиці. Інформацію про вулицю з максимальною кількістю мешканців позначте фіолетовим кольором. Визначте і виведіть кількість мешканців у кожному населеному пункті.
10	● Визначте масив, що містить інформацію про студентів (прізвище, ім'я) та бали з 5 предметів, які кожен вивчив. Результат виведіть у вигляді таблиці. Визначте середній бал кожного студента. ● Визначте масив, що містить інформацію про трьох студентів (прізвище) та бали з 4-х предметів, які кожен вивчив. Результат виведіть у вигляді таблиці. Інформацію про студента (-тів) з найнижчою оцінкою за предмет позначте зеленим кольором. Визначте і виведіть сумарний бал кожного студента.

6. У HTML формі користувач вводить у п’ять різних полів: прізвище, ім'я, e-mail, пароль і повторити пароль. Після натискання кнопки ***“Готово”*** запускається PHP-скрипт, який вносить ці дані в асоціативний масив і далі виводить їх в таблицю, використовуючи конструкцію **foreach**. Перед виводом передбачити перевірку заповнення всіх полів і співпадіння паролів, якщо перевірку не пройдено, вивести відповідне повідомлення.

7. Створити анкету, яка містить чотири запитання, кожне з яких має різні види варіантів відповіді: радіокнопки, перемикачі, списки єдиного і множинного вибору. Створити скрипт, який перевіряє наявність відповіді на кожне із запитань і виводить текст запитання і вибрані до них відповіді, у випадку відсутності відповіді виводить в вікно браузера відповідне повідомлення.

8. Здайте і захистіть лабораторну роботу.

4.3 Контрольні запитання

1. Наведіть приклад одного з можливих варіантів заповнення масиву значеннями у PHP.

2. Наведіть приклад одного з можливих варіантів заповнення асоційованого масиву значеннями у PHP.

3. Поясніть схему роботи оператора foreach у PHP.

4. Поясніть механізм передачі значень за посиланнями.

5. Наведіть приклади функцій, які можуть бути використані для опрацювання масивів у PHP.

Лабораторна робота №5. Робота з файлами. Робота з текстом

Тривалість: 2 акад. години

Мета: набути практичних навичок для роботи з файлами і текстом.

Завдання: дослідити можливості PHP для роботи з файлами і текстом і виконати завдання відповідно до ходу роботи.

5.1 Теоретичні відомості

5.1.1 Робота з файлами

5.1.1.1 Включення зовнішніх файлів

Робота з файлами - важливий інструмент PHP.

У кожен PHP-документ можна включити файл за допомогою спеціальної конструкції мови include . Її аргумент: шлях до файлу. Цією конструкцією зручно користуватися при наявності однакових шматків коду в багатьох PHP-програмах. Вміст файлу, що включається, обробляється як простий HTML-текст. Для того, щоб вміст цього файлу оброблявся як PHP-програма, потрібно оточувати його відкриваючими і закриваючими тегами PHP.

Приклад 5.1

```
<html>
<head>
<title>Використання include</title>
 </head>
 <body>
<?php
 include 'top.php';
echo "<H2>  ...Основна частина...</H2>";
?>
 </body>
</html>
```

Вміст файлу top.php з PHP-кодом:

```
echo “<H1>  ... Загальне  вітання ... </H1>”;
```

Результат прикладу 5.1:

... Загальне вітання ...
...Основна частина...

Якщо файл top.php містить тільки рядок HTML-тексту, результат буде тим же:

<H1>... Загальне вітання ... </H1>

Файли, що включаються, можуть повертати значення подібно до функцій. Використання оператора return перериває виконання цього файлу так само, як і функції.

Приклад 5.2

```
<html>
<head>
<title> Використання include, що повертає значення </title>
</head>
<body>
<?php
$res =  include 'top.php';
echo "<H2> Включений файл повернув $res</H2>";
?>
</body>
</html>
```

Вміст включеного файлу top.php з PHP-кодом:

```
<?php $a = 7 * 8; return $a; ?>
```

Результат прикладу 5.2:
Включений файл повернув 56

Інструкцію include можна використовувати всередині циклу. У циклі include виконується при кожній ітерації. Це можна використовувати для включення декількох файлів, наприклад:

for($i=1; $i<=5; $i++) include "incfile{$i}.html"; /*тут використано подвійні лапки замість одинарних, щоб замість змінної $i підставилися її значення, приклад назви файлу incfile1.html, incfile2.html і так до 5*/

Визначення імені файлу, який включається і його завантаження виконуються повторно при кожному виклику include. Це означає, що якщо вміст файлу, що включається, з моменту попереднього виклику змінився, то завантажиться новий зміст.

Конструкцію include також можна включати в тіло умовного оператора.

Конструкція include не є функцією, а є спеціальною конструкцією мови.

Для вказівки того, що файл потрібно підключати тільки один раз використовується оператор include_once

Дія констукції require аналогічна include, крім того, що у разі виникнення помилки вона видасть фатальну помилку рівня E_COMPILE_ERROR. Іншими словами, вона зупинить виконання скрипту, тоді як include тільки видасть попередження E_WARNING, яке дозволить продовження виконання скрипту.

5.1.1.2 Аналіз файлів

PHP містить множину функцій, що надають інформацію про файли. Детальнішу інформацію про наведені нижче та інші функції для роботи із файлами шукайте тут. Найбільш вживаними є:

file_exists() - визначає існування файлу, наприклад:

```
if(!file_exists( "aaa.php"))
  echo "Увага! Файл aaa.php не знайдений!";
```

is_file() - визначає, чи є досліджуваний об'єкт файлом, наприклад:

```
if(is_file( "bbb.txt"))
  echo "Поза всяким сумнівом, bbb.txt - це файл";
```

is_dir() - визначає, чи є досліджуваний об'єкт каталогом, наприклад:

```
if(is_dir( "/tmp"))
  echo "Дійсно, /tmp - це каталог";
```

is_readable() - визначає, чи доступний файл для читання, наприклад:

```
if(is_readable( "db.dbf"))
```

```
echo "db.dbf можна читати";
```

is_writable() - визначає, чи доступний файл для запису, наприклад:
if(is_writable("db.dbf")) echo "В db.dbf писати можна";
filesize() - визначає розмір файлу в байтах.
filemtime() - визначає дату і час останньої зміни файлу.
fileatime() - визначає дату і час останнього звернення до файлу.

Час повертається в форматі Unix, тобто являє собою кількість секунд, що пройшли після 1 січня 1970. У прикладі 5.3 це число перетворюється в зрозумілий для людини формат за допомогою функції date().

Приклад 5.3

```
 <html>
<head>
<title> Інформація про файл </title>
</head>
 <body>
 <?php $file = "top.php";
infoFile( $file );
function infoFile( $f ){
 if( !file_exists( $f ) ) { echo "$f не знайдений!"; return; }
 echo "$f - ".( is_file( $f ) ? "" : "не " )."файл<br>";
echo "$f - ".( is_dir( $f ) ? "" : "не " )."каталог<br>";
echo "$f ".( is_readable( $f ) ? "" : "не " )."доступний для читання<br>";
echo "$f ".( is_writable( $f ) ? "" : "не " )."доступний для запису<br>";
echo "размір $f в байтах - ".( filesize( $f ) )."<br>";
echo "остання зміна $f - ".( date( "d MYH:i", filemtime( $f ) ) )."<br>";
echo "останнє звернення до $f - ".( date( "d MYH:i", fileatime( $f ) )
)."<br>"; }
?>
</body>
</html>
```

Результат прикладу 5.3:

top.php - файл
top.php - не каталог
top.php доступний для читання
top.php доступний для запису
размір top.php в байтах - 32
остання зміна top.php - 16 Feb202315:34
останнє звернення до top.php - 16 Feb202315:34

5.1.1.3 Управління файлами

PHP містить множину функцій управління файлами. Найбільш вживаними є:

touch() - створює порожній файл з заданим ім'ям. Якщо такий файл вже існує, то функція змінить дату модифікації, наприклад:

```
touch( "ex1.txt");
```

copy() - копіює файл.

Опис ¶

copy(string $from, string $to, ?resource $context = null): bool

Копіює файл $from у файл з ім'ям $to.

Якщо потрібно перейменувати файл, використовуйте функцію rename(). Приклад застосування функції copy:

```
<?php
$file = 'example.txt';
$newfile = 'example.txt.bak';
if (!copy($file, $newfile)) {
   echo "не вдалося скопіювати $file...\n";
}
?>
```

unlink() - видаляє заданий файл, наприклад:

```
<?php if(unlink('filename.txt')) { echo "Файл видалений"; } else { echo "Помилка видалення файлу"; } ?>
```

fopen() - відкриває локальний або віддалений файл і повертає покажчик на нього. Покажчик використовується в усіх операціях з вмістом файла. Аргументи: ім'я файлу і режим відкриття.

r читання. Покажчик файлу встановлюється на його початок

r+ читання і запис. Покажчик файлу встановлюється на його початок

w запис. Покажчик файлу встановлюється на його початок. Весь старий вміст файлу втрачається. Якщо файл з вказаним ім'ям не існує, функція намагається його створити

w+ читання і запис. Покажчик файлу встановлюється на його початок. Весь старий вміст файлу втрачається. Якщо файл з вказаним ім'ям не існує, функція намагається його створити

a запис. Покажчик файлу встановлюється в його кінець. Якщо файл з вказаним ім'ям не існує, функція намагається його створити

a+ читання і запис. Покажчик файлу встановлюється в його кінець. Якщо файл з вказаним ім'ям не існує, функція намагається його створити

Наприклад:

$fp = fopen("http://www.php.net/", "r"); // для читання

$fp = fopen("ex1.txt", "w"); // для запису, покажчик файлу встановлюється на його початок

$fp = fopen("ex2.txt", "a"); // для запису, покажчик файлу встановлюється на його початок

Якщо відкрити файл не вдалося, то можна перервати виконання програми, наприклад:

$ fp = fopen("ex1.txt", "w") or die("Не вдалося відкрити файл");

fclose() - закриває файл. Аргумент: покажчик файлу, отриманий раніше від функції fopen(). Наприклад: fclose($fp);

feof() - перевірка кінця файлу. Аргумент: покажчик файлу.

fgetc() - читання чергового символу з файлу. Аргумент: покажчик файлу.

fgets() - читання чергового рядка файлу. Аргументи: покажчик файлу і зчитування довжина рядка. Операція припиняється або після зчитування зазначеної кількості символів, або після виявлення кінця рядка або файлу.

Приклад 5.4

```
<html>
<head>
<title> Читання рядків з файлу </title>
</head>
<body>
<?php $fp = fopen( "ex1.txt", "r" ) or die( "Не вдалося відкрити файл!" );
while( ! feof( $fp ) ) echo( fgets( $fp, 1024 ) )."<br>";
?>
</body>
</html>
```

fread() - загальна функція читання з файлу. Аргументи: покажчик файлу і кількість зчитувальних символів.

fseek() - відступ від початку файлу. Аргументи: покажчик файлу і зсув.

Приклад 5.5

```
<html>
<head>
<title> Виведення на екран другої половини файлу</title>
</head>
<body>
<?php $f = "ex1.txt"; $fp = fopen( $f, "r" ) or die( "Не вдалося відкрити $f" );
$fsize = filesize( $f );
$half =(int)( $fsize / 2 );
fseek( $fp, $half );
echo( fread( $fp,($fsize - $half) ) );
?>
</body>
</html>
```

fputs() - запис рядка в файл. Аргументи: покажчик файлу і рядок.

fwrite() - повний аналог функції fputs().

Приклад 5.6

```
<html>
<head>
<title> Запис і додавання в файл </title>
</head>
<body>
<?php $fp = fopen( "ex2.txt", "w" ) or die( " Не вдалося відкрити файл " );
fputs( $fp, " Запис в файл \n" );
fclose( $fp );
$fp = fopen( "ex2.txt", "a" ) or die( " Не вдалося відкрити файл " );
fputs( $fp, " Додавання в кінець файлу " ); fclose( $fp ); ?> </body>
</html>
```

flock() - блокує файл, тобто не дозволяє іншим користувачам читати цей файл або писати в нього, поки той, хто наклав блокування не завершить роботу з даним файлом.

Аргументи: покажчик файлу і номер режиму блокування.

1 Можна читати, не можна писати

2 Не можна ні читати, ні писати

3 Розблокування

Приклад 5.7

```
<html>
<head>
<title> Блокування файлу </title>
</head>
<body>
<?php $fp = fopen( "ex1.txt", "a" ) or die( " Не вдалося відкрити файл " );
flock( $fp, 2 ); // Повне блокування
fputs( $fp, " Запис в файл \n" );
flock( $fp, 3 ); // Розблокування
fclose( $fp );
?>
</body>
</html>
```

Блокування за допомогою flock() не є абсолютним. Нею будуть блокуватися лише ті програми, які теж використовують цю функцію.

file_get_contents() - прочитати весь файл або URL

file_put_contents() - записати у файл

Функція file() читає файл і повертає його вміст у вигляді масиву. Кожен елемент масиву відповідає одному рядку.

```
<?php
$arr = file( "text.txt" );
for( $i = 0 ; $i < count( $arr ); $i++)
{
echo $arr [ $i ]. "<br />" ;
}
?>
```

5.1.1.4 Завантаження файлу на сервер

Для локального серверу:

У вашому "php.ini" повинні бути наступні три параметри:

File_uploads = On /* Включаємо підтримку завантаження.*/

Upload_tmp_dir = ПОЛНИЙ_ШДЯХ_ДО_ПАПКИ_ДЕ_БУДУТЬ_ЗБЕРІГАТИСЯ_ЗАВАНТАЖЕНІ (Тимчасові) _ФАЙЛИ

Upload_tmp_dir Наприклад = D:/сервер/PHP/завантаження

Upload_max_filesize = 2M /* Максимальний розмір файлів (в нашому випадком 2 МБ).*/

Приклад 5.8

Реалізація завантаження файлу, назва файлу upload.php, у поточній папці має бути створеним каталог files для збереження завантажених файлів:

```
<?php
if(!empty($_GET["subdir"])) {echo "<p>Ім'я переданої змінної ".$_GET["subdir"];} else {echo "<p>zminna subdir not found</p>";}
$subdir=$_GET["subdir"];
echo "subdir=$subdir";
$dir = "";
```

```
$result = move_uploaded_file($_FILES[ufile][tmp_name], $dir . $_FILES[ufile][name]);
echo
"<div align='center'>
<form action='upload.php?subdir=$subdir' method='post' enctype='multipart/form-data'>
<input type='file' name='uploadfile'>
<input type='submit' value='Завантажити'>
</form></div>
<hr/>";
 $uploaddir = "./files/$subdir/";
echo "<p>uploaddir=$uploaddir";
$uploadfile = $uploaddir.basename($_FILES['uploadfile']['name']);
echo "<p>uploadfile=$uploadfile";
if(copy($_FILES['uploadfile']['tmp_name'],$uploadfile))
{
echo "<p>Файл завантажений на сервер";
}
else {echo "<p>Файл не завантажений на сервер!";}
echo "<p>Оригінальна назва".$_FILES['uploadfile']['name']."</p>";
echo "<p>Тип файлу".$_FILES['uploadfile']['type']."</p>";
echo "<p>Розмір".$_FILES['uploadfile']['size']."</p>";
echo "<p>Тимчасове ім'я "  .$_FILES['uploadfile']['tmp_name']."</p>";
?>
```

Під час завантаження файлу в Unix-подібні операційні системи в кінці рядків можуть відображатися символи ^M. Для їх видалення в командному рядку записуємо:

```
cat name_file_in.php|tr –d '\r' > name_file_out.php
```

5.1.2 Функції роботи з рядками в PHP

Особливості операторів порівняння стосовно рядків.

```
$one = 1; // Присвоюємо число один.
$zero = 0; // Присвоюємо число нуль.
echo "1: ";
```

```
if($one == "") echo 1; // Очевидно, не дорівнює - не виводить 1.
echo "<br>2: ";
if($zero == "") echo 2; //* Увага! Всупереч очікуванням друкує 2!
echo "<br>3: ";
if( "" == $zero) echo 3; // * І це теж не допоможе - друкує!..
echo "<br>4: ";
if( "$zero" == "") echo 4; // Так правильно - не виводить 4.
echo "<br>5: ";
if(strval($zero) == "") echo 5; // Так теж правильно - не виводить 5.
echo "<br>6: ";
 if($zero === "") echo 6; // Кращий спосіб.
?>
```

Результати прикладу:

```
1:
2:2
3:3
4:
5:
6:
```

5.1.2.1 Функції роботи з рядками

chop()

Функція chop() повертає рядок після видалення з нього завершальних пропусків і символів нового рядка. Синтаксис функції chop():

string chop(string рядок)

У наступному прикладі функція chop() видаляє зайві символи нового рядка:

```
<?php
echo "<pre>"; //without <pre> you can't see desired output in your browser
echo "1: Ramki "; //right spaces aren't eliminated
echo chop("SRamkrKishnaTRRE", "A..Z");
echo chop("<br>2: Ramki "); //right spaces are eliminated
echo chop("SRamkriLshna", "a..z");
echo "</pre>";
?>
```

Результат:

```
1:   Ramki   SRamkrKishna
2:   RamkiSRamkriL
```

str_pad()

Функція str_pad() доповнює рядок до певної довжини заданими символами і повертає відформативаний рядок. Синтаксис функції str_pad():

string str_pad(string рядок, int довжина доповнення [, string доповнення [, int тип доповнення]])

Якщо необов'язковий параметр доповнення не вказано, рядок доповнюється пропусками. В іншому випадку рядок доповнюється заданими символами.

trim()

Функція trim() видаляє всі пропуски з обох країв рядка і повертає отриманий рядок. Синтаксис функції trim():

string trim(string рядок)

До числа пропусків, що видаляються відносяться і спеціальні символи \n, \r, \t, \v і \0.

ltrim()

Функція ltrim() видаляє всі пропуски і спеціальні символи з лівого краю рядка і повертає отриманий рядок. Синтаксис функції ltrim():

string ltrim(string рядок)

strlen()

Визначення довжини рядка. Синтаксис:

int strlen(string рядок)

Наступний приклад демонструє визначення довжини рядка функцією strlen():

```
$string = "hello";
$length = strlen($string);
// $length = 5
```

5.1.2.2 Порівняння двох рядків

Порівняння двох рядків належить до числа найважливіших рядкових операцій будь-якої мови. Хоча це завдання можна вирішити кількома різними способами, в PHP існують чотири функції порівняння рядків:

strcmp()

strcasecmp()

strspn()

strcspn()

strcmp()

Функція strcmp() порівнює два рядки з урахуванням регістру символів. Синтаксис функції strcmp(): int strcmp(string рядок1, string рядок2)

Після завершення порівняння strcmp() повертає одне з трьох можливих значень:

0, якщо рядок1 і рядок2 збігаються;

<0, якщо рядок1 менший, ніж рядок2;

> 0, якщо рядок2 менший, ніж рядок1.

