Міністерство освіти і науки України

Тернопільський національний економічний університет

Факультет комп'ютерних інформаційних технологій

***Кафедра комп’ютерних наук***

**Лабораторна робота №1**

**з дисципліни** «WEB - програмування»

**на тему**: Створення простих сценаріїв на мові PHP

**Виконав:**

студент групи ІПЗ-32

Фатюк В.І.

**Перевірила:**

викл. Дарморост І.А.

**Тернопіль – 2018**

**Варіант №28**

**Тема роботи:** Створення простих сценаріїв на мові PHP.

**Мета роботи:** Одержання практичних навиків розробки простих сценаріїв з використанням PHP.

**Теоретичні відомості**

РНР (створений у якості передобробника гіпертекстів — Hypertext Preprocessor) увібрав в себе всю простоту сценаріїв і міць засобів розробки серверної частини Web-додатків. Якщо взяти до уваги його простий C-подібний синтаксис, підтримку найрізноманітніших баз даних, об’єктно-орієнтовного підходу і можливість безкоштовного використання, то стане зрозуміла зростаюча популярність мови РНР.

### **Принципи обробки сценаріїв PHP**

Коли від клієнтського броузера надходить запит на одержання звичайної сторінки HTML, то Web-сервер виконує наступну просту послідовність дій:

1. Аналіз HTTP-запиту.

2. Пошук потрібного серверного ресурсу (сторінки).

3. Передача даних назад клієнту.

У випадку запиту сторінки, до складу якої входять PHP-сценарії, наведена вище послідовність дій ускладнюється. Web-сервер в цьому випадку повинен виконати аналіз файлу, до якого відбувається запит, а при виявленні операторів PHP викликати відповідний інтерпретатор. У трохи спрощеному вигляді послідовність дій, яка виконується, виглядає в такий спосіб.

1. Аналіз HTTP-запиту.

2. Пошук необхідного серверного ресурсу (сторінки).

3. Аналіз сценаріїв, виявлених на сторінці, шляхом виклику відповідного інтерпретатора.

4. Передача назад клієнтові результуючої сторінки.

Необхідно відзначити, що на сторінці, що передається клієнтському броузеру, вже не міститься ніяких сценаріїв. Їх місце зайняли дані, згенеровані в процесі їхньої інтерпретації. В загальному випадку такими даними може бути не тільки HTML-код, але й різні фрагменти на мовах клієнтських сценаріїв (JavaScript і т.д.).

Як і у будь-якій іншій мові, в мові РНР існують різні вбудовані типи даних.

* цілий
* тип з плаваючою крапкою
* логічний (можуть приймати одне з двох значень — істина (true) або хиба (false))
* строковий
* масиви
* класи
* тип NULL

Виконання роботи:

**Завдання 1.**

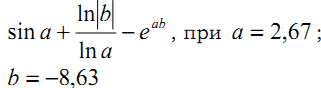


Рисунок 1.1: Завдання шо буде виконуватися

**<?php  
if** ($\_SERVER[**'REQUEST\_METHOD'**] == **'POST'** && **isset**($\_POST[**'first'**])){  
$a=$\_POST[**'first'**];  
$b=$\_POST[**'second'**];  
  
$result=*sin*($a) + (*log*(*abs*($b)) / *log*($a)) - ***M\_E***\**pow*($a\*$b);  
$result=*round*($result, 4);  
**echo "Отримані результати <b>a =</b>** $a **<b>x =</b>** $b **<b>y =</b>** $result**"**;  
}  
**?>**<**form action="../lab\_1/task\_1.php" method="post"**>  
<**p**>Значення а = : <**input type="text" name="first"** /></**p**>  
<**p**>Значення x = : <**input type="text" name="second"** /></**p**>  
<**p**><**input type="submit"** /></**p**>  
</**form**>

Рисунок 1.2: Виконання завдання мовою PHP

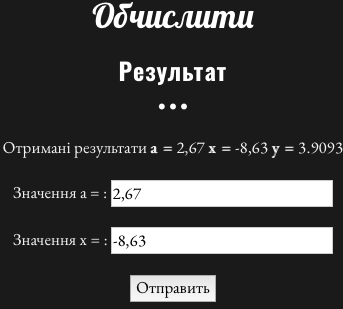


Рисунок 1.3: Результат роботи програми

**Завдання 2.**

Визначити, чи є задане натуральне число паліндромом, тобто таким, що його десятковий запис читається однаково зліва на право і справа наліво.

Рисунок 2.1: Завдання шо буде виконуватися

**<?php  
if** ($\_SERVER[**'REQUEST\_METHOD'**] == **'POST'** && **isset**($\_POST[**'first'**])){  
$number=$\_POST[**'first'**];  
  
$strrev = *strrev*($number);  
**if** ($number == $strrev) {  
 **echo 'Дане число <b>Паліндром</b>!!!!'**;  
} **else** {  
 **echo 'Дане число<b> Не</b> Паліндром!!!'**;  
}  
 }  
**?>**<**form action="../lab\_1/task\_2.php" method="post"**>  
<**p**>Введіть число: <**input type="text" name="first"** /></**p**>  
<**p**><**input type="submit"** /></**p**>  
</**form**>

Рисунок 2.2: Виконання завдання мовою PHP

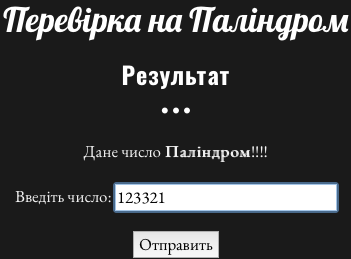


Рисунок 2.3: Результат роботи програми

**Завдання 3.**

Перевірити, чи є в одновимірному числовому масиві хоча б одна пара протилежних чисел.