У наступному фрагменті порівнюються два однакових рядки:

```
$string1 = "butter";
$string2 = "butter";
if((strcmp($string1, $string2) ) == 0):
print "Strings are equivalent!"; endif;
// Команда if повертає TRUE
```

strcasecmp()

Функція strcasecmp() працює точно так само, як strcmp(), за одним винятком - регістр символів при порівнянні не враховується. Синтаксис функції strcasecmp():

int strcasecmp(string рядок1, string рядок2)

У наступному фрагменті порівнюються два однакових рядки:

```
$string1 = "butter";
$string2 = "Butter";
if((strcmp($string1, $string2) ) == 0):
print "Strings are equivalent!";
endif;
```

```
// Команда if повертає TRUE
```

strspn()

Функція strspn() повертає довжину першого сегмента рядка1, що містить символи, присутні в рядку2. Синтаксис функції strspn():

int strspn(string рядок1, string рядок2)

Наступний фрагмент показує, як функція strspn() використовується для перевірки пароля:

```
$password = "123845";
 echo strspn($password, '1234567890'); // виведе 6
 if(strspn($password, "1234567890") == strlen($password) ):
 print "Password can not consist solely of numbers!";
 endif;
```

strcspn()

Функція strcspn() повертає довжину першого сегмента рядка1, що містить символи, відсутні в рядку2. Синтаксис функції strcspn():

int strcspn(string рядок1, string рядок2)

У наступному фрагменті функція strcspn() використовується для перевірки пароля:

```
 $password = "12345";
 if(strcspn($password, "1234567890") == 0):
 print "Password can not consist solely of numbers!";
 endif;
```

5.1.2.3 Обробка рядкових даних без застосування регулярних виразів

При обробці великих обсягів інформації функції регулярних виразів сильно уповільнюють виконання програми. Ці функції слід застосовувати лише при обробці щодо складних рядків, в яких регулярні вирази дійсно необхідні. Якщо ж аналіз тексту виконується по відносно простих правил, можна скористатися стандартними функціями PHP, які помітно прискорюють обробку. Всі ці функції описані нижче.

parse_str()

Функція parse_str() виділяє в рядку пари <змінна-значення> і присвоює значення змінних в поточній області видимості. Синтаксис функції parse_str():

parse_str(string рядок)

Функція parse_str() особливо зручна при обробці URL, що містять дані форм HTML або іншу розширену інформацію. У наступному прикладі аналізується інформація, передана через URL. Рядок являє собою стандартний спосіб передачі даних між сторінками або введених в форму HTML:

```
$url = "fname=wj&lname =gilmore&zip =43210";
parse_str($url);
// Після виконання parse_str() доступні такі змінні:
// $fname = "wj":
// $lname = "gilmore";
// $zip = "43210"
```

Оскільки ця функція створювалася для роботи з URL, вона ігнорує символ амперсанд(&).

explode()

Функція explode() ділить рядок на елементи і повертає ці елементи у вигляді масиву. Синтаксис функції explode():

array explode(string роздільник, string рядок [, int поріг])

Розбиття відбувається по кожному примірнику роздільника, причому кількість отриманих фрагментів може обмежуватися необов'язковим параметром поріг.

Поділ рядка функцією explode() продемонстровано в наступному прикладі:

```
$info = "wilson | baseball | indians";
$user = explode( "|", $info);
// $user [0] = "wilson";
// $user [1] = "baseball";
// $user [2] = "Indians";
```

implode()

Якщо функція explode() розділяє рядок на елементи масиву, то її двійник - функція implode() - об'єднує масив в рядок. Синтаксис функції implode():

string implode(string роздільник, array фрагменти)

Формування рядка з масиву продемонстровано в наступному прикладі:

```
$ohio_cities = array( "Columbus", "Youngstown", "Cleveland", "Cincinnati");
 $city_string = implode( "|", $ohio_cities);
 // $city_string = "Columbus | Youngstown | Cleveland | Cincinnati";
```

У implode() є псевдонім - функція join().

strpos()

Функція strpos() знаходить в рядку перший екземпляр заданого підрядка. Синтаксис функції strpos():

int strpos(string рядок, string підрядок [, int offset])

Необов'язковий параметр offset задає позицію, з якої повинен починатися пошук. Якщо підрядок не знайдено, strpos() повертає FALSE(0).

У наступному прикладі визначається позиція першого входження дати в файл журналу:

```
 $log = "
 206.169.23.11: /www/: 2000-08-10
 206.169.23.11: /www/logs/: 2000-02-04
 206.169.23.11: /www/img/: 1999-01-31 ";
 // В якій позиції в журналі вперше зустрічається 1999 рік?
 $pos = strpos($log, "1999");
 // $pos = 95, оскільки перший екземпляр "1999"
 // Знаходиться в позиції 95 рядка, що міститься в змінній $log
```

strrpos()

Функція strrpos() знаходить в рядку останній екземпляр заданого символу. Синтаксис функції strrpos():

int strpos(string рядок, char символ)

По можливостях ця функція поступається своєму двійнику - функції strpos(), оскільки вона дозволяє шукати тільки окремий символ, а не весь

рядок. Якщо у другому параметрі strrpos() передається рядок, при пошуку буде використаний лише її перший символ.

str_replace()

Функція str_replace() шукає в рядку всі входження заданого підрядка і замінює їх новим підрядком. Синтаксис функції str_replace():

string str_replace(string підрядок, string заміна, string рядок);

Функція substr_replace() дозволяє провести заміну в певній частині рядка. Нижче показано, як функція str_replace() використовується для проведення глобальної заміни в рядку.

Якщо підрядок жодного разу не зустрічається в рядку, вихідний рядок не змінюється:

```
$favorite_food = "My favorite foods are ice cream and chicken wings";
$favorite_food = str_replace( "chicken wings", "pizza", $favohte_food);
// $favorite_food = "My favorite foods are ice cream and pizza"
```

strstr()

Функція strstr() повертає частину рядка, що починається з першого входження заданого підрядка. Синтаксис функції strstr():

string strstr(string рядок, string підрядок)

У наступному прикладі функція strstr() використовується для виділення імені домену з URL:

```
$url = "http: //www.apress.com";
$domain - strstr($url, ".");
// $domain = ".apress.com"
```

substr()

Функція substr() повертає частину рядка, що починається із заданої початкової позиції і має задану довжину. Синтаксис функції substr():

string substr(string рядок, int початок [, int довжина])

Якщо необов'язковий параметр довжина не вказано, вважається, що підрядок починається із заданої початкової позиції і триває до кінця рядка. При використанні цієї функції необхідно враховувати чотири обставини:

➢ якщо параметр початок позитивний, підрядок повертається починаючи з позиції рядка із заданим номером;

➢ якщо параметр початок негативний, підрядок повертається починається з позиції рівній довжина рядка - початок;

➢ якщо параметр довжина позитивний, в повернутий підрядок включаються всі символи від позиції початок до позиції початок + довжина. Якщо остання величина перевищує довжину рядка, повертаються символи до кінця рядка;

➢ якщо параметр довжина негативний, повернутий підрядок закінчується на заданій відстані від кінця рядка.

Пам'ятайте про те, що параметр початок визначає зміщення від першого символу рядка; таким чином, в дійсності повертається рядок починаючи з символу з номером: початок+1.

Наступний приклад демонструє виділення частини рядка функцією substr():

```
$car = "1944 Ford";
$model = substr($car, 6);  // Smodel = "Ford"
```

Приклад з позитивним параметром довжина:

```
 $car = "1944 Ford";
 $model = substr($car, 0, 4);   // $model = "1944"
```

Приклад з негативним параметром довжина:

```
 $car = "1944 Ford";
 $model = substr($car, 2, -5);  // $model = "44"
```

substr_count()

Функція substr_count() повертає кількість входжень підрядка в заданий рядок.

Синтаксис функції substr_count(): int substr_count(string рядок, string підрядок). У наступному прикладі функція substr_count() підраховує кількість входжень підрядка ain:

```
 $tng_twist = "The rain falls mainly on the plains of Spain";
 $count = substr_count($tng_twist, "ain");
 // $count = 4
```

substr_replace()

Функція substr_replace() замінює частину рядка, яка починається із заданої позиції. Якщо заданий необов'язковий параметр «довжина», замінюється фрагмент заданої довжини; в іншому випадку проводиться заміна по всій довжині рядка. Синтаксис функції substr_replace():

string substr_replace(string рядок, string заміна, int початок [, int довжина])

Параметри початок і довжина задаються за певними правилами:

➢ якщо параметр початок позитивний, заміна починається із заданою позиції;

➢ якщо параметр початок негативний, заміна починається з позиції: довжина рядка - початок;

➢ якщо параметр довжина позитивний, замінюється фрагмент заданої довжини;

➢ якщо параметр довжина негативний, заміна завершується в позиції: довжина рядка - довжина.

Проста заміна тексту функцією substr_replace() продемонстрована в наступному прикладі:

```
$favs = " 's favorite links";
$name = "Alessia";
// Параметри "0, 0" означають, що замінний фрагмент починається
// і завершується в першій позиції рядка.
$favs - substr_replace($favs, $name, 0, 0);
print $favs:
```

Результат:

Alessia's favorite links

5.1.2.4 Перетворення рядків і файлів до формату HTML і навпаки

Перетворити рядок або цілий файл до формату, придатному для перегляду в web-браузері (або навпаки), простіше, ніж може здатися на перший погляд. У PHP для цього існують спеціальні функції.

Перетворення тексту в HTML

Швидке перетворення простого тексту до формату web-браузера - досить поширене завдання. В її рішенні вам допоможуть функції, описані в цьому розділі.

nl2br()

Функція nl2br() замінює всі символи нового рядка (\n) еквівалентними конструкціями HTML <br>.

Синтаксис функції nl2br():

string nl2br(string рядок)

Символи нового рядка можуть бути як видимими (тобто явно включеними в рядок), так і невидимими (наприклад, введеними в редакторі). У наступному прикладі текстовий рядок перетвориться в формат HTML за допомогою заміни символів \n розривами рядків:

```
// Текстовий рядок, що відображається в редакторі
$text_recipe = "
Party Sauce recipe:
1 can stewed tomatoes
3 tablespoons fresh lemon juice
Stir together, server cold. ";  // Перетворити символи нового рядка в
$html_recipe = nl2br($text_recipe)
```

При подальшому виведенні $html_recipe браузеру буде переданий наступний текст в форматі HTML:

```
Party Sauce recipe: <br>
1 can stewed tomatoes <br>
3 tablespoons fresh lemon juice <br>
Stir together, server cold. <br>
```

htmlspecialchars()

Функція htmlspecialchars() (детальніше див. тут) замінює деякі символи, які мають особливий сенс в контексті HTML, еквівалентними конструкціями HTML.

Синтаксис:

string htmlspecialchars(string рядок)

Здійснюються наступні перетворення

Символ	Заміна
&	&amp;
"	&quot;
'	&apos;
<	&lt;
>	&gt;

Зокрема, ця функція дозволяє запобігти введення користувачами розмітки HTML в інтерактивних web-додатках (наприклад, в електронних форумах). Помилки, допущені в розмітці HTML, можуть привести до того, що вся сторінка буде відображатися неправильно. Втім, у цього завдання існує і більш ефективне рішення - повністю видалити теги з рядка функцією strip_tags().

Наступний приклад демонструє видалення потенційно небезпечних символів функцією htmlspecialchars():

$user_input = "I just can not get <enough> of PHP & those fabulous cooking recipes!";

$conv_input = htmlspecialchars($user_input);

// $conv_input = "I just can not <enough> of PHP & those fabulous cooking recipes! "

Якщо функція htmlspecialchars() використовується в поєднанні з nl2br(), то останню слід викликати після htmlspecialchars(). В іншому випадку конструкції <br>, згенеровані при виклику nl2br(), перетворюються в видимі символи.

Перетворення HTML в простий текст

Іноді виникає необхідність перетворити файл у форматі HTML в простий текст. Функції, описані нижче, допоможуть вам у вирішенні цього завдання.

strip_tags()

Функція strip_tags() видаляє з рядка всі теги HTML і PHP, залишаючи в ній тільки текст. Синтаксис функції strip_tags():

string strip_tags(string рядок [, string дозволені теги])

Необов'язковий параметр “дозволені теги” дозволяє вказати теги, які повинні пропускатися в процесі видалення.

Нижче наведено приклад видалення з рядка всіх тегів HTML функцією strip_tags():

```
$user_input = "<b>I</b> just love PHP <i>and </i> gourmet recipes!";
$stripped_input = strip_tags($user_input);
// $stripped_input = "I just love PHP and gourmet recipes!";
```

У наступному прикладі видаляються не всі, а лише деякі теги:

```
$input = "I <b>love</b> to <a href = 'http: //www.eating.com' >eat</a>!";
echo "$input"; //I love to eat!
$strip_input = strip_tags($input, "<a>");
echo "$strip_input"; //I love to eat!
```

Видалення тегів з тексту також проводиться функцією fgetss().

5.1.2.5 Перетворення рядка до верхнього і нижнього регістру

У PHP існує чотири функції, призначених для зміни регістра рядка:

strtolower();
strtoupper();
ucfirst();
ucwords().

strtolower()

Функція strtolower() перетворює все алфавітні символи рядка до нижнього регістру. Синтаксис функції:

string strtolower(string рядок)

Неалфавітні символи функцією не змінюються. Перетворення рядка до нижнього регістра функцією strtolower() продемонстровано в наступному прикладі:

```
$sentence = "COOKING and PROGRAMMING PHP are my TWO favorites!";
$sentence = strtolower($sentence);
// Після виклику функції $sentence містить рядок
// "Cooking and programming php are my two favorites!"
```

strtoupper()

Рядки можна перетворювати не тільки до нижнього, а й до верхнього регістру. Перетворення виконується функцією strtoupper(), що має наступний синтаксис:

string strtoupper(string рядок)

Неалфавітні символи функцією не змінюються. Перетворення рядка до верхнього регістру функцією strtoupper() продемонстровано в наступному прикладі:

```
$sentence = "cooking and programming PHP are my two favorites!";
$sentence = strtoupper($sentence);
// Після виклику функції $sentence містить рядок
// "COOKING AND PROGRAMMING PHP ARE MY TWO FAVORITES!"
```

ucfirst()

Функція ucfirst() перетворює до верхнього регістру перший символ рядка - за умови, що він є алфавітним символом. Синтаксис функції ucfirst():

string ucfirst(string рядок)

Неалфавітні символи функцією не змінюються. Перетворення першого символу рядка функцією ucfirst() продемонстровано в наступному прикладі:

```
& Sentence = "cooking and programming PHP are my two favorites!";
$sentence = ucfirst($sentence);
// Після виклику функції $sentence містить рядок
// "Cooking and programming PHP are my two favorites!"
```

ucwords()

Функція ucwords() перетворює до верхнього регістру першу букву кожного слова в рядку. Синтаксис функції ucwords():

string ucwords(string рядок ")

Неалфавітні символи функцією не змінюються. "Слово" визначається як послідовність символів, відокремлена від інших елементів рядка пропусками. У наступному прикладі продемонстровано перетворення перших символів слів функцією ucwords():

```
$sentence = "cooking and programming PHP are my two favorites!";
$sentence = ucwords($sentence);
// Після виклику функції $sentence містить рядок
// "Cooking And Programming PHP Are My Two Favorites!"
```

strrchr()

strrchr("рядок", "о") - находить останнє входження підрядка

Якщо підрядок не знайдено, повертає FALSE.

На відміну від strchr(), якщо шукана стрічка складається більш ніж з одного символу, використовується тільки перший символ.

Якщо другий параметр не є рядком, він приводиться до цілого і трактується як код символу.

```
// Отримати останню директорію з $PATH
$dir = substr(strrchr($PATH, ": "), 1);
// Отримати все після останнього переведення рядка
$text = "Line 1 \n Line 2 \n Line 3";
$last = substr(strrchr($text, 10), 1);
```

addslashes()

Екранує спецсимволи в рядку. Повертає рядок, в якій перед кожним спецсимволи доданий зворотний слеш (\). Екрануються одинарні лапки ('), подвійні лапки ("), зворотний слеш (\) і NUL (байт NULL).

```
$str = "Is your name O'reilly?";
// Виводить: Is your name O \'reilly?
echo addslashes($str);
```

stripslashes

Видаляє екранування символів, додане функцією addslashes()

5.1.2.6 Опрацювання кирилиці

Для PHP типові випадки, коли функції некоректно або взагалі не працюють з кирилицею. Деякі функції з приставкою mb вирішують проблеми з кирилицею. Наприклад, mb_strtolower - приведення рядка до нижнього

регістру на відміну від strtolower. Те, що символ є літерою визначається на підставі властивостей символу Юнікоду.

Функції для роботи з багатобайтовими рядками

5.1.2.7 Довідник функцій для роботи з рядками в PHP

addcslashes - Екранує спецсимволи в стилі мови C

addslashes - Екранує спецсимволи в рядку

bin2hex - Перетворює бінарні дані в шістнадцяткові

chr - Повертає символ по його коду

chunk_split - Розбиває рядок на фрагменти

convert_cyr_string - Перетворює рядок з одного кириличного кодування в іншу

count_chars - Повертає інформацію про символи, що входять в рядок

crc32 - Обчислює CRC32 для рядка

crypt - Необоротне шифрування (хешування)

echo - Виводить один або більше рядків

explode - Розбиває рядок на підрядки

fprintf - Записує відформатований рядок в потік

get_html_translation_table - Повертає таблицю перетворень

hebrev - Перетворює текст на івриті з логічного кодування в візуальне

hebrevc - Перетворює текст на івриті з логічного кодування в візуальне з перетворенням переклад

htmlentities - Перетворює символи до відповідних HTML тегів

htmlspecialchars - Перетворює спеціальні символи в HTML теги

html_entity_decode - Перетворює HTML теги до відповідних символи

implode - Об'єднує елементи масиву в рядок (масиви в рядок)

localeconv - Повертає інформацію про числових форматах

ltrim - Видаляє пропуски з початку рядка

md5 - Повертає MD5 хеш рядка

md5_file - Повертає MD5 хеш файлу

metaphone - Повертає ключ metaphone для рядка

nl2br - Вставляє HTML код розриву рядка перед кожним переведенням рядка

number_format - Форматує число з поділом груп

ord - Повертає ASCII код символу

parse_str - Розбирає рядок на змінні

print - Виводить рядок

printf - Виводить відформатований рядок

quoted_printable_decode - Розкодовує рядок, закодований методом quoted printable

quotemeta - Екранує спеціальні символи

rtrim - Видаляє пропуски з кінця рядка

sha1 - Повертає SHA1 хеш рядка

sha1_file - Повертає SHA1 хеш файлу

similar_text - Обчислює ступінь схожості двох рядків

soundex - Повертає ключ soundex для рядка

sprintf - Повертає відформатований рядок

sscanf - Розбирає рядок відповідно до заданого формату

strcasecmp - Порівняння рядків без урахування регістру, безпечне для даних в двійковій формі

strcmp - Порівняння рядків, безпечне для даних в двійковій формі

strcoll - Порівняння рядків з урахуванням поточної локалі

strcspn - Повертає довжину ділянки на початку рядка, який відповідає масці

stripcslashes - Видаляє екранування символів, вироблене функцією addcslashes()

stripos - Повертає позицію першого входження підрядка без урахування регістру

stripslashes - Видаляє екранування символів, вироблене функцією addslashes()

strip_tags - Видаляє HTML і PHP теги з рядка

stristr - Аналог функції strstr, але незалежний від регістру

strlen - Повертає довжину рядка

strnatcasecmp - Порівняння рядків без урахування регістру з використанням алгоритму

strnatcmp - Порівняння рядків з використанням алгоритму "природного упорядкування"

strncasecmp - Порівняння перших n символів рядків без урахування регістру, безпечне для даних в двійковій формі

strncmp - Порівняння перших n символів рядків без урахування регістру, безпечне для даних в двійковій формі

strpos - Знаходить перше входження підрядка в рядок

strrchr - Знаходить останнє входження символу в рядок

strrev - Перевертає рядок

strripos - Повертає позицію останнього входження підрядка без урахування регістру

strrpos - Знаходить останнє входження символу в рядок

strspn - Повертає довжину ділянки на початку рядка, відповідного масці

strstr - Знаходить перше входження підрядка

strtok - Розбиває рядок

strtolower - Перетворює рядок в нижній регістр

strtoupper - Перетворює рядок в верхній регістр

strtr - Перетворює задані символи

str_ireplace - Регістро-незалежний варіант функції str_replace()

str_pad - Доповнює рядок інший рядком до заданої довжини

str_repeat - Повертає повторювану рядок

str_replace - Замінює рядок пошуку на рядок заміни

str_rot13 - Виконує над рядком перетворення ROT13

str_shuffle - Переставляє символи в рядку

str_split - Перетворює рядок в масив

str_word_count - Повертає інформацію про слова, які входять в рядок

substr - Функція повертає частину рядка

substr_count - Підраховує кількість входжень підрядка в рядок

substr_replace - Замінює частину рядка

trim - Видаляє пробіли з початку і кінця рядка

ucfirst - Перетворює перший символ рядка в верхній регістр

ucwords - Перетворює в верхній регістр перший символ кожного слова в рядку

vprintf - Виводить відформатований рядок

vsprintf - Повертає відформатований рядок

wordwrap - Виконує перенесення рядка на дану кількість символів з використанням символу розриву рядка

5.2 Хід роботи

Виконайте наведені нижче завдання, посилання (текст посилання повинна містити «Завдання Номер») на файл із реалізацією кожного із завдань розмістіть у файлі lastname_lab5.php папки lab5.

1. Ознайомтеся із теоретичними відомостями роботи з файлами і реалізуйте як мінімум п’ять наведених прикладів, обов'язково реалізуйте

приклад 5.8, у якому файли повинні завантажуватися на вебхостинг в папку lab5/files.

2. Використовуючи PHP-скрипт і форму в рамках одного документа створіть сценарій, в якому користувач буде вводити у текстове поле ім'я файла і після натискання кнопки “Готово” виконається перевірка, чи існує такий файл у поточному каталозі. Якщо він не існує, виведеться повідомлення: “Файл з іменем Zrazok.txt у поточному каталозі не існує”. Якщо ж файл існує, виведеться повідомлення про існування файлу, крім того дані про розмір, час створення, час останньої модифікації і вміст файла. Біля поля вкажіть як підказку назву вашого файлу із роботою last_name_lab5.php

3. У текстовому файлі lab5/files/tag1.txt в першому рядку впишіть тег (без дужок <>) у другому - його опис, в третьому - інший тег, у четвертому – його опис і так далі, усього 5 - 6 тегів. Далі в PHP-скрипті організуйте зчитування даних файлу по одному рядку і виведення його вмісту у вигляді:

** **	розриває рядок
<td>	відкриває комірку в таблиці

4. Модифікована задача п. 3: У текстовому файлі lab5/files/tag2.txt в першому рядку впишіть тег (з кутовими дужками <>) і його опис, в другому — інший тег і його опис, і так далі, усього 5 - 6 тегів. Далі в PHP-скрипті організуйте зчитування даних файлу і виведення його вмісту у вигляді:

** **	розриває рядок
<td>	відкриває комірку в таблиці

Для поділу вмісту рядка на два елементи масиву можете використати функцію explode().

Скрипт також повинен порахувати, скільки всього тегів описано у файлі. Відповідь запишіть у файл lab5/files/out.txt у вигляді наступного тексту: “Всього у файлі tag2.txt описано тегів: 5 ”.

Якщо даний файл відсутній, створіть його для читання і запису.

Зчитайте вміст з файлу lab5/files/out.txt і виведіть його під таблицею.

5. Реалізуйте наступні функції (текст візьміть згідно свого варіанту із наступного завдання і збережіть у файл під назвою last_name_text.txt.

Файли з текстами розміщуйте у папці lab5/files, під час виконання функцій текст зчитується з відповідних файлів):

5.1. Функція, що виводить слова заданого тексту (файл last_name_text.txt) у алфавітному порядку (для розбиття тексту на окремі елементи можна скористатися функцією explode).

5.2. Задано текст (файл last_name_text.txt). Виведіть список двох слів, які найчастіше зустрічаються у тексті.

5.3. Задано текст (файл last_name_text.txt). Виведіть найдовше слово тексту і його довжину, якщо декілька слів мають найбільшу довжину, то вивести всі з них.

5.4. Задано текст (файл last_name_text.txt). Виведіть найкоротше слово тексту і його довжину, якщо декілька слів мають найкоротшу довжину, то виведіть всі з них.

5.5. Знайдіть в тексті (файл last_name_text.txt) всі слова, які починаються на першу літеру вашого імені. Якщо таких слів не буде, то додайте до тексту будь-яке речення, яке містить необхідні слова.

5.6. В тексті (файл last_name_text.txt) замініть всі малі літери “о” на великі “О”.

5.7. Створіть PHP-сценарій, який випадковим чином виводить абзац тексту з п’яти заданих абзаців (текст візьміть зі свого варіанту + текст із наступних 4-х варіантів).

6. Виконайте завдання згідно Вашого варіанту і двох наступних, використовуючи функції обробки рядкових змінних:

№	Умова	Текст
1	Визначити кількість слів, які починаються з символу «м».	Синтаксис мови PHP бере початок з C, Java і Perl. PHP досить простий для вивчення. Перевагою PHP є надання web- розробникам можливості швидкого створення динамічних web-сторінок.
2	Визначити кількість слів, які містять хоча б один символ «є».	Ефективність є виключно важливим чинником при програмуванні для розрахованих на багато користувачів середовищ, до яких належить і web. Дуже важлива перевага PHP полягає в його транслюючому інтерпретаторі. Такий пристрій дозволяє обробляти сценарії з достатньо високою швидкістю.

3	Визначити кількість слів, які містять символ «і» рівно два рази.	З точки зору типізації, PHP є мовою програмування з динамічною типізацією. Немає необхідності явного визначення типу змінних, хоча така можливість існує. В разі звернення до змінної, ядро PHP трактує її тип відповідно до контексту. За необхідності можливе приведення змінної до певного типу за допомогою відповідних конструкцій мови.
4	Визначити кількість слів у тексті і кількість літер в першому і останньому словах.	До базових типів належать булеві дані, цілі та дійсні числа із плаваючою крапкою, а також рядки. Булеві дані виражають істинність значення. Цілі числа можуть бути подані у вісімковому, десятковому та шістнадцятковому вигляді. Розмір цілого числа може змінюватись залежно від платформи, як правило, розрядність становить 32 біти.
5	Знайти та роздрукувати слово в тексті максимальної довжини.	PHP не підтримує беззнакові цілі числа. Дійсні числа із плаваючою крапкою можуть бути подані в десятковій або експоненційній формі. Рядки розділяють на два класи — рядки, що підлягають аналізу, та рядки, що не підлягають аналізу.
6	Роздрукувати в стовпчик всі різні слова заданого тексту, вказавши для кожного кількість входжень в текст.	PHP надає широкий спектр функцій для пошуку та заміни тексту в рядках. Для цього використовують як традиційний підхід, так і спеціальний підхід, що базується на використанні регулярних виразів.
7	Надрукувати заданий текст, видаливши з нього повторне входження слів. Повідомити про кількість	Константи в PHP - ідентифікатори простих значень. Можливе визначення константи, причому після її оголошення стає неможливою зміна її значення чи анулювання. Константи можуть мати лише скалярні значення. Підтримується можливість отримання значення константи за динамічним ім'ям.

	зроблених видалень.	
8	Надрукувати заданий текст, видаливши в ньому зайві пробіли (тобто з декількох поспіль пробілів залишити тільки один).	Оператори бувають трьох типів - унарні, бінарні та тернарні. Оператори, як і в інших мовах характеризуються не лише дією, а й асоціативністю та пріоритетністю.
9	Здійснити заміну символів ':' на ',' в першому рядку, починаючи з п'ятої позиції.	Функції в сенсі мови є контейнерами коду: причому можливе поміщення інших функцій та класів. На цьому і базується можливість умовного визначення функції. В цьому випадку висувається вимога попередньої декларації викликаної функції, що не обов'язково в інших випадках.
10	Дано текст, що містить кілька круглих дужок. Якщо дужки розставлені правильно (тобто кожній відкриваючій відповідає одна закриваюча), то вивести число 1. Якщо ні, то вказати, кількість незакритих дужок.	Протокол HTTP, (засобами якого, як правило, обмінюються інформацією клієнт та Web-сервер не надає змогу зберегти стан сеансу взаємодії. Це випливає із того, що між клієнтом та сервером не встановлюється постійне з'єднання (і клієнт не надає жодних відомостей, що можуть виділити його з поміж інших активних в деякому околі часу).
11	Дано два рядки. Визначити кількість входжень слів з першого рядка у другий.	Мова програмування - система позначень для опису алгоритмів і структур даних. Мова програмування визначає набір лексичних, синтаксичних та семантичних правил, які

		задають зовнішній вигляд програми і дії, які виконує комп'ютер під її управлінням.
12	Потроїти кожне входження символу «Р» в кожному слові.	PHP інтерпретується веб-сервером в HTML-код, який передається на сторону клієнта. На відміну від таких скриптових мов програмування, як JavaScript, користувач не має доступу до PHP-коду, що є перевагою з точки зору безпеки, але значно погіршує інтерактивність сторінок
13	Вивести на екран речення у якому знаходиться слово «програмування».	У більшості візуальних середовищ програмування реалізовано функції автоматичної генерації коду. Традиційно автоматична генерація коду використовується для створення форм, кнопок, перемикачів та інших візуальних елементів. Деякі сучасні середовища можуть автоматично генерувати мало не будь-які фрагменти програм, а то навіть і цілі програми.
14	У тексті між словами вставити замість «,» символ «;».	Сесія, наприклад в мові програмування PHP, дуже важлива одиниця. Завдяки сесії можна передавати дані від одного файлу до наступного без явних ознак відправлення даних, тобто метод POST та GET не використовується.
15	Видалити частину тексту, яка взята в дужки. Дужки не видаляти.	PHP - мова, яка може бути вбудованою безпосередньо в html-код сторінок, які, в свою чергу коректно будуть оброблені PHP -інтерпретатором. Механізм PHP просто починає виконувати код після першої екрануючої послідовності (<?) і продовжує виконання до того моменту, коли він зустріне парну екрануючу послідовність (?>).
16	Підрахувати кількість слів у тексті. Роздрукувати текст, виводячи на екран слово	Прикладом мови низького рівня є асемблер. Мови низького рівня орієнтовані на конкретний тип процесора і враховують його особливості, тому для перенесення програми на асемблері на іншу апаратну платформу її потрібно майже повністю переписати

	«асемблер» в зворотньому порядку.	
17	Підрахувати кількість речень у тексті і кількість слів «PHP».	PHP 3.0 був офіційно випущений в червні 1998 року після 9 місяців публічного тестування. Оновлення PHP 4 здійснювалося тільки до кінця 2007 року.
18	Визначити кількість слів, у яких довжина перевищує 7 символів.	PHP застосовувався при розробці таких CMS, як Drupal, Joomla, PHP-Nuke. З його використанням розроблялися системи для веб комерції - osCommerce і Magento, утиліти адміністрування СУБД - phpMyAdmin, галереї зображень - Gallery Project, Coppermine і багато іншого.
19	Розділити заданий текст на декілька масивів, що можуть містити не більше 20 символів.	MySQL - це одна з найпопулярніших і найпоширеніших СУБД в Інтернеті завдяки вдалому поєднанні користувацьких властивостей, відкритому коду і добрій технічній підтримці. Офіційний сайт - www.mysql.com
20	Перевірити текст на наявність однакових слів, вивести ці слова.	PHP є мовою програмування з динамічною типізацією. Синтаксис PHP подібний синтаксису мови Сі. Деякі елементи, такі як асоціативні масиви і цикл foreach, запозичені з Perl. PHP є мовою програмування з динамічною типізацією.
21	Замінити в тексті усі подвійні лапки на одинарні, а крапки - трьома крапками.	Розробники PHP відмовилися від доповнення про персональне використання, яке було в абревіатурі PHP/FI. Мова була названа просто PHP - абревіатура, що містить рекурсивний акронім (англ. PHP: Hypertext Preprocessor - "PHP: Препроцесор Гіпертексту").
22	Роздрукувати в стовпчик усі слова тексту, окрім слів «PHP» та «MySQL».	PHP - мова програмування, спеціально розроблена для написання web-додатків (скриптів, сценаріїв), що виконуються на Web-сервері. Мова програмування PHP, особливо в зв'язці з популярною базою даних

		MySQL - оптимальний варіант для створення інтернет-сайтів різної складності.
23	Перевірити, чи є в тексті числа, визначити, перед якими словами вони розташовані.	Оновлення PHP 4 випускатимуться тільки до кінця 2007 року. До цього ж часу здійснюватиметься офіційна підтримка четвертої версії. Далі до 8 серпня 2008 року в міру необхідності з'являтимуться тільки критичні оновлення безпеки.
24	Визначити кількість речень у заданому тексті. Роздрукувати текст, виводячи на екран слово «CSS» в зворотньому порядку.	CSS (Cascading Style Sheets) — каскадні таблиці стилів, які застосовуються для візуального форматування документу в мовах розмітки. CSS використовується для того, щоб визначити кольори, шрифти, та інші аспекти вигляду сторінки.
25	Визначити, скільки слів у тексті починається з літери, з якої починається третє речення. Дану літеру визначити засобами функцій опрацювання рядків.	Функцією називається фрагмент програмного коду, що володіє унікальним ім'ям і призначений для вирішення конкретної задачі. Функція викликається по імені в різних точках програми, що дозволяє багато разів виконувати фрагмент з вказаним ім'ям. Перевага такого рішення полягає в тому, що блок коду пишеться лише один раз, а потім легко модифікується по мірі необхідності.
26	Замінити символи «!» на «?» у тих реченнях, які містять символ «:». Обчислити кількість здійснених замін.	Без чого подальше вивчення Web-програмування неможливе! Щоб створити сайт потрібно знати хоча б мову розмітки! Найлегшою і найпопулярнішою є мова HTML. Що таке: HTML!
27	Визначити довжину кожного	Окрім функцій та операторів PHP існують змінні. Змінна — це іменована ділянка пам'яті, де знаходяться якісь дані. Ім'я змінної

	речення (рядка) в тексті.	починається з знака $, за яким йде буква або знак підкреслення з подальшими в будь-якій кількості буквами, цифрами або символами підкреслення.
28	Перетворіть у верхній регістр перший символ кожного слова у тексті.	JavaScript — мова програмування, за допомогою якої Ви можете створювати інтерактивні Web-сторінки. Величезною перевагою JavaScript перед іншими мовами програмування є те, що їй не потрібно ніяких сторонніх інтерпретаторів, а достатньо тільки одного браузера.
29	Змінити регістр кожного слова в тексті, якщо воно не починається з цифри.	П'ята версія PHP була випущена розробниками 13 липня 2004 року. Зміни включають оновлення ядра Zend (Zend Engine 2), що істотно збільшило ефективність інтерпретатора. Введена підтримка мови розмітки XML
30	Знайти в тексті слова «унарні» та «мови» і перенести їх у кінець тексту.	Оператором називається конструкція мови програмування, що отримує один або більше аргументів, виконує над ними операцію, та повертає результат, який може бути новим аргументом. В залежності від того, скільки значень приймає оператор, виділяють три виду операторів: унарні, бінарні та тернарні.

7. Здайте і захистіть лабораторну роботу..

5.3 Контрольні запитання

1. Для чого призначена спеціальна конструкція мови PHP include?
2. Для чого призначена спеціальна конструкція мови PHP require?
3. Яка різниця між include і require?
4. Які ви знаєте режими відкриття файлів?
5. Які ви знаєте функції порівняння рядків?
6. Які ви знаєте функції перетворення рядків і файлів до формату HTML і навпаки?

Лабораторна робота №6. Тема: PHP. Регулярні вирази

Тривалість: 2 акад. години

Мета: навчитися будувати і застосовувати регулярні вирази

Завдання: вивчити правила формування регулярних виразів та виконати всі пункти лабораторної роботи.

6.1 Теоретичні відомості

1. Синтаксис регулярних виразів
2. Функції роботи із регулярними виразами

6.2 Хід роботи

1. Задано текст (файл text.txt). Виведіть з нього на веб-сторінку:
   - весь заданий текст;
   - тільки назви відкриваючих тегів;
   - назви відкриваючих тегів разом і кутовими дужками;

Примітка: теги із кутовими дужками не повинні впливати на форматування веб-сторінки.

2. У тексті (файл text.txt), використовуючи регулярні вирази, знайти всі входження слова незалежно від регістру (в тому числі слова, частиною яких є вказане слово):
   - тег;
   - HTML;
   - частини слова, введеної в однорядкове текстове поле.

Вивести тільки ті речення, в яких дане слово присутнє, в порядку спадання числа знайдених відповідностей шаблону пошуку (першим повинно йти речення, в якому шукане слово зустрічається найчастіше), знайдені слова виділити жирним шрифтом.

3. Створіть власний фрагмент html тексту з відкриваючими і закриваючими тегами, який починається і закінчується тегом body. З допомогою функції, замініть відкриваючі і закриваючі теги на пропуски, залишивши тільки змістовну частину. Замініть подвійні пропуски, які утворилися при заміні розмітки, на один пробіл. Виведіть утворений текст.

4. Створити форму, яка міститиме наступні поля заповнення:

- ім’я;
- прізвище;
- логін;
- пароль;
- повторити пароль
- адреса електронної пошти (e-mail)

Використовуючи регулярні вирази і функцію preg_match(), виконати перевірку валідності інформації, введеної в поля форми (ім’я та прізвище повинно містити лише літери: перша літера велика, наступні маленькі, дозволено кирилиця і латина; логін тільки латина малими літерами; пароль: мінімум 6 символів, з яких мінімум по 1 букві і цифрі; повторити пароль: повинен співпадати із попередньо введеним; e-mail на наявність символу @ і відповідної структури. Якщо поле заповнене вірно, то навпроти поставити зображення галочки як позначення коректності введення, інакше колір тексту і рамки стає червоним, а сама рамка подвійною, збоку поля з’являється вимога до заповнення.

5. Перевірте коректність введення почтового індексу з використанням регулярних виразів за варіантом, що надано в таблиці.

Перед полем для введення напишіть назву країни, вимоги для введення поштового індексу і зразок коректного введення. Якщо поле заповнене вірно, збоку від поля виведіть повідомлення про коректність введення, інакше - повідомлення про некоректність введення і вказівкою для повторного введення

Індивідуальні варіанти завдання

№	Країна	Алгоритм формування індексу
1	Сполучені Штати Америки	Числовий код з дев’ятьма цифрами. Після п’ятої дефіс
2	Нідерланди	Чотири цифри. Перша цифра не нуль. Після пропуску дві літери
3	Італія	П’ять цифр. Попереду “V-“ чи “I-“.
4	Індія	Шість цифр. Перша цифра не нуль. Пропуск після третьої
5	Польща	Дві цифри, дефіс, три цифри

6	Іспанія	П’ять цифр. Після третьої цифри пропуск. Попереду “I-“.
7	Великобританія	Від 5 до 8 цифр і літер, що розділені після четвертої пропуском. Наприклад, AA9A 9AA
8	Латвія	Чотири цифри. Попереду “V-“ чи “I-“.
9	Македонія	Чотири цифри. Перша від 1 до 7
10	Португалія	Чотири цифри, дефіс, три цифри, пропуск, 5 літер

6. Здайте і захистіть лабораторну роботу.

6.3 Контрольні запитання

1. Назвіть відомі Вам формати регулярних виразів.

2. Поясніть використання функцій preg_match(), preg_match_all(), preg_replace().

3. Який пошук здійснюватиме вираз:

- “/^\w{6,9}\d{3}/s”;
- “/ \d{3}-\d{2}-\d{2}/m”?

4. Що можна вказувати в параметрі модифікатор?

Лабораторна робота №7. Об'єктно-орієнтоване програмування на PHP

Тривалість: 2 акад. години

Мета: ознайомитися із основами об'єктно-орієнтоване програмування на PHP.

Завдання: вивчити та навчитися застосовувати основні поняття об'єктно-орієнтованого програмування.

7.1 Теоретичні відомості

Стало стандартом, що вивчення об'єктно-орієнтованого програмування (ООП) завжди починається з поняття «класу». Ідея об'єктно-орієнтованого програмування полягає в тому, що в ООП ми починаємо думати сутностями. А що таке сутність? Це все, що нас оточує, тобто це звичайні об'єкти. А об'єктом у нас може бути все що завгодно (автомобіль, будинок, комп'ютер і т.д.). Що таке клас? - Клас описує ці об'єкти, тобто є їх формальною моделлю.

Наприклад, у нас є об'єкт «автомобіль». Перш ніж його зібрати, нам потрібне креслення. У цьому кресленні має прописуватися те, що ми повинні знати про цей автомобіль - це те, що автомобіль має деякі властивості (колір, розмір, максимальна швидкість, рівень безпеки) і методи (автомобіль може рухатися вперед і назад, повертати кермо, стежити за втомою водія і т.д.).

В об'єктно-орієнтованому програмуванні для того, щоб описати таке креслення існує поняття «клас». «Клас» обов'язково повинен включати в себе опис таких понять, як властивості і методи даного об'єкту.

Отже, як же записується «клас»? Записується він у такий спосіб - спочатку йде ключове слово class, потім ім'я класу і фігурні дужки, в яких описуються властивості і методи:

```
<?php
class Car{
// Опис властивостей
// Опис методів
}
?>
```

Зверніть увагу, що імена класів чутливі до регістру.