Рисунок 3.1: Завдання шо буде виконуватися

**<?php**$numbers = [1, 2, 3, 4, -5, 6, -7, 8, 9, -9];  
  
$arr = *array\_filter*($numbers, **function**($int)**use**($numbers){  
 **return** *in\_array*(~--$int, $numbers);  
});  
  
*var\_dump*($arr);  
**?>**

Рисунок 3.2: Виконання завдання мовою PHP

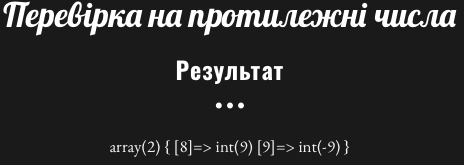


Рисунок 3.3: Результат роботи програми

**Завдання 4.**

Задано прямокутну матрицю E(5,4). Отримати нову матрицю, поділивши всі невід’ємні елементи на 5, а всі від’ємні елементи — на 10.

Рисунок 4.1: Завдання шо буде виконуватися

**<?php**$array = **array**();  
**for** ($i = 0; $i< 5; $i++) {  
 **for** ($j = 0; $j<4; $j++){  
 $array[$i][$j] = *rand*(-100, 100);  
 }  
}  
  
**echo "<br>"**;  
**echo "<br>"**;  
**echo "Початкова матриця:"**;  
**echo "<br>"**;  
renderArray($array);  
**echo "<br>"**;  
**echo "<br>"**;  
  
**for** ($i = 0; $i< 5; $i++) {  
 **for** ($j = 0; $j<4; $j++) {  
 **if** ($array[$i][$j] > 0 ) $array[$i][$j] = $array[$i][$j] / 5;  
 **if** ($array[$i][$j] < 0 ) $array[$i][$j] = $array[$i][$j] / 10;  
 }  
}  
**echo "<br>"**;  
**echo "Результат:"**;  
**echo "<br>"**;  
renderArray($array);  
**function** renderArray($array) {  
 **for** ($i = 0; $i< 5; $i++) {  
 **echo "<br>"**;  
 **for** ($j = 0; $j<4; $j++) {  
 **echo** $array[$i][$j];  
 **echo "&nbsp;&nbsp;&nbsp"**;  
 }  
 }  
}  
**?>**

Рисунок 4.2: Виконання завдання мовою PHP



Рисунок 4.3: Результат роботи програми

**Завдання 5.**

Обчислити значення змінної s=s1+s2, де s1 – сума елементів масиву A1(7), абсолютна величина яких не перевищує числа 7, а s2 – сума елементів масиву A2(7), що володіють тією ж властивістю.

Рисунок 5.1: Завдання шо буде виконуватися

**<?php**$a1 = **array**();  
**for** ($i=0; $i < 7; $i++) $a1[$i] = *rand*(0, 7);  
  
$a2 = **array**();  
**for** ($i=0; $i < 7; $i++) $a2[$i] = *rand*(0, 7);  
  
  
**function** getSumArrayElements($array) {  
 $suma = 0;  
 **for** ($i=0; $i < *count*($array); $i++) $suma += $array[$i];  
 **return** $suma;  
}  
  
$s1 = getSumArrayElements($a1) + getSumArrayElements($a2);  
  
**echo "<br>"**;  
**echo "Результат: "**;  
**echo** $s1;  
**?>**

Рисунок 5.2: Виконання завдання мовою PHP



Рисунок 5.3: Результат роботи програми

**Завдання 6.**

Дано текст з прописними латинськими літерами, за яким йде крапка. Надрукувати літери, що входять в текст чотири рази.

Рисунок 6.1: Завдання шо буде виконуватися

**<?php  
if** ($\_SERVER[**'REQUEST\_METHOD'**] == **'POST'** && **isset**($\_POST[**'first'**])) {  
 $text = $\_POST[**'first'**];  
  
 **function** str\_to\_array($str)  
 {  
 $arr = **array**();  
 **for** ($i = 0; $i < *strlen*($str); $i++) *array\_push*($arr, *strtolower*($str[$i]));  
 **return** $arr;  
 }  
  
  
 $arr = str\_to\_array($text);  
 $arr = *array\_diff*($arr, [**"."**, **","**, **" "**, **"?"**, **"!"**, **";"**, **"/"**]);  
  
 $arr = *array\_count\_values*($arr);  
 **echo "Повторень: "**;  
 **foreach** ($arr **as** $char => $amount) {  
 **if** ($amount == 4) {  
 **echo** $char . **"&nbsp;"**;  
 }  
  
 }  
}  
**?>**<**form action="../lab\_1/task\_6.php" method="post"**>  
<**p**>Текст = : <**input type="text" name="first"** /></**p**>  
<**p**><**input type="submit"** />  
</**p**>  
</**form**>

Рисунок 6.2: Виконання завдання мовою PHP

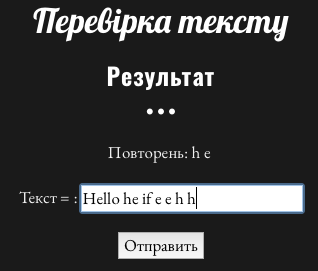


Рисунок 6.3: Результат роботи програми

Висновок: виконуючи данну лабораторну роботу я навчився створювати прості сценарії мовою програмування PHP, засвоїв такі базові поняття як змінні, функції, логічні та арифметичні оператори, тощо.