Ґрунтуючись на цьому класі можна створювати об'єкти, в ООП вони ще називаються «екземплярами класу». Об'єкт класу створюється наступним чином - ставиться змінна, далі йде ключове слово new і ім'я класу з круглими дужками:

```
<? php
// Опис об'єкта
class Car {
// Опис властивостей
// Опис методів
}
// Створення об'єкта
$car = new Car();
?>
```

Для одного класу (креслення) можна створювати кілька екземплярів класу або об'єктів (автомобілів):

```
<? php // Опис об'єкта
class Car {   // Опис властивостей
// Опис методів
} // Створення об'єкта
$car1 = new Car();
$car2 = new Car();
?>
```

Тепер давайте розберемо, що ж таке властивості і методи об'єкта. Почнемо з властивостей. Це звичайні змінні. Але не зовсім. Якщо змінна знаходиться десь в коді, то це звичайна змінна. А якщо змінна знаходиться в класі, то ця змінна - властивість.

Наприклад, у автомобіля є рік випуску. Відповідно ініціалізуємо змінну $year. Ми можемо присвоїти їй значення по замовчуванню, наприклад 2018. Також у автомобіля є максимальна швидкість, ініціалізуємо змінну $speed. Ще у автомобіля є марка, ініціалізуємо змінну $model. Всі ці змінні - це властивості об'єкта, тобто нашого автомобіля.

Однак в об'єктно-орієнтованому програмуванні перед такими змінними-властивостями прийнято ставити модифікатори доступу. Що таке модифікатори доступу? Це ключові слова, які позначають можливість доступу або можливість звернутися з методів об'єкта або з примірників класу до властивостей цього об'єкта. Ці ключові слова записуються як public, protected, private.

Доступ до властивостей і методів класу, оголошених як public (загальнодоступний), дозволений звідусіль. Модифікатор protected (захищений) надає доступ наслідуваним і батьківським класам. Модифікатор private (закритий) обмежує область видимості так, що тільки клас, де оголошений сам елемент, має до нього доступ.

Поки що будемо використовувати модифікатор public:

```
<? php
// Опис об'єкта
class Car {
public $year = 2018;
public $speed;
public $model;
}
// Створення об'єкта
$car1 = new Car();
$car2 = new Car();
?>
```

Тепер розберемо, що у нас тут відбувається. Ми з класу створили два об'єкти або два екземпляри класу, які мають однакові властивості, два з яких не визначені ($speed, $model) і одну властивість, у якої задане значення по замовчуванню ($year = 2018).

Щоб звернутися до однієї з властивостей створеного об'єкта, необхідно написати ім'я змінної, якій присвоєно об'єкт, далі поставити ось таку стрілку -> (дефіс і знак більше) і ім'я властивості, але без знака долара:

```
<? php // Опис об'єкта
class Car {
public $year = 2018;
public $speed;
```

```
public $model;
}
// Створення об'єкта
$car1 = new Car();
$car2 = new Car();
echo $car1-> year; // звернення до властивості об'єкта
?>
```

Далі присвоюємо першому і другому автомобілю швидкість і модель. Так як у нас немає значення швидкості за замовчуванням і немає значення моделі, то можна зробити наступним чином: $car1-> speed = 210, $car1-> model = «bmw», $car2-> speed = 260, $car2 -> model = «lexus»:

```
<? php
// Опис об'єкта
class Car {
public $year = 2018;
public $speed;
public $model;
}

// Створення об'єкта
$car1 = new Car();
$car1-> speed = 210; // присвоєння значення властивості об'єкта
$car1-> model = "bmw"; // присвоєння значення властивості об'єкта

$car2 = new Car();
$car2-> speed = 260; // присвоєння значення властивості об'єкта
$car2-> model = "lexus"; // присвоєння значення властивості об'єкта
echo $car2-> year; // звернення до властивості об'єкта
?>
```

Методи

Як було зазначено вище, всередині класу є дві речі: перша - це властивості, і друга - це методи. Тепер детальніше зупинимося на них. Що таке методи об'єкта? Якщо говорити зовсім простою мовою, то це поведінка об'єкта.

Згадайте про об'єкт «Автомобіль». Автомобіль може рухатися вперед і назад, а також вправо або вліво, вона може розвивати певну швидкість. Все це є методами, тобто те, що може робити даний об'єкт.

Метод - це та ж сама функція! І описується він точно так само, як звичайна функція.

Тільки функція, яка лежить десь в коді, це звичайна функція, а функція, описана в класі (або по-іншому сказати, що знаходиться всередині класу) - це метод. Запам'ятайте це!

У всіх методів, так само, як і у властивостей є точно такі ж модифікатори доступу (public, protected, private). Якщо не вказати жоден з модифікаторів, то за замовчуванням буде вважатися модифікатор public. Однак, краще завжди вказувати один з них.

Давайте на прикладі розгляне, як створюється метод. Продовжимо розгляд на прикладі класу, де ми описували властивості об'єкта «автомобіль». Усередині цього класу ми опишемо найпростіший метод і називатися він у нас буде function takeSpeed().

Отримати доступ до цього методу можна точно так само, як і до властивості, наприклад:

$car1-> takeSpeed():

```
<? php
// Опис об'єкта
class Car {
// властивості
public $year = 2018;
public $speed;
public $model;

// метод
public function takeSpeed() {
// Тут метод щось робить
echo "Швидкість автомобіля =";
}
}
// Створення об'єкта
$car1 = new Car();
// Отримуємо доступ до методу
```

```
$car1-> takeSpeed();
$car2 = new Car();
?>
```

Якщо запустити цей код, то у нас викликається наш метод і буде виведено: «Швидкість автомобіля =».

Ось тут починається найцікавіше. Ви звернули увагу, швидкість якого автомобіля ми хочемо отримати? Логічно припустити, що раз ми звертаємося з екземпляра класу car1, то і швидкість ми хочемо отримати автомобіля bmw. Однак, в методі, який у нас відповідає за отримання швидкості автомобіля, не вказано якого саме об'єкту швидкість нам потрібна. У цьому випадку нам необхідно звернутися до властивості speed класу Car. Але вона у нас одна, а автомобіля (об'єкта) два.

Так ось, нам потрібно вказати властивість (швидкість) якого об'єкта ми маємо на увазі.

Таким покажчиком в ООП є ключове слово $this.

І записується це так: $this-> speed.

```
<? php
// Опис об'єкта
class Car {
// властивості
public $year = 2018;
public $speed;
public $model;

// метод
public function takeSpeed() {
// Вказуємо метод, чию швидкість ми хочемо отримати
echo "Швидкість автомобіля =". $this-> speed;
}
}
// Створення об'єкта
$car1 = new Car();
$car1-> speed = 210; // присвоєння значення властивості об'єкта
// Отримуємо доступ до методу
$car1-> model = "bmw"; // присвоєння значення властивості об'єкта
$car1-> takeSpeed();
```

```
$car2 = new Car();
$car2-> speed = 260; // присвоєння значення властивості об'єкта
// Отримуємо доступ до методу
$car2-> model = "lexus"; // присвоєння значення властивості об'єкта
?>
```

Тепер у нас вже виведеться: Швидкість автомобіля = 210. Тобто ми отримали швидкість об'єкта $car1 (bmw).

Тут важливо запам'ятати - щоб звернутися з методу до властивості об'єкта, ми звертаємося через ключове слово $this.

Давайте розглянемо ще один важливий момент - припустимо у нас в класі Car крім методу takeSpeed(), є метод paintSpeed(). Ми зробимо так, що метод paintSpeed() буде виводити швидкість автомобіля, а метод takeSpeed() буде викликати метод paintSpeed(). Це досить поширена ситуація, що показує - для того щоб звернутися з методу до іншого методу, також використовується ключове слово $this. Поглянемо на код:

```
<? php
// Опис об'єкта
class Car {
// властивості
public $year = 2018;
public $speed;
public $model;

// метод
public function takeSpeed() {
// Звертаємося з методу takeSpeed() до методу paintSpeed()
$this-> paintSpeed();
}

// метод
function paintSpeed() {
// Вказуємо методу, чию швидкість ми хочемо отримати. Йде звернення з методу до властивості
echo "Швидкість автомобіля =". $this-> speed;
```

```
}
}
// Створення об'єкта
$car1 = new Car();
$car1-> speed = 210; // присвоєння значення властивості об'єкта
// Отримуємо доступ до методу
$car1-> model = "bmw"; // присвоєння значення властивості об'єкта
$car1-> takeSpeed();
$car2 = new Car();
$car2-> speed = 260; // присвоєння значення властивості об'єкта
// Отримуємо доступ до методу
$car2-> model = "lexus"; // присвоєння значення властивості об'єкта
$car2-> takeSpeed();
?>
```

Отже, звернення з методу до методу і звернення з методу до властивості здійснюється через ключове слово $this.

Дивіться, що тут відбувається. Ще нічого серйозного не зробили, а вже якось багато всього понаписувано. Щось тут не так і взагалі це йде в розріз з ідеологією ООП. Чи не можна це скоротити? Можна, можливо. І надалі теж будемо скорочувати кількість коду. У зв'язку з чим, ми може це зробити?

На даному етапі вивчення об'єктно-орієнтованого програмування з'являється таке поняття, як «конструктор класу». Що ж це таке?

Конструктор класу - це спеціальний метод, який автоматично викликається в момент створення об'єкту. У PHP конструктор класу має спеціальну назву - подвійне підкреслення construct (function __construct).

Давайте з вами перевіримо, як викликається конструктор для кожного створеного об'єкта. Запустіть цей код і Ви побачите, що для кожного об'єкта викликається конструктор:

```
<? php
class Users {
public $name;
public $login;
public $password;

// Створення конструктора
```

```
function __construct () {
echo "<p> Конструктор викликався автоматично!";
}

// Створюємо метод getInfo ()
function getInfo () {
echo "<p> Name:". $this-> name. "<br>";
echo "Login:". $this-> login. "<br>";
echo "Password:". $this-> password. "<br>";
}
}

$user1 = new Users ();
$user1-> name = "Vasya";
$user1-> login = "vas";
$user1-> password = 123;
// Виводимо метод getInfo ()
$user1-> getInfo ();

$user2 = new Users ();
$user2-> name = "Petya";
$user2-> login = "pet";
$user2-> password = 321;
// Виводимо метод getInfo ()
$user2-> getInfo ();

$user3 = new Users ();
$user3-> name = "Vova";
$user3-> login = "vov";
$user3-> password = 456;
// Виводимо метод getInfo ()
$user3-> getInfo ();
?>
```

Коли ми створювали об'єкт: $user1 = new Users (); ми ставили круглі дужки біля об'єкту Users(). Ці круглі дужки біля об'єкту - це дужки самого конструктора. Тобто, якщо ми передамо в круглих дужках об'єкта якийсь

параметр, то він потрапить в конструктор. Давайте подивимося, як це виглядає в коді. Ми в конструкторі вкажемо параметр у вигляді змінної і будемо її виводити, а коли створюємо об'єкт, в ньому вкажемо числа (1,2,3). Запустіть цей код і подивіться, що вийде:

```
<? php
class Users {
public $name;
public $login;
public $password;

// Створюємо конструктор з параметром
function __construct ($number) {
echo "<p> Конструктор викликався автоматично $number!";
}

// Створюємо метод getInfo ()
function getInfo () {
echo "<p> Name:". $this-> name. "<br>";
echo "Login:". $this-> login. "<br>";
echo "Password:". $this-> password. "<br>";
}
}

$user1 = new Users (1);
$user1-> name = "Vasya";
$user1-> login = "vas";
$user1-> password = 123;
// Виводимо метод getInfo ()
$user1-> getInfo ();

$user2 = new Users (2);
$user2-> name = "Petya";
$user2-> login = "pet";
$user2-> password = 321;
// Виводимо метод getInfo ()
$user2-> getInfo ();
```

```
$user3 = new Users (3);
$user3-> name = "Vova";
$user3-> login = "vov";
$user3-> password = 456;
// Виводимо метод getInfo ()
$user3-> getInfo ();
?>
```

Запам'ятайте - все, що ми передаємо в дужки об'єкта, потрапляє в конструктор.

Поняття “конструктор” є у всіх мовах програмування, але в PHP він позначається саме, як нижня __construct. Це метод автоматично викликається. Для чого використовується конструктор? Для ініціалізації чого-небудь. Наприклад, коли створюється новий об'єкт, ми хочемо щоб, щось відбувалося (щось в сесію записувалося, в файл що-небудь записувалося). Тобто, щось відбувалося автоматично. Це дуже схоже на автозагрузку операційної системи. Завантажилася операційна система і разом з нею сталися якісь дії, завантажилися програми і т.д. Тут те ж саме.

Однак, поряд з конструктором у нас є ще один метод, який використовується, коли закінчується код, тобто, коли код відпрацював і об'єкти видаляються. Звичайно ми можемо видалити об'єкт і раніше, ніж закінчиться код. Наприклад, функцією unset (). Але нам потрібно, щоб у разі вилучення об'єктів, неважливо примусово ми видаляємо їх чи вони видаляються після завершення коду, щось відбувалося, як і при створенні об'єктів.

Для цього у нас є метод, який називається деструктор. Цей метод автоматично викликається при видаленні об'єкта. У PHP він позначається, як подвійне нижнє підкреслення destruct (__destruct):

```
<? php
class Users {
public $name;
public $login;
public $password;

// Створення конструктора
```

```
function __construct ($number) {
echo "<p> Конструктор зголосився автоматично $number!";
}

// Створюємо деструктор
function __destruct () {
echo "<p> Об'єкт видалився!";
}

// Створюємо методод getInfo ()
function getInfo () {
echo "<p> Name:". $this-> name. "<br>";
echo "Login:". $this-> login. "<br>";
echo "Password:". $this-> password. "<br>";
}
}

$user1 = new Users (1);
$user1-> name = "Vasya";
$user1-> login = "vas";
$user1-> password = 123;
// Виводимо метод getInfo ()
$user1-> getInfo ();

$user2 = new Users (2);
$user2-> name = "Petya";
$user2-> login = "pet";
$user2-> password = 321;
// Виводимо метод getInfo ()
$user2-> getInfo ();

$user3 = new Users (3);
$user3-> name = "Vova";
$user3-> login = "vov";
$user3-> password = 456;
// Виводимо метод getInfo ()
$user3-> getInfo ();
```

?>

Запустивши цей код, ми бачимо, що при видаленні об'єктів, а вони видаляються у нас тут автоматично, викликається деструктор із записом - Об'єкт пішов. У нас було три об'єкти - відповідно виводяться три рази запис - «Об'єкт видалився!».

Зверніть увагу, що деструктор має круглі дужки, але передавати параметри в ці круглі дужки не можна! Ще один важливий момент - порядок видалення об'єктів ніколи не визначений. Тобто у нас є три об'єкти і їх треба видалити. Так ось, в якому порядку вони будуть видалятися невідомо. Тому є таке правило - з деструктора не звертатися до інших об'єктів, так як на момент виклику об'єкта, він може бути вже видалений.

Копіювання і передача об'єктів за посиланням

Тепер давайте поговоримо ось про що - копіювання і передача об'єктів за посиланням, яке так чи інакше завжди присутній в ООП. У чому ідея всього цього? Дивіться: $object2 = $object1. Як Ви думаєте, що це означає? Тут все залежить від того, якої версії PHP використовують — 4 чи >=5. Бо в 4 версії PHP це означало копіювання. Тобто об'єкт 2 - це копія об'єкта 1. А якщо ми хотіли передати об'єкт за посиланням, то ми писали ось таку річ — $object2 = & $object1. Що ж це означає - передати об'єкт за посиланням? Я думаю, що всі знають, що таке ярлик на робочому столі в Windows - це посилання на файл. Так ось тут те ж саме. Якщо більш зрозуміліше, то це два імені одного і того ж об'єкта. Все це стосується тільки 4 версії PHP.

А що ж у нас 5+ версії? А тут у нас ось цей вираз $object2 = $object1 вже буде посиланням. Тоді виникає питання, а як же мені тоді скопіювати об'єкт? А копіюється він за допомогою чарівного слова clone: $object2 = clone $object1.

Зверніть увагу, що при копіюванні об'єкта, конструктор не викликається. Але ж при копіюванні створюється новий об'єкт і у нас може виникнути необхідність, щось автоматично виконати.

У цьому випадку ми можемо описати спеціальний метод - подвійне нижнє підкреслення clone (__clone), який буде автоматично викликатися при копіюванні об'єктів:

```
<? php
class Users {
public $name;
```

```
public $login;
public $password;

// Створення конструктора
function __construct ($name, $login, $password) {
$this-> name = $name;
$this-> login = $login;
$this-> password = $password;
}

// Створюємо метод clone
function __clone () {
echo "<p> Об'єкт скопійований!";
}

// Створюємо метод getInfo ()
function getInfo () {
echo "<p> Name:". $this-> name. "<br>";
echo "Login:". $this-> login. "<br>";
echo "Password:". $this-> password. "<br>";
}
}

$user1 = new Users ( "Vasya", "vas", "123");
// Виводимо метод getInfo ()
$user1-> getInfo ();

$user2 = new Users ( "Petya", "pet", "321");
// Виводимо метод getInfo ()
$user2-> getInfo ();

$user3 = new Users ( "Vova", "vov", "456");
// Виводимо метод getInfo ()
$user3-> getInfo ();
// Об'єкт $user4 копія об'єкта $user3
$user4 = clone $user3;
?>
```

7.2 Хід роботи

Завдання 1:

1. створіть клас Student з властивостями name, suname, group;
2. створіть три об'єкти класу Student;
3. задайте довільні значення властивостям для кожного з об'єктів.

Завдання 2:

1. в класі Student потрібно описати метод getInfo ();
2. метод getInfo () повинен виводити значення властивостей об'єкта;
3. викличте метод getInfo () для кожного об'єкта.

Завдання 3:

1. в класі Student необхідно описати конструктор;
2. конструктор повинен задавати початкові значення властивостей name, suname, group;
3. створіть крім існуючих трьох об’єктів класу Student, додатково ще три об'єкти з використання конструктора.

Завдання 4:

1. в класі Student опишіть метод __clone ();
2. метод __clone повинен задавати початкові значення властивостей за замовчуванням при копіюванні об'єктів;
3. створіть сьомий об'єкт, скопіювавши один з наявних об'єктів.

Завдання 5:

Створіть клас для виведення таблиці множення для вказаного числа (передавати в конструкторі). Створити окремий метод для обчислення. Далі створити кілька об'єктів даного класу для демонстрації працездатності класу. Вивід оформити у вигляді таблиці.

Завдання 6:

Створіть клас країни в якому будуть поля: назва країни, населення і назва столиці (дані вказуйте тих країн, перша літера англійської назви яких співпадає з першою літерою Вашого прізвища, якщо такі країни відсутні, тоді імені, потім по-батькові). Створіть масив об'єктів, виведіть кожний з них у

таблицю в три рядки по дві комірки в кожному. У лівій комірці ім'я елемента, в правій - його значення.

Завдання 7:

Створіть клас користувача, з полями: прізвище, ім'я, вік і e-mail.

У HTML формі користувач вводить в чотири різні поля: прізвище, ім'я, вік і e-mail. Після натискання клавіші кнопки ГОТОВО створюється об'єкт користувача, з методом, який вносить ці дані в поля об'єкту і далі виводить їх використовуючи другий метод класу користувача.

У формі передбачити перевірку, що всі поля перед відправленням не порожні.

Завдання 8:

Здайте і захистіть лабораторну роботу.

7.3 Контрольні запитання

Що таке клас в ООП і як він описується?

Що таке об'єкт в ООП і як він описується?

Що таке метод в ООП і як він описується?

Що таке конструктор в ООП і як він описується?

Що таке деструктор в ООП і як він описується?

Лабораторна робота №8. PHP. Робота із СУБД

Тривалість: 2 акад. години
Мета: набути практичних навичок для роботи з PHP і СУБД.
Завдання: дослідити взаємне використання бази даних і PHP

8.1. Теоретичні відомості.

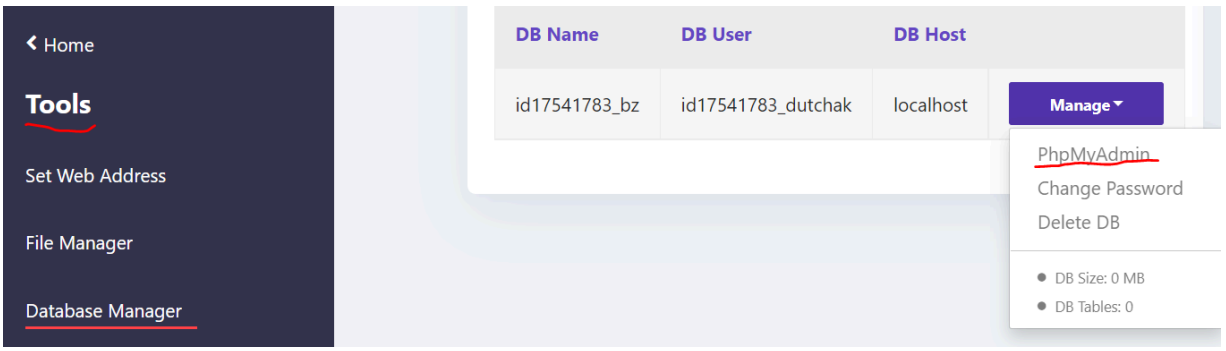
Уроки SQL
Взаємодія PHP-додатків з базами даних MySQL

8.2 Хід роботи

Увага! Для кожного із наступних завдань створіть посилання, при кліку на які будуть виконуватися наступні завдання.

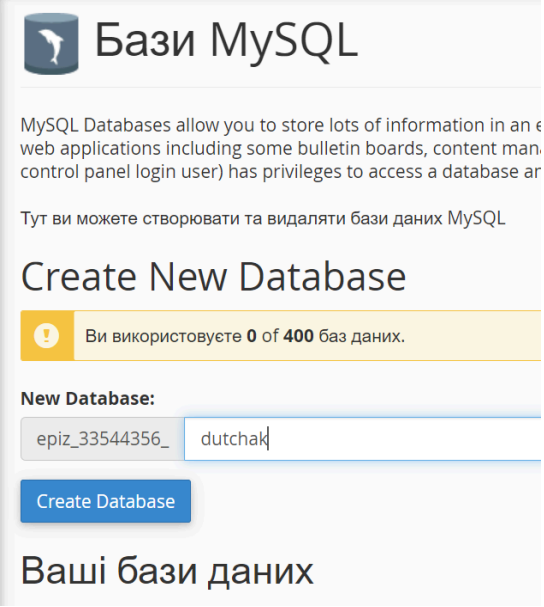
1. Скопіюйте на свій вебхостинг файли з папки PHP/lab8DB (якщо ви цього не зробили під час виконання Лабораторної роботи 1, зверніть увагу що у назві папки я вказала номер роботи), введіть свої дані для з'єднання із базою даних, розберіть і забезпечте коректне відображення запропонованих викладачем прикладів взаємодії PHP і СКБД MySQL. Назва Вашої бази даних, ім’я користувача і пароль знайдіть на Вашому вебхостингу). Переіменуйте папку labDB на lab8DB_lastname, де lastname - це Ваше прізвище.
2. Перейдіть в phpMyAdmin. Дослідіть меню додатка і створену таблицю.

https://www.000webhost.com/:

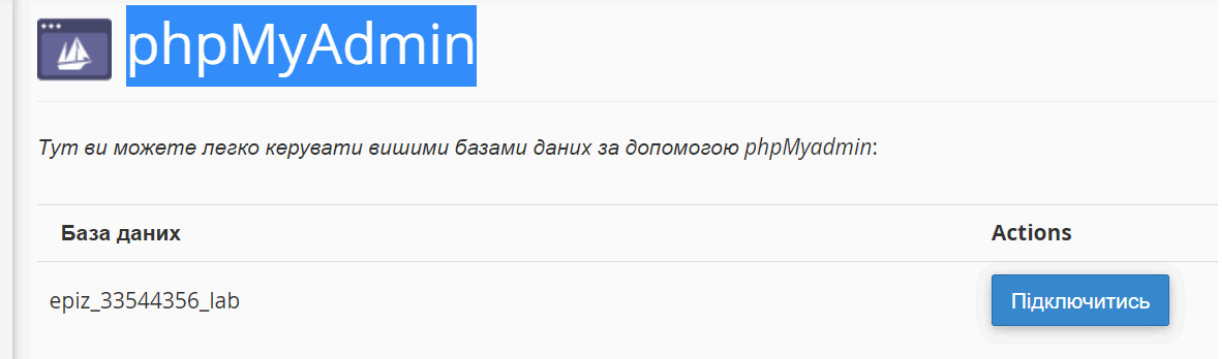


https://infinityfree.net/:

Для створення бази даних, перейдіть у Control panel -> Бази MYSQL, далі:



Для створення переходу в phpMyAdmin перейдіть у Control panel -> phpMyAdmin, далі виберіть потрібну базу даних і клікніть “Підключитись”:



3. Перейдіть у закладку SQL, додайте і видаліть кілька записів у таблицю з дисциплінами за допомогою мови запитів SQL.

4. У базі даних створіть таблицю користувачів (обов'язковими полями є: логін (має бути унікальним) та пароль, інші поля на свій розсуд). Розробіть PHP-сценарій обробки форми реєстрації і запису даних про користувача в таблицю користувачів. Занесіть мінімум трьох користувачів (себе і двох одногрупників, наступних по списку, в якості логіну вводьте прізвище на латині). Напишіть php-код, який під формою реєстрації виведе у вигляді таблиці всі дані (обов'язково логін і пароль, інші внесені дані за наявності) всіх внесених у базу даних користувачів.

5. Створіть таблицю Склад, і внесіть в неї дані мінімум про чотири найменування товару згідно варіанту (номер у списку групи): найменування, назва файлу із зображенням товару (або саме зображення), ціна за одиницю товару (шт., кг. і т.д.), наявна кількість. Виведіть на сторінку всю інформацію

про доступні товари із таблиці Склад у дві колонки. При кліку на зображення чи назву товару повинна відкритися сторінка, на яку виведіть всю інформацію про обраний товар і лічильник вибору кількості товару. Додайте кнопку «Купити», після кліку на яку зменшити наявну кількість товару на відповідну кількість купленого товару і вивести змінені дані з таблиці склад бази даних.

1. Птахи
2. Їжа
3. Домашні тварини
4. Гриби
5. Транспорт
6. Дерева
7. Фрукти
8. Овочі
9. Ягоди
10. Косметика
11. Кондитерські вироби
12. Двоколісний транспорт
13. Меблі
14. Комп’ютерна техніка
15. Жіночий одяг
16. Чоловічий одяг
17. Дитячий одяг
18. Взуття
19. Солодощі
20. Спортивні аксесуари
21. Музичні інструменти
22. Транспорт
23. Кухонна техніка
24. Побутова техніка
25. Автомобілі
26. Комп’ютерна техніка
27. Птахи
28. Їжа
29. Домашні тварини
30. Транспорт

6. На сторінку покупки товару, поруч із кнопкою «Купити», додати кнопку «Поповнити склад», яка навпаки буде збільшувати наявну кількість товару на вказану кількість.

7. Здайте роботу. При здачі роботи, у папці роботи також збережіть скріни екрану, які демонструють наповнення таблиць бази даних.

8.3 Контрольні запитання

1. Що таке База Даних?
2. Що таке SQL?
3. Синтаксис команди вибору окремих полів.
4. Назвіть призначення операторів: ORDER BY, WHERE, LIKE
5. Що таке СКБД MySQL?
6. Назвіть призначення і параметри функції mysqli

Лабораторна робота №9. PHP. Робота з базою даних. Створення сайту із новинами

Тривалість: 2 акад. години

Мета: набути практичних навичок для роботи з PHP і базою даних.

Завдання: дослідити взаємне використання бази даних і PHP та виконати завдання відповідно до ходу роботи.

9.1. Теоретичні відомості.

Уроки SQL

Взаємодія PHP-додатків з базами даних MySQL

9.2 Хід роботи

1. Створіть таблицю last_name_news, із полями що містять порядковий номер, тематику, заголовки, контент, дата новин. Задайте автоматичне додавання порядкового номеру (create sequence), поле із заголовками зробіть унікальним. Приклад створення і видалення таблиці:

mysqli_query($db_server,"drop table duscup_id");

mysqli_query($db_server,"drop sequence duscup_seq");

mysqli_query($db_server,"create sequence duscup_seq minvalue 1000 maxvalue 1999 start 1000");

mysqli_query($db_server,"select setval('duscup_seq',1000)");

mysqli_query($db_server,"create table duscup_id (id integer default nextval('duscup_seq') primary key, name_d varchar(100) unique, key_concepts text[])");

2. Занесіть у дану таблицю новини, розміщені у файлі file/news.txt. Зверніть увагу, що поля розділені «~», а записи «&». Внесення даних перевірте через phpMyAdmin.

fopen($file,"r") -відкриття файлу для читання;

fclose($file) - закриття файлу.

```
$file="file/news.txt";
$fdata_my = fopen($file,"r") or die ("Can't open file $file!");
$mas = fread($fdata_my,filesize($file));
$mas=explode("&",$mas);
```

```
for ($i=0;$i<count($mas);$i++){
echo "mas[$i]=$mas[$i]<p>";
$mas_vidm=explode("~",$mas[$i]);
for ($j=0;$j<count($mas_vidm);$j++){
echo "<p>mas_vidm[$j]=$mas_vidm[$j]<p>";
}
$res=mysqli_query($db_server,"insert into
last_name_news(tema,zagol,content,time) values('$mas_vidm[0]', '$mas_vidm[1]',
'$mas_vidm[2]', '$mas_vidm[3]')");
```

3. Додайте в таблицю last_name_news по кожній тематиці мінімум по одній новині за поточний рік. Виведіть на сторінку всі внесені дані із таблиці last_name_news у вигляді таблиці.

4. По зразку і стилях сайту ukr.net, виведіть спочатку три свіжих новини із заголовком Головне, а виведіть назву кожної тематики і для кожної тематики заголовок і дату створення трьох найновіших новин, першою виводьте найновішу. При кліку по заголовку тематики, повинні відкриватися всі новини відповідної тематики, а при кліку по заголовку новини - відкривається сторінка із відповідною новиною, що містить: заголовок, дату, контент. Створіть файл file/out.txt і запишіть в нього загальну кількість новин, доданих в таблицю баз даних;

5. Напишіть запит, би по кліку на кнопку видаляв з таблиці last_name_news новину, яка має порядковий номер 3. Виведіть на сторінку всі записи із таблиці last_name_news у вигляді таблиці, окрім даних поля з контентом.

6. Створіть форму для додавання новини у таблицю last_name_news.

7. Здайте роботу. При здачі роботи, у папці роботи також збережіть скріни екрану, які демонструють наповнення таблиць бази даних.

9.3 Контрольні запитання

1. Що таке База Даних?
2. Що таке SQL?
3. Синтаксис команди вибору окремих полів.
4. Назвіть призначення операторів: ORDER BY, WHERE, LIKE
5. Що таке СКБД MySQL?
6. Назвіть призначення і параметри функції mysqli
7. Назвіть призначення функцій fopen, fclose, fread

Лабораторна робота №10. Сеанси та сесії в PHP

Тривалість: 2 акад. години

Мета: набути практичних навичок для роботи з сеансами та сесіями в PHP.

Завдання: дослідити взаємне використання бази даних і PHP

10.1. Теоретичні відомості.

Тема 5. Сеанси та сесії в PHP Навчального посібника з дисципліни «Програмування мовою PHP»

10.2. Хід роботи

1. Створіть паппу lab10_lastname.
2. Реалізуйте приклади коду, наведені у “Тема 5. Сеанси та сесії в PHP” Навчального посібника з дисципліни «Програмування мовою PHP» до підпункту “5.3.4 Встановлення масиву cookies і його читання” включно.
3. Створіть ще одну сторінку secret_other.php, на яку дозвольте перехід авторизованому користувачу із сторінки secret_info.php, у протилежному випадку додайте перехід на сторінку authorize.php. На кожній сторінці даної лабораторної роботи передбачте вивід імені користувача: “Ви увійшли як користувач <ім'я користувача>”, або “Ви увійшли як гість”, якщо ви не авторизувались.
4. Із сторінки secret_other.php передбачте перехід на сторінки secret_info.php (дані сесії мають зберегтися) і на сторінку index.php (головну сторінку лабораторної роботи 10, дані сесії мають видалитися, потрібно заново авторизуватися).
5. На головній сторінці лабораторної роботи index.php додайте посилання на файл із формою для реєстрації нових користувачів. У базі даних створіть таблицю внесення записів зареєстрованих користувачів user_for_session (обов'язковими полями є: логін (має бути унікальним) та пароль, інші поля на свій розсуд). Розробіть (скопіюйте із лабораторної роботи 8) PHP-сценарій обробки форми реєстрації і запису даних про користувача в таблицю користувачів. Занесіть мінімум трьох користувачів (себе і двох одногрупників, наступних по списку, в якості логіну вводьте прізвище на

латині). Напишіть php-код, який під формою реєстрації виведе у вигляді таблиці всі дані (обов'язково логін і пароль) зареєстрованих користувачів із таблиці user_for_session.

6. Модифікувати код перевірки коректності авторизованого користувача:

```
if (!($_SESSION['login']=="pit" &&  $_SESSION['passwd']=="123")) {
Header("Location: authorize.php");
}
```

таким чином, щоб значення логіна і пароля користувача перевірялися не на відповідність "pit" і"123", а щоб користувач із відповідним логіном і паролем був зареєстрованим (тобто його логін і пароль мають бути внесеними у таблицю бази даних user_for_session, див. наведений приклад у розділі “5.4 Безпека при роботі із сеансами підручника”) Навчального посібника з дисципліни «Програмування мовою PHP»

7. Здайте і захистіть лабораторну роботу.

10.3. Контрольні питання

1. В чому полягає потреба авторизації?
2. Які є способи авторизації і які можливі проблеми при їх використанні?
3. Як працює механізм сесій?
4. Які параметри визначають сесію?
5. Які функції використовуються при роботі із сесіями в PHP?
6. Для чого існує масив масиву $_SESSION?
7. Що таке Cookies?
8. Як задаються Cookies в PHP?
9. Як прочитати значення Cookies?
10. Які заходи безпеки потрібно вживати при роботі із сесіями?

Лабораторна робота №11. Підсумковий проєкт. PHP. СУБД. Створення сайту Інтернет-магазину.

Термін виконання: 6 год.

Мета: навчатися комплексному використанню вмінь і навичок, отриманих під час виконання попередніх лабораторних робіт.

Завдання: Створити сайт Інтернет-магазину для трьох категорій користувачів: покупця, продавця і гостьовий вхід відповідно до поданих нижче вимог і тематики згідно варіанту (номер варіанту - це номер у списку групи, тематика наведена в кінці роботи).

Обов'язкове використання: PHP, СУБД (MySql, Postgresql тощо).

Увага! У файл config.php в межах блоку php має бути доданий код про вивід часу останньої зміни (див. Лабораторну роботу 1) . Файл config.php повинен бути приєднаним до всіх файлів даної роботи (як і у попередніх роботах) і бути єдиним, даний код повинен виводити на кожну сторінку час і дату її останньої зміни.

11. 1 Хід роботи

1. На вебхостингу створіть папку <lab11_lastname>, де lastname - ваше прізвищ, в ній зберігайте всі файли. Назви усіх файлів повинні починатися з Вашого прізвища на латині (окрім назв файлів index.php і config.php). На файл index.php зробіть посилання із стартової сторінки вебхостингу, де розміщені посилання на інші лабораторні роботи.

2. Стартова сторінка Інтернет-магазину (index.php): заголовок «Інтернет-магазин <Ваше ПІБ> <Тема відповіднор до варіанту>», п'ять кнопок (або посилань): **Реєстрація, Вхід, Забули пароль, Зареєстровані користувачі, Склад.**

3. Стартова сторінка являє собою гостьову сторінку з інформацією «Ви увійшли як гість» та дає можливість тільки перегляду деякої інформації про чотири випадкових товари із таблиці **lastname_storage** бази даних: найменування, зображення і ціна за одиницю товару. Для цього у базі даних створіть таблицю **lastname_storage**, у яку додайте поля для внесення наступних даних про товар: найменування, назва файлу із зображенням товару, ціна за одиницю товару (шт., кг. і т.д.), наявна кількість, дата додавання товару в таблицю **lastname_storage**. У таблицю **lastname_storage** додайте мінімум шість

найменувань товарів відповідно до варіанту. На вебсторінку перелік товарів виводьте у дві колонки.

4. При кліку на **Реєстрація** відкривається форма для введення і запису в таблицю **lastname_users** бази даних наступних даних: **Ім'я, Прізвище на латині, логін** (у ролі логіна вказати e-mail на домені pnu.edu.ua)**, пароль, повторити пароль** і вибір однієї із двох категорій користувачів: **“Продавець”, “Покупець”.**

5. Після вдалого проходженні перевірки:

   a. Ім'я, Прізвище на відповідність шаблону: всі літери на латині, слова починаються з великої літери;

   b. логіна на відповідність шаблону e-mail та його приналежності домену pnu.edu.ua, а також його відсутності серед уже зареєстрованих користувачів відповідної категорії;

   c. тотожності введених паролів, пароль має містити мінімум 6 символів, серед них мінімум одну латинську літеру, одну цифру і один із спеціальних символів, таких як _ - . @ # $ % ^ & ! ? *;

передані дані записуються у таблицю **lastname_users** бази даних і виводиться повідомлення «Реєстрація пройшла успішно» чи «Помилка реєстрації, спробуйте ще раз» при невдалому проходженні перевірки, при цьому потрібно виділити текст червоним кольором у полях, в яких дані введені не відповідно до вимог і вивести підказки щодо коректності заповнення полів. В обох випадках користувач має повернутися на стартову сторінку Інтернет-магазину. Передбачити, щоб користувач із тим самим email міг бути зареєстрований як Покупець, так і Продавець.

6. З допомогою створеного інтерфейсу зареєструйте мінімум 4-х одногрупників (двох зверху і двох знизу із списку групи).

7. При кліку на “**Зареєстровані користувачі”** потрібно вивести на веб-сторінку у вигляді таблиці весь вміст таблиці **lastname_users** із внесеними даними всіх зареєстрованих користувачів, в тому числі і їх паролі**.**

8. При кліку на “**Забули пароль”** повинна відкритися сторінка для введення e-mail і чекбокси для множинного вибору категорій користувача і кнопки “**Нагадати**”. Після кліку на кнопку “**Нагадати**” на сторінку виводяться

паролі (пароль) із вказанням категорій користувача, які відповідають введеному e-mail і вибраним категоріям користувача.

9. При кліку на **Вхід** відкривається форма для введення логіна, пароля і радіокнопок для вибору категорії користувача.

Якщо введений логін і пароль не знайдений серед зареєстрованих користувачів відповідної категорії, видається повідомлення “Введено невірний логін чи пароль”, користувач залишається на стартовій сторінці.

10. Якщо введений логін і пароль відповідає зареєстрованому користувачу відповідної категорії, формується персональне повідомлення із використанням сесій з наступною інформацією: “Ви увійшли як покупець (продавець) під іменем <ім'я та прізвище користувача>” і виведеним у дві колонки переліком всіх товарів занесених у таблицю **lastname_storage** із наступною інформацією: найменування, зображення і ціна за одиницю товару. При кліку на зображення чи назву товару повинна відкритися сторінка, на яку виведіть з використанням сесій повідомлення “Ви увійшли як покупець (продавець) під іменем <ім'я та прізвище користувача>” та всю інформацію про обраний товар, а також додайте лічильник вибору кількості товару і, в залежності від категорії користувача, кнопку «**Додати в кошик**» (для покупців) чи «**Поповнити склад**» (для продавців), після кліку на які відповідно зменшити (без переходу в мінус) чи збільшити наявну кількість товару на вказану кількість і вивести змінені дані з таблиці **lastname_storage**. Також додайте посилання “**Продовжити покупки**” для переходу на попередню сторінку із переліком всіх товарів, при переході на яку дані авторизованого користувача зберігаються і також виводяться у вигляді відповідного повідомлення. Окрім того, при кліку на кнопку «**Додати в кошик**» мають акумулюватися дані про додані в кошик відповідним покупцем товари: ідентифікатор товару та кількість.

11. На сторінці покупця додайте кнопку “**Кошик**”, при кліку на яку виведіть інформацію про додані в кошик товари відповідного покупця: найменування, зображення, ціна за одиницю, обрана кількість та загальна вартість кожного найменування товару, а також загальна вартість всіх товарів, доданих у кошик. Передбачити можливість видалення товарів із кошика без їх повернення на склад.

12. На сторінці продавця додайте кнопки “**Додати новий товар**” і “**Аудит**”. При кліку на кнопку “**Додати новий товар**” створіть форму для внесення у таблицю **lastname_storage** даних про нове найменування товару, в

тому числі передбачити завантаження на сервер файлу із зображенням товару. При кліку на кнопку “**Аудит**” виведіть загальну вартість товарів, яка наявна на складі.

13. Кнопку **Вихід** додайте на всі сторінки, окрім стартової. При кліку на кнопку **Вихід** передбачити перехід на стартову сторінку Інтернет-магазину і вихід із облікового запису.

Варіанти тематики:

1. Транспорт
2. Птахи
3. Їжа
4. Домашні тварини
5. Гриби
6. Транспорт
7. Дерева
8. Фрукти
9. Овочі
10. Ягоди
11. Косметика
12. Кондитерські вироби
13. Двоколісний транспорт
14. Меблі
15. Комп’ютерна техніка
16. Жіночий одяг
17. Чоловічий одяг
18. Дитячий одяг
19. Взуття
20. Солодощі
21. Спортивні аксесуари
22. Музичні інструменти
23. Транспорт
24. Кухонна техніка
25. Побутова техніка
26. Автомобілі
27. Комп’ютерна техніка
28. Птахи
29. Їжа
30. Домашні тварини

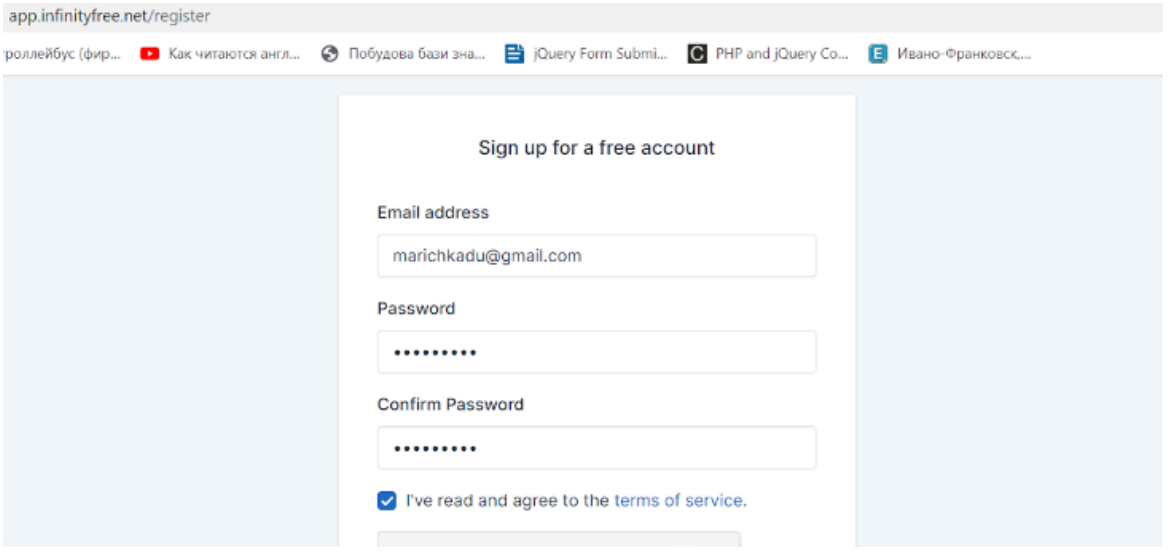
Виконану роботу потрібно розмістити на вебхостингу та заархівувати, а також оформити звіт, в якому вставте всі завдання даної роботи і скріни фінального вигляду всіх вебсторінок, а також скрін вмісту таблиць **lastname_storage** і **lastname_users** бази даних із внесеними товарами і користувачами. Посилання на стартову сторінку лабораторної роботи, архів і файл із звітом здайте в Google Classroom.

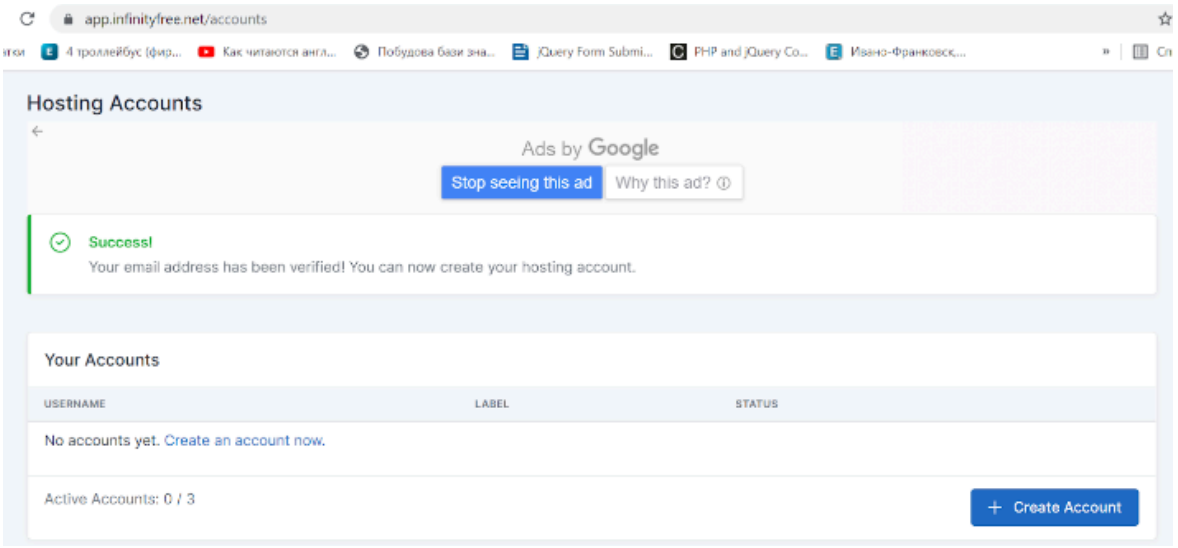
Список літератури

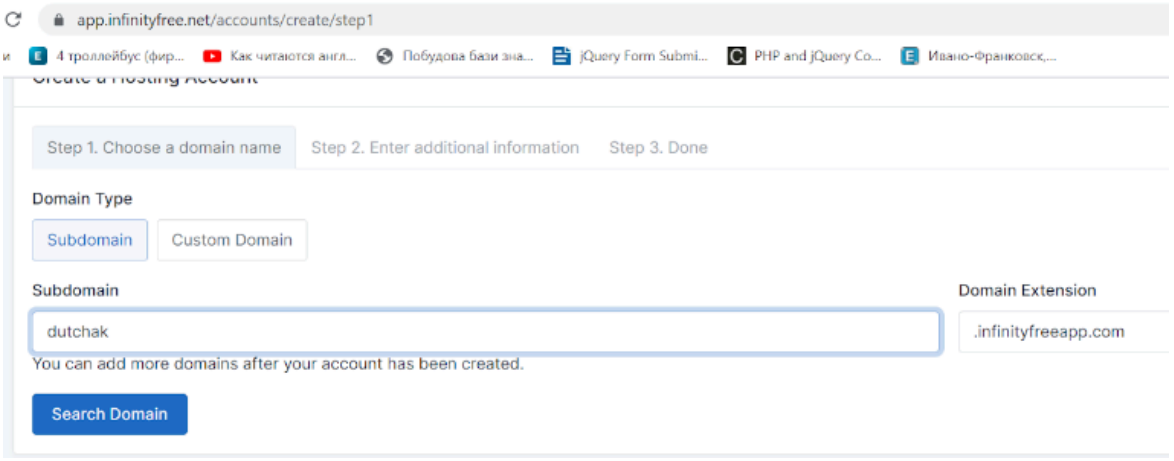
1. PHP Manual
2. Передача значення змінних через посилання
3. Уроки SQL
4. Навчальний посібник з дисципліни «Програмування мовою PHP»

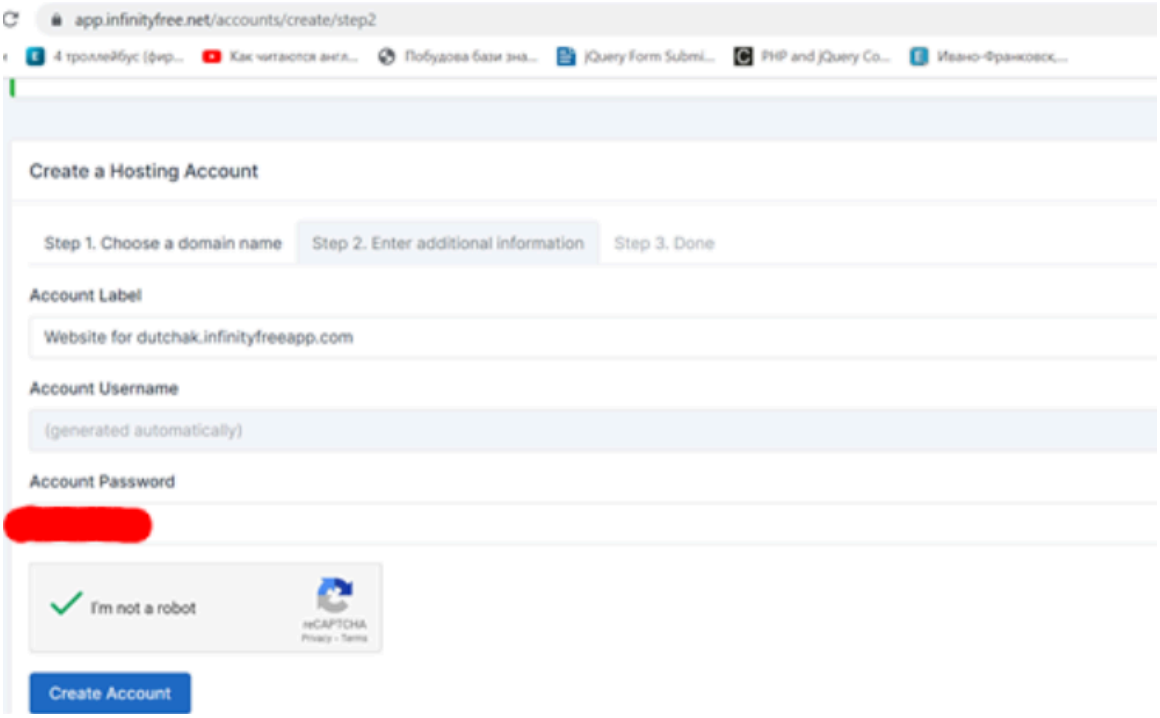
Додатки. Реєстрація на вебхостингу та налаштування FTP-з'єднання для роботи з ним за допомогою файлових менеджерів і редактора вихідного коду.

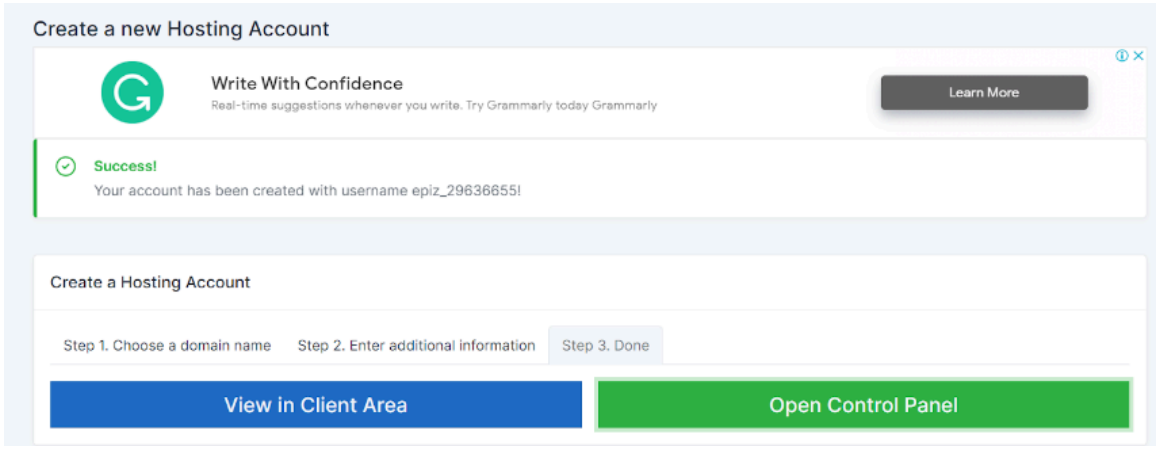
Додаток А. Етапи реєстрації облікового на вебхостингу https://infinityfree.net/:

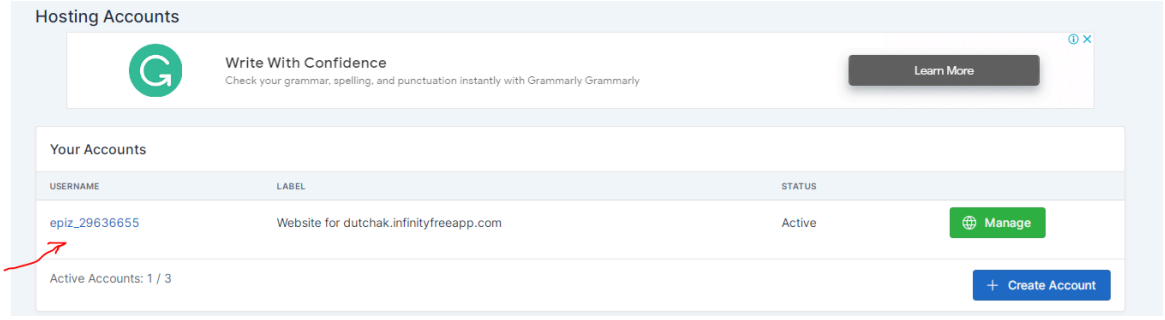


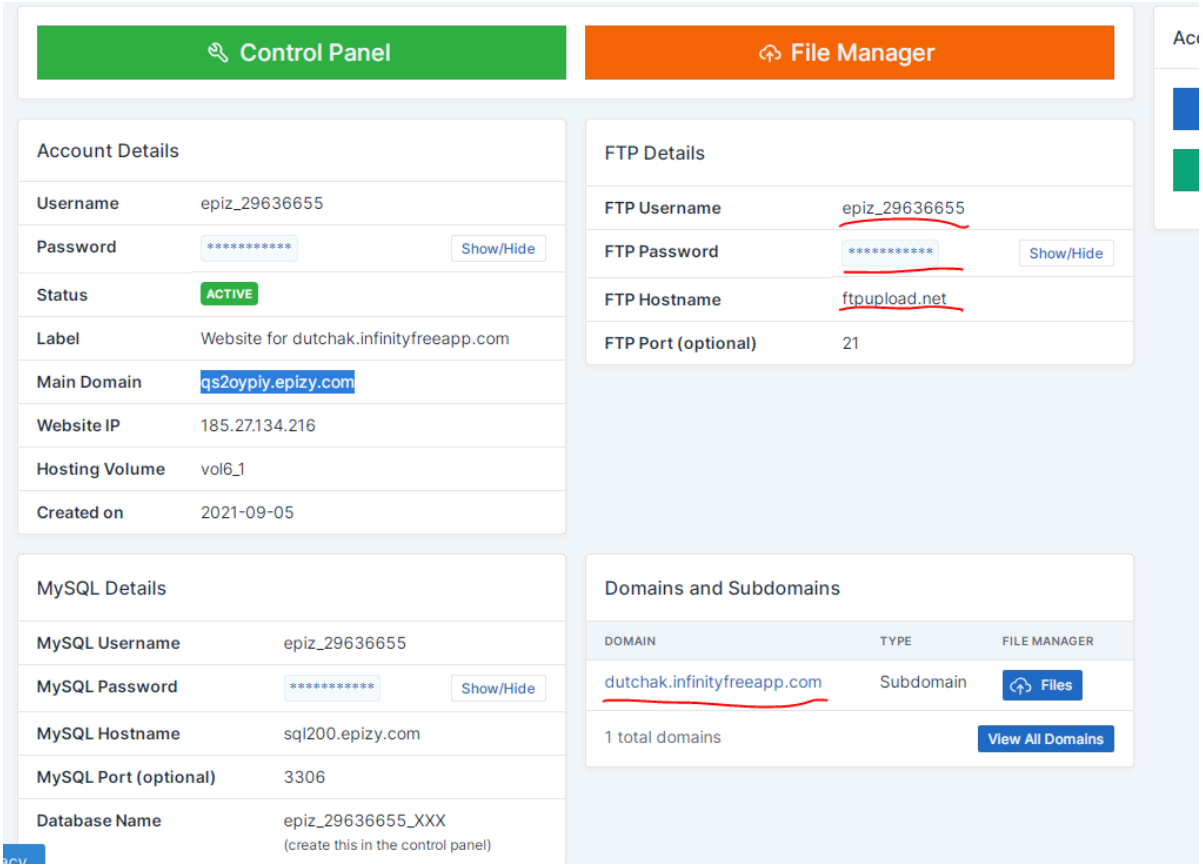




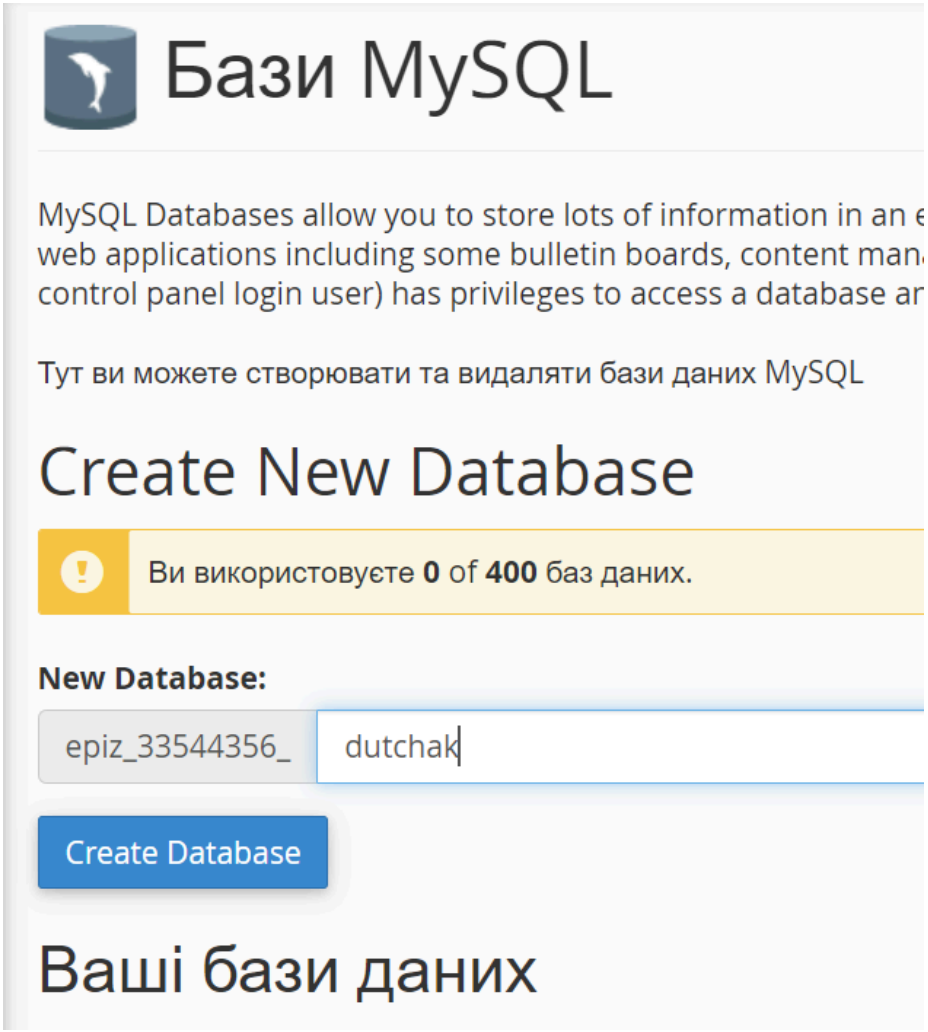

app.infinityfree.net/accounts/create/step2
Create a Hosting Account
Step 1. Choose a domain name
Step 2. Enter additional information
Step 3. Done
Account Label
Website for dutchak.infinityfreeapp.com
Account Username
(generated automatically)
Account Password
I'm not a robot
reCAPTCHA
Create Account


Create a new Hosting Account
Write With Confidence
Real-time suggestions whenever you write. Try Grammarly today Grammarly
Learn More
Success!
Your account has been created with username epiz_29636655!
Create a Hosting Account
Step 1. Choose a domain name
Step 2. Enter additional information
Step 3. Done
View in Client Area
Open Control Panel

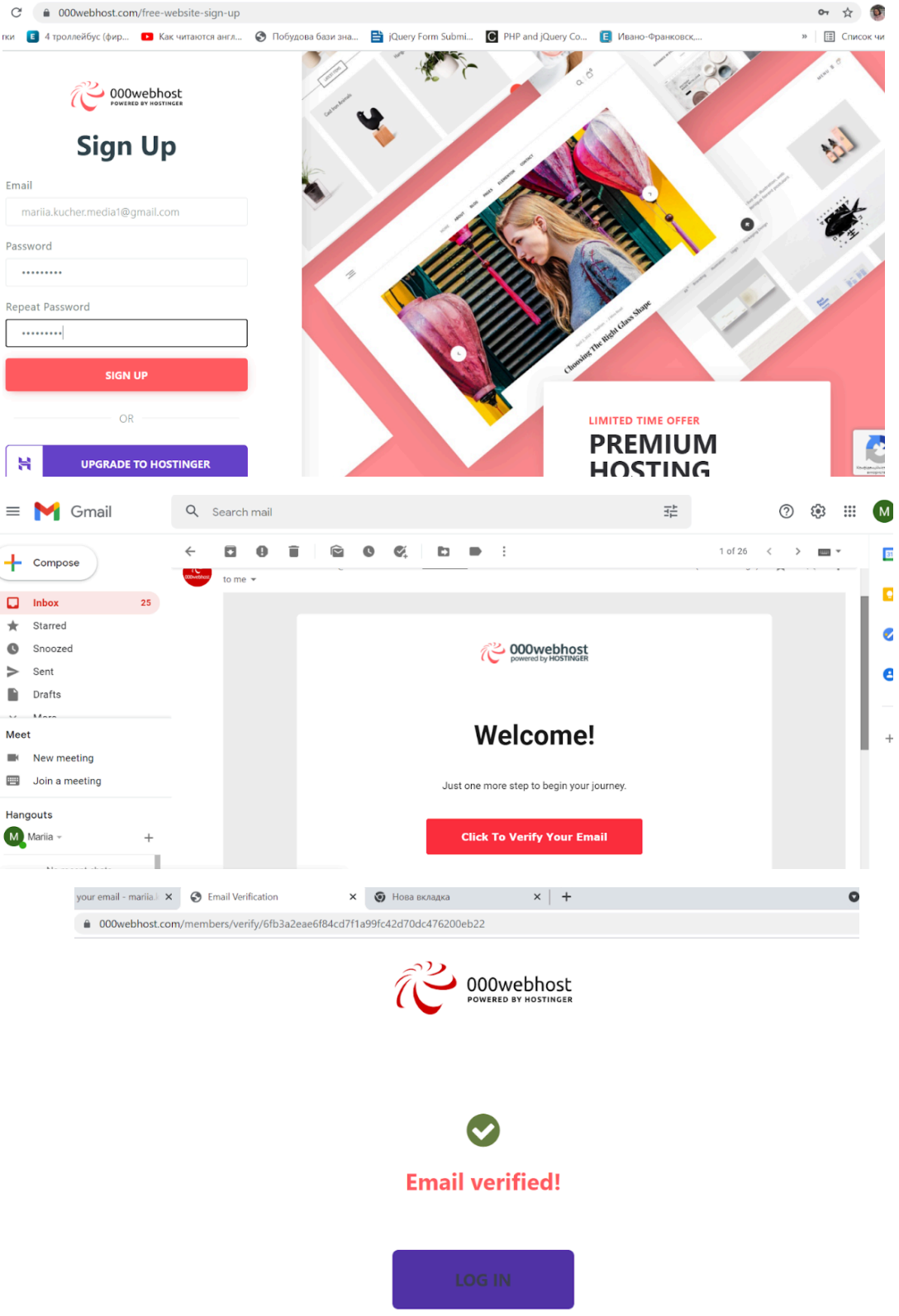

Hosting Accounts
Write With Confidence
Check your grammar, spelling, and punctuation instantly with Grammarly Grammarly
Learn More
Your Accounts
USERNAME
LABEL
STATUS
epiz_29636655
Website for dutchak.infinityfreeapp.com
Active
Manage
Active Accounts: 1 / 3
Create Account

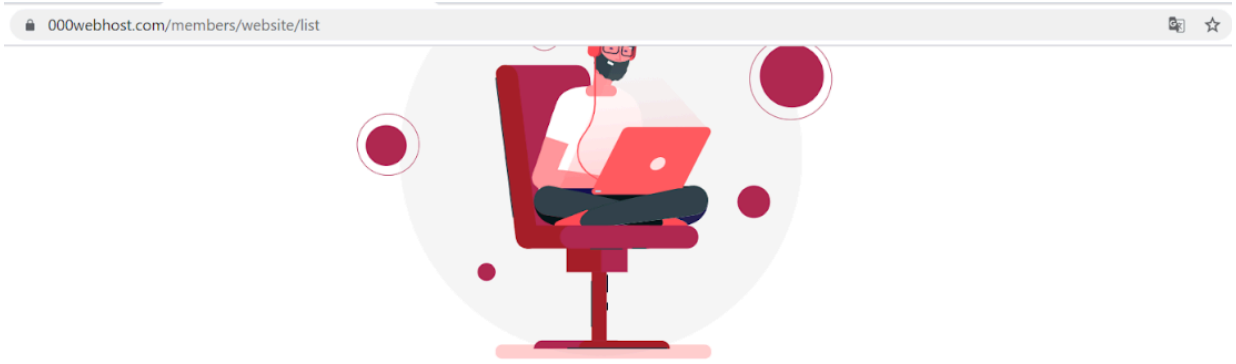


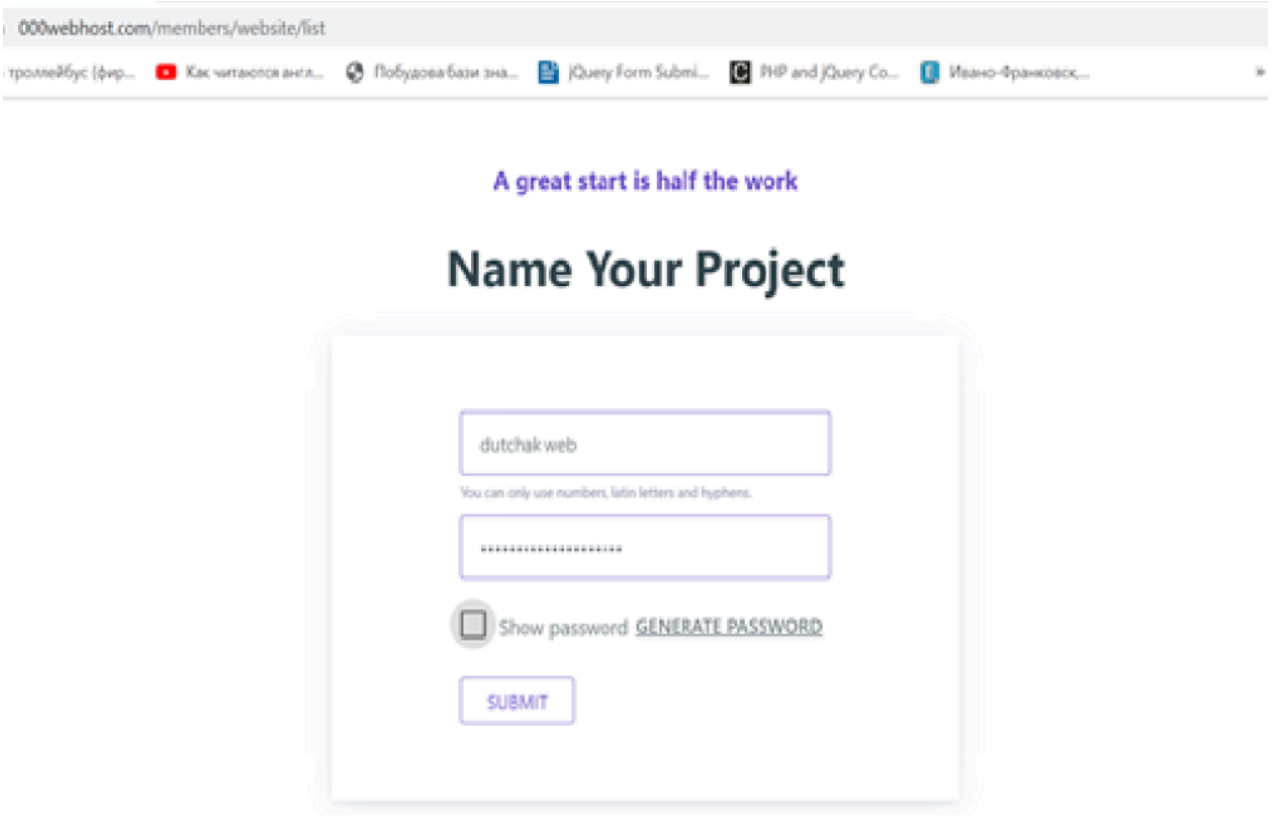
Для створення бази даних, перейдіть у Control panel -> Бази MYSQL, далі:



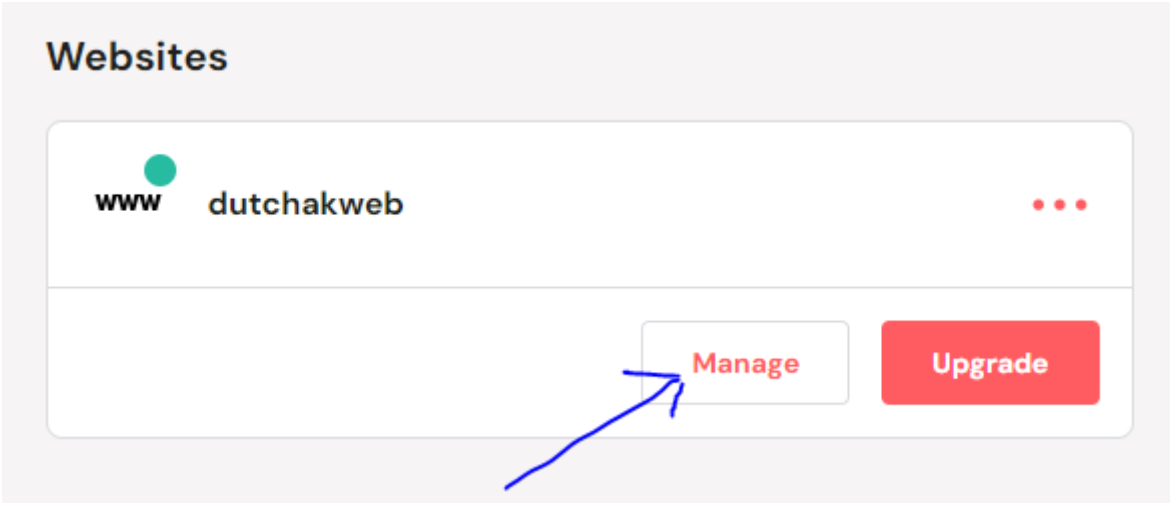
Додаток Б. Етапи реєстрації на вебхостингу https://www.000webhost.com/

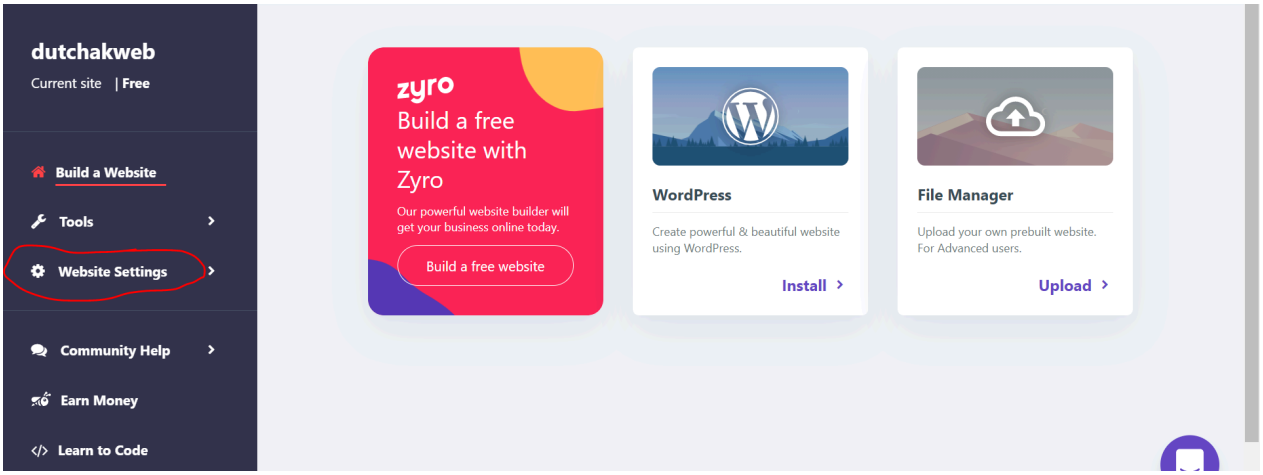


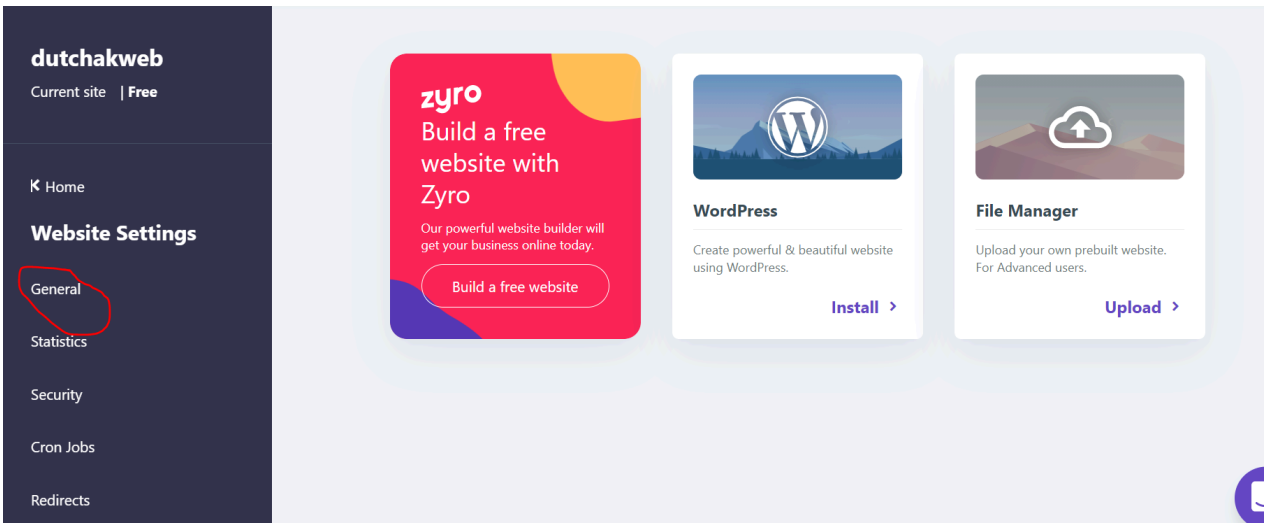


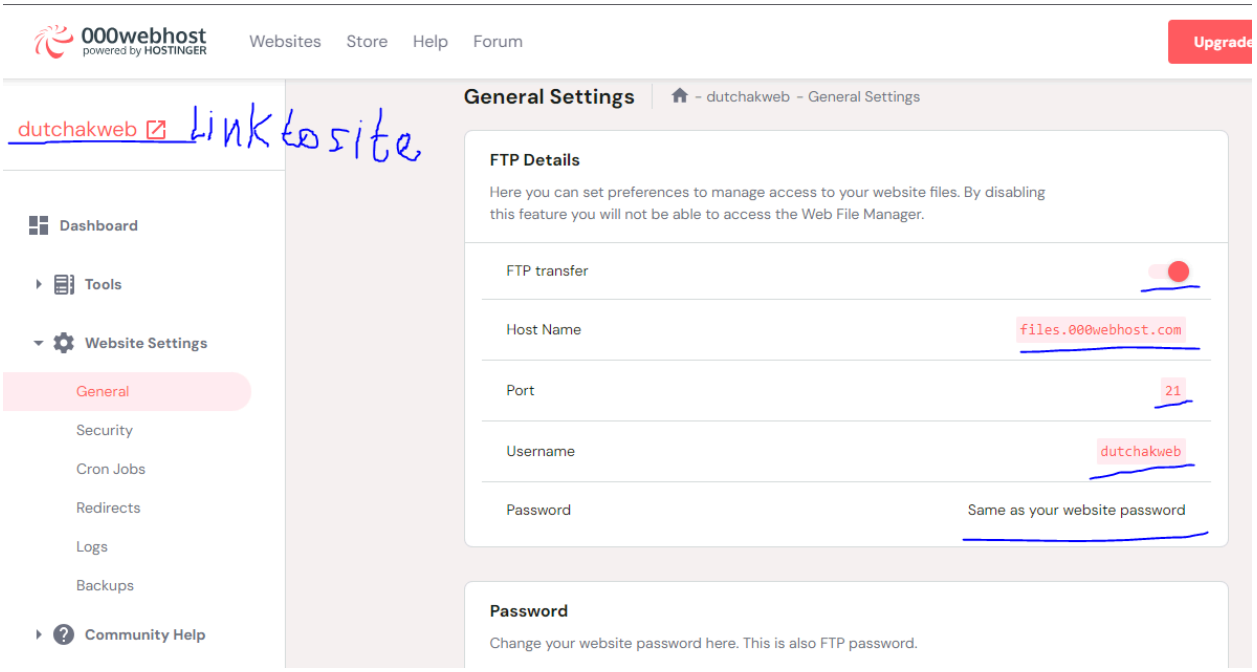


Десь на цьому етапі пропонує вибір типу сайту, то обирайте перше, щось типу “Learn development web”


Websites
www
dutchakweb
Manage
Upgrade

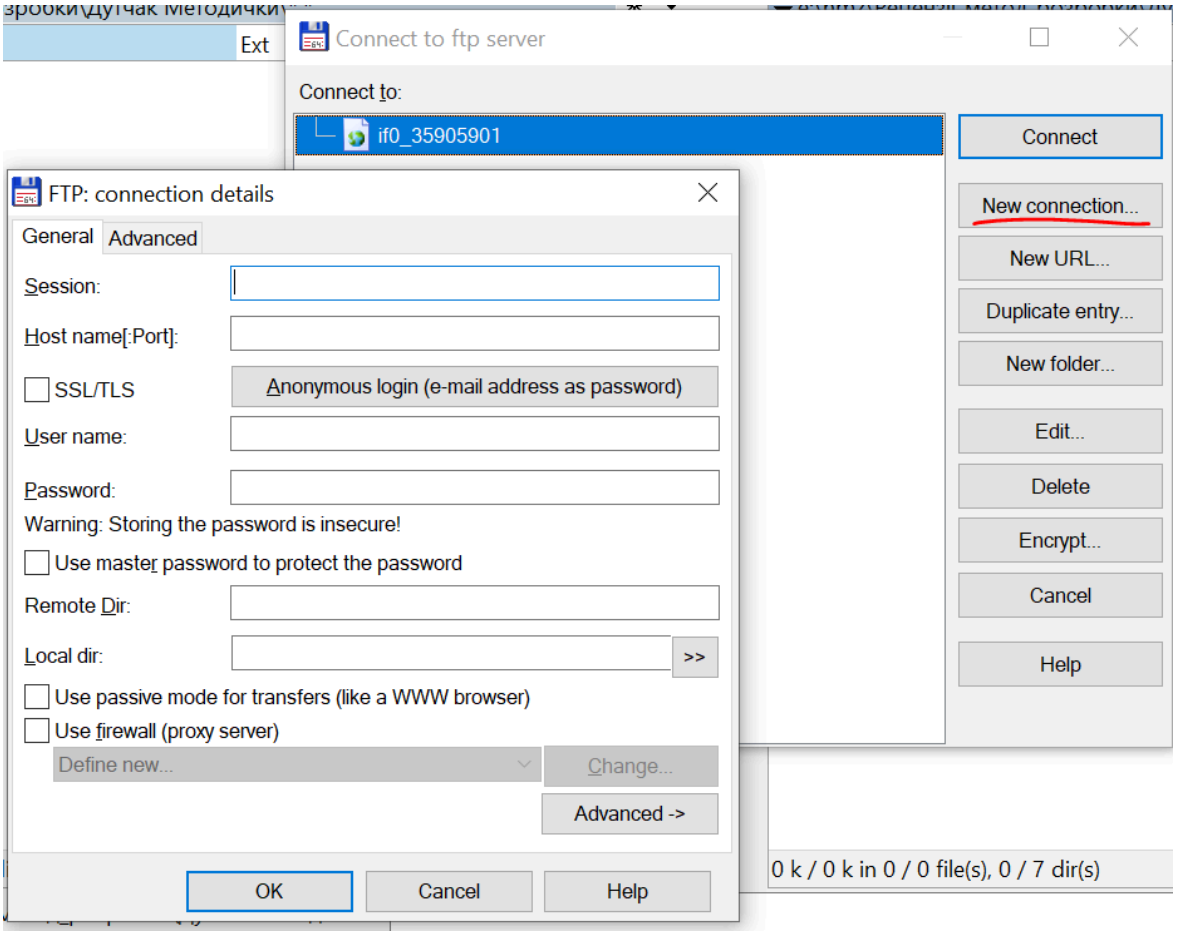

dutchakweb
Current site | Free
Build a Website
Tools
Website Settings
Community Help
Earn Money
Learn to Code
zyro
Build a free website with Zyro
Our powerful website builder will get your business online today.
Build a free website
WordPress
Create powerful & beautiful website using WordPress.
Install
File Manager
Upload your own prebuilt website. For Advanced users.
Upload


dutchakweb
Current site | Free
Home
Website Settings
General
Statistics
Security
Cron Jobs
Redirects
zyro
Build a free website with Zyro
Our powerful website builder will get your business online today.
Build a free website
WordPress
Create powerful & beautiful website using WordPress.
Install
File Manager
Upload your own prebuilt website. For Advanced users.
Upload

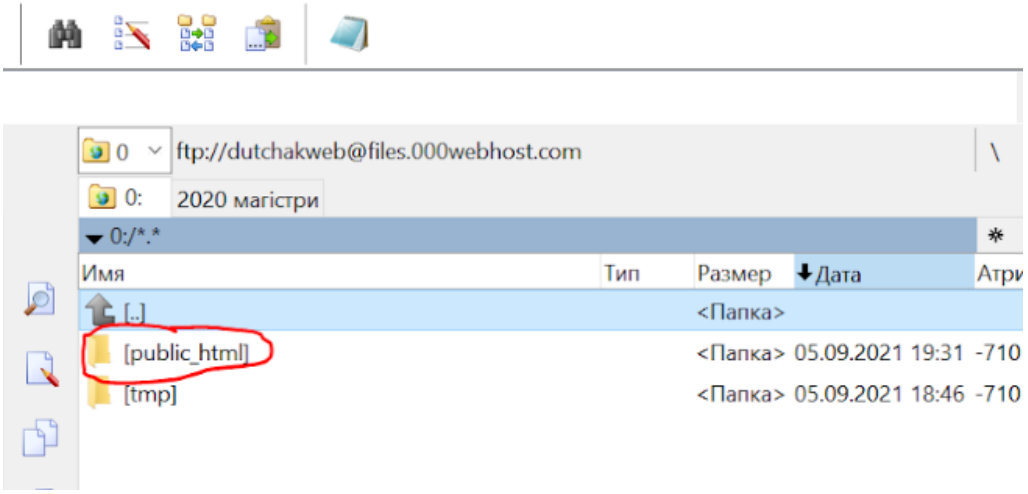

000webhost
powered by HOSTINGER
Websites
Store
Help
Forum
Upgrade
dutchakweb
link to site
Dashboard
Tools
Website Settings
General
Security
Cron Jobs
Redirects
Logs
Backups
Community Help
General Settings
dutchakweb - General Settings
FTP Details
Here you can set preferences to manage access to your website files. By disabling this feature you will not be able to access the Web File Manager.
FTP transfer
Host Name
files.000webhost.com
Port
21
Username
dutchakweb
Password
Same as your website password
Password
Change your website password here. This is also FTP password.

Додаток В. Налаштування FTP-з’єднання у програмах Total Commander, Файловий провідник та Far Manager

У програмі Total Commander заходите в меню програми «Мережа/З’єднання з ftp-сервером» (Ctrl+F), дальше «Нове з'єднання» і ввести налаштування сервера (із закладки “Website Settings/General” на https://000webhost.com, Account Details на https://infinityfree.net).



Далі обираємо щойно створене з’єднання і клікаємо на «З’єднати». Після з’єднання із ftp-сервером, перейти у папку public_html і туди скопіювати необхідні web-файл. Файл index.html (index.php тощо) є файлом стартової сторінки сайти.



Файловий провідник

Якщо не вдалося з'єднатися із FTP через Total Commander, можете приєднатися через програму Файловий провідник. У адресному рядку потрібно вказати:

ftp://user_name@host_name

Наприклад:

ftp://epiz_33544356@ftpupload.net

ftp://dutchakweb@files.000webhost.com

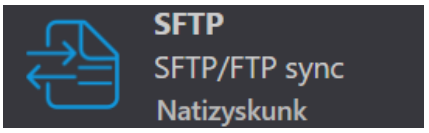
Налаштуванн FTP-єднання у файловому менеджері Far Manager

У файловому менеджері Far Manager: Alt+F1 (або F2) -> Net Box -> Shift+F4 і ввести налаштування сервера (із закладки “Website Settings/General” на https://000webhost.com, Account Details на https://infinityfree.net).

Щоб з'єднатися Alt+F1 (або F2) -> Net Box -> далі обираєте потрібну назву.

Far Manager також дозволяє напряму редагувати файли, розміщені на віддаленому сервері, для редагування клікаєте F4.

Додаток Г. Налаштувати VS Code для роботи із FTP-сервером хостингу.

Запустіть програму VS Code. Викличте меню додатків  Extensions або Ctrl+Shift+X і встановіть розширення 

. (якщо встановлений, ще раз не потрібно встановлювати)

Клікніть зліва вверху на  Explorer або Ctrl+Shift+E, далі Open Folder відкрийте папку дисципліни. Після того, як вона з'явиться у Вашому робочому просторі, виділіть її, далі клікнути Ctrl+Shift+P і ввести sftp: config.

У файлі налаштувань sftp.json внести свої дані:

```
{
   "name": "My Server",
   "host": "host_name",
   "protocol": "ftp",
   "port": 21,
   "username": "your Username",
   "password": "your Password",
   "remotePath": "/",
   "uploadOnSave": true,
   "useTempFile": true,
   "openSsh": true,
   "downloadOnOpen": true,
   "ignore": []
}
```

Якщо Ви працюєте не за власним ПК, пароль можна не вносити, тоді його потрібно буде вводити в діалогове вікно. Або вказати пароль, а в кінці роботи видалити.

Збережіть і закрийте файл sftp.json.

Якщо все зроблено вірно, зліва появиться піктограма SFTP  , перехід на яку дозволяє переглядати і редагувати файли на Вашому FTP-сервері. Відредагуйте будь-який файл із Вашого хостингу, після чого кореневий каталог

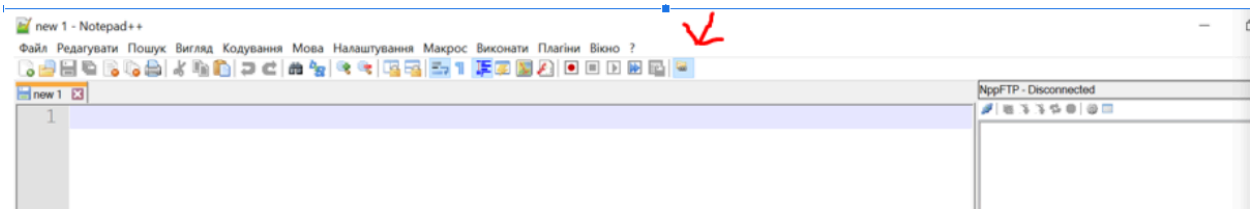
Вашого хостингу і файл, який Ви редагували, появиться в локальній папці дисципліни.

Якщо редагування файлу недоступне, у контекстному меню вибрати Edit in local. Зберегти зміни.

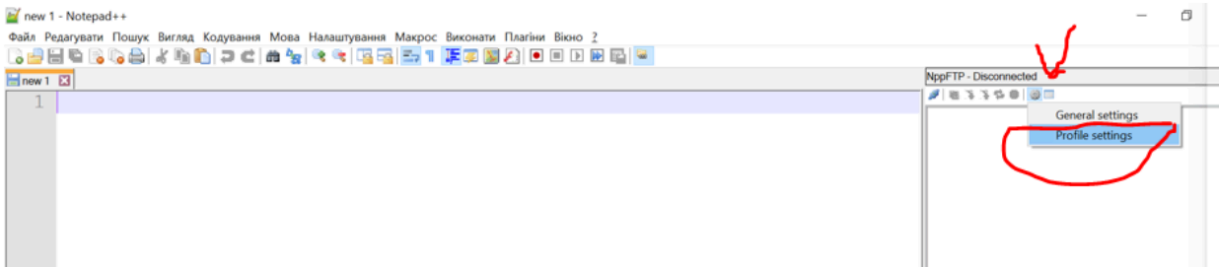
Щоб в локальній папці дисципліні появились всі файли і папки хостингу, їх потрібно скопіювати з хостингу (наприклад, з допомогою ТС) в кореневий каталог хостингу локальної папки. Тоді ці папки будуть повністю аналогічні і зміни в одній з них будуть автоматично вноситися в іншу. Крім того, Ви будете мати змогу контролювати для них версії і відправляти на віддалений репозиторій.

Додаток Д. Налаштування ftp-з'єднання за допомогою програми Notepad++

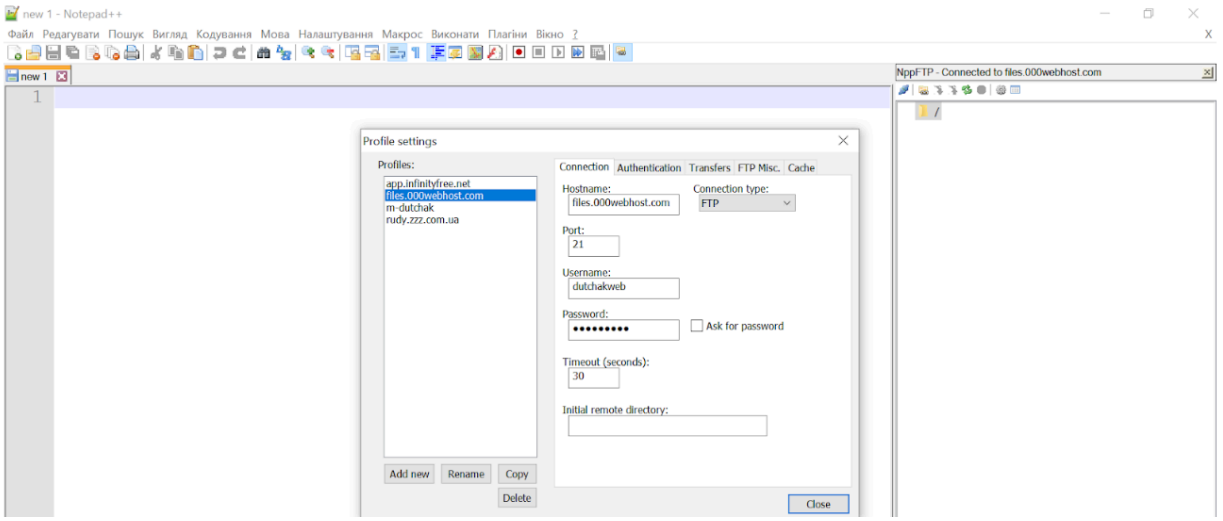
Запускаємо Notepad++ і в панелі інструментів клікаємо мишкою на значок "Show NppFTP Window" (див наступні скріни). Якщо відсутня дана можливість, потрібно завантажити плагін NppFTPl (файл NppFTP.dll із папки bin копіюємо в c:\Program Files\Notepad++\plugins\



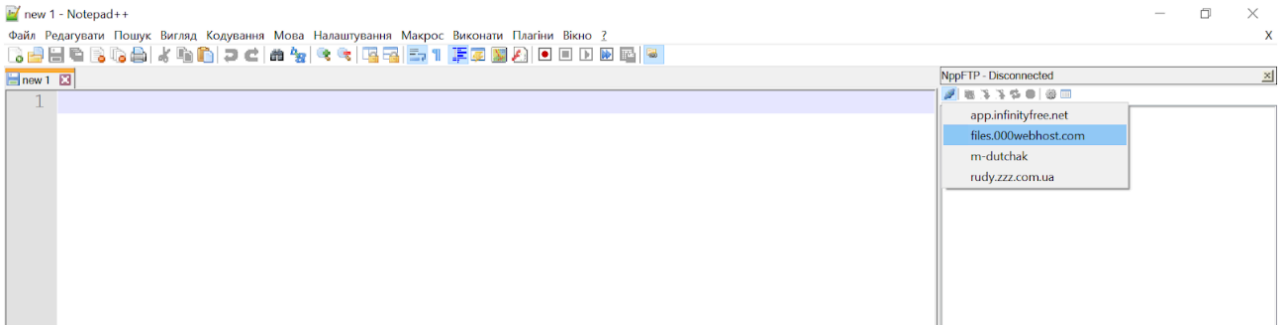
У вікні вбудованого FTP-клієнта тиснемо на "шестерю" *("Setting")* і вибираємо *"Profile settings"("Налаштування профілю").*



У вікні профілю заповнюємо стандартні дані для ftp-з'єднання хостингу:

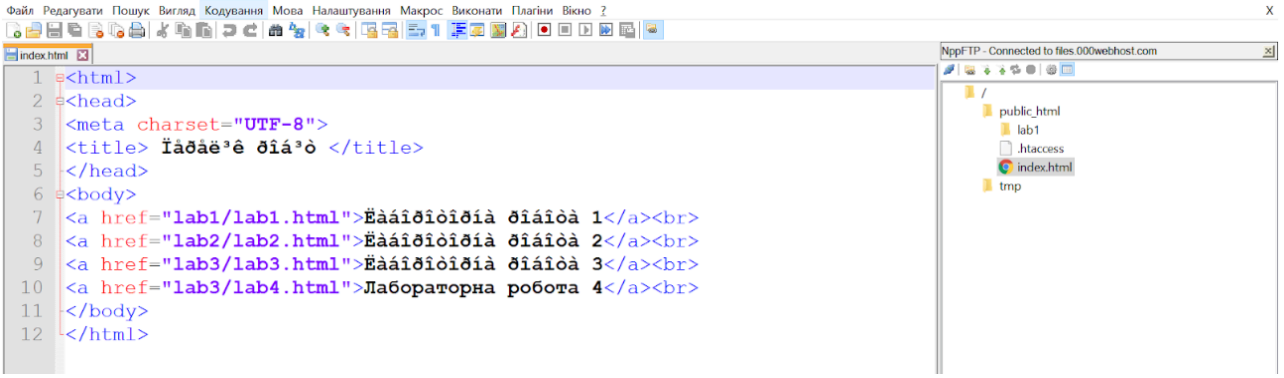


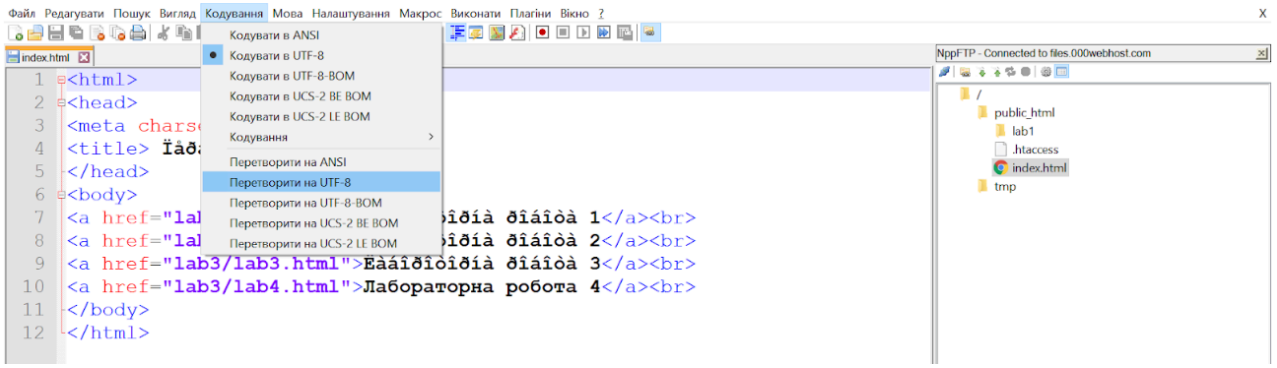
Потім обираємо створене з’єднання і переходимо на хостинг через ftp-з’єднання:

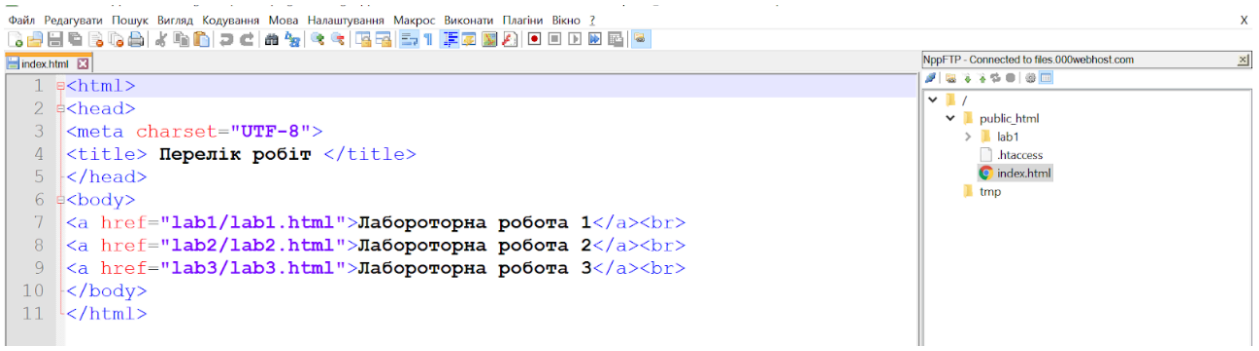


Дальше знаходимо скопійовані на вебхостинг html-файли і продовжуємо виконувати лабораторні роботи. Щоб кирилиця коректно відображалася, перед копіюванням на вебхостинг конвертуйте web-файли і задати кодування UTF-8:

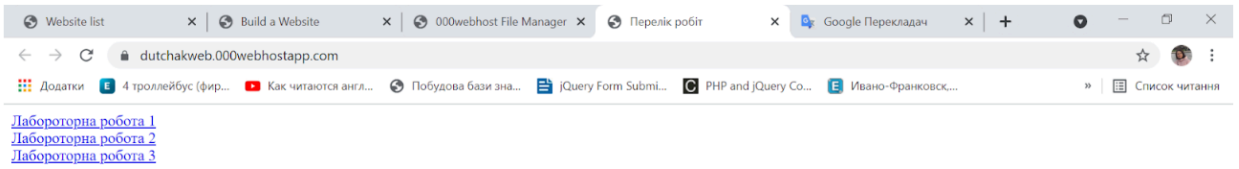
Кодування-Перетворити на UTF-8.







Для перегляду файлу index.html через браузер, в адресному рядку введіть Website Name, яке вказане в налаштуванні хостингу в закладці “Website Settings/General”.



Код web-документу можна переглядати викликавши контекстне меню сторінки, дальше «Переглянути джерело сторінки» (Ctrl+U)