

Лабораторна робота №2

Тема: Використання функцій, рядків, масивів та форм в PHP

Хід роботи

Завдання 1. Робота з рядками (5 завдань)

- 1) Створіть 4 текстових поля з мітками *Текст*, *Знайти*, *Замінити* і *Результат*. Виконати можливість заміни символів і вивести результат.
- 2) Користувач вводить назви міст через пробіл. Переставте назви так, щоб назви були впорядковані за алфавітом.
- 3) Дан рядок, що містить повне ім'я файлу, наприклад:

D: \ WebServers \ home \ testsite \ www \ myfile.txt

Виділіть з цього рядка ім'я файлу без розширення

- 4) У двох рядках містяться дати виду День-Місяць-Рік (наприклад, 10-02-2015). Визначте кількість днів між датами.
- 5) Генератор паролів: Напишіть програму, яка генерує випадковий пароль заданої довжини. Пароль може містити великі та малі літери, цифри та спеціальні символи.

Перевірка пароля: Реалізуйте функцію, яка перевіряє, чи є введений користувачем пароль достатньо міцним. Вважайте пароль міцним, якщо він містить принаймні одну велику літеру, одну малу літеру, одну цифру та один спеціальний символ, і його довжина є не менше 8 символів.

У цьому завданні я створила веб-сторінку, яка надає кілька інтерактивних інструментів для виконання поширених операцій з рядками та даними в PHP. Кожен інструмент представлений окремою формою та обробляється відповідним PHP-кодом на цій же сторінці.

Основні виконані частини:

1. **Заміна символів у тексті**: Я створила форму, яка дозволяє користувачеві ввести текст, рядок для пошуку та рядок для заміни. Після відправки форми PHP-код використовує функцію `str_replace()` для виконання заміни та відображення результату:

```
<?php
if(isset($_POST['replace'])) {
    $original = $_POST['original_text'];
    $find = $_POST['find_text'];
    $replace = $_POST['replace_text'];
    $result = str_replace($find, $replace, $original);
    echo "<div class='result'>Результат: $result</div>";
}
?>
```

					ІПТР.420001.123-ЗЛ		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Мартинюк Ю.І.			Звіт з лабораторної роботи	Лім.	Арк.
Перевір.		Ковтун В.В.					1
Керівник						ФІКТ, гр. ІПЗ-23-3	
Н. контр.							
Затверд.							

Головний момент: Використання вбудованої функції `str_replace()` для простої та ефективної заміни всіх входжень підрядка в рядку.

2. **Сортування списку міст:** Я розробила форму для введення назв міст, розділених пробілами. Після відправки форми PHP-код використовує функцію `explode()` для розбиття рядка на масив міст, потім функцію `sort()` для сортування масиву за алфавітом і, нарешті, функцію `implode()` для об'єднання відсортованого масиву назад у рядок:

```
<?php
if(isset($_POST['sort_cities'])) {
    $cities = explode(' ', $_POST['cities']);
    sort($cities);
    $sorted = implode(' ', $cities);
    echo "<div class='result'>Відсортовані міста: $sorted</div>";
}
?>
```

Головний момент: Демонстрація роботи з рядками як з масивами через функції `explode()` та `implode()`, а також використання функції `sort()` для сортування масивів.

3. **Виділення імені файлу з повного шляху:** Я створила форму з попередньо заповненим прикладом повного шляху до файлу. Після відправки форми PHP-код використовує функцію `pathinfo()` з константою `PATHINFO_FILENAME` для отримання тільки імені файлу без розширення та шляху:

```
<?php
if(isset($_POST['extract_filename'])) {
    $path = $_POST['filepath'];
    $filename = pathinfo($path, PATHINFO_FILENAME);
    echo "<div class='result'>Ім'я файлу: $filename</div>";
}
?>
```

Головний момент: Використання потужної функції `pathinfo()` для розбору шляху до файлу та отримання його окремих компонентів.

4. **Обчислення різниці між двома датами:** Я розробила форму для введення двох дат у форматі ДД-ММ-РРРР. Після відправки форми PHP-код використовує класи `DateTime` та метод `diff()` для обчислення різниці між датами в днях. Також передбачена обробка помилок у разі неправильного формату введення дат:

```
<?php
if(isset($_POST['calc_dates'])) {
    $date1 = DateTime::createFromFormat('d-m-Y', $_POST['date1']);
    $date2 = DateTime::createFromFormat('d-m-Y', $_POST['date2']);

    if($date1 && $date2) {
        $diff = $date1->diff($date2);
        echo "<div class='result'>Різниця: " . abs($diff->days) . "
днів</div>";
    } else {
        echo "<div class='result error'>Помилка: Введіть дати у форматі
DD-MM-YYYY</div>";
    }
}
?>
```

					ІРТР.420001.123-ЗЛ	Арк.
						3
Змн.	Арк.	№ докум.	Підпис	Дата		

Головний момент: Демонстрація використання об'єктно-орієнтованих можливостей PHP для роботи з датами за допомогою класу DateTime та його методів.

5. **Генератор випадкових паролів:** Я створила форму для введення бажаної довжини пароля (за замовчуванням 12 символів, мінімум 8). Після відправки форми PHP-функція generatePassword() генерує випадковий пароль, що містить символи верхнього та нижнього регістру, цифри та спеціальні символи, а потім перемішує його:

```
<?php

function generatePassword($length) {
    // ... логіка генерації пароля ...
    return str_shuffle($password);
}

if(isset($_POST['generate'])) {
    // ... виклик функції та відображення результату ...
}
?>
```

Головний момент: Розробка власної функції для генерації випадкових рядків з урахуванням різних типів символів та використання функції str_shuffle() для забезпечення випадкового порядку.

6. **Перевірка надійності пароля:** Я створила форму для введення пароля та функцію checkPasswordStrength(), яка перевіряє пароль на відповідність кільком критеріям надійності (мінімальна довжина, наявність літер у верхньому та нижньому регістрі, цифр та спеціальних символів) за допомогою регулярних виразів:

```
<?php
function checkPasswordStrength($password) {
    // ... логіка перевірки за допомогою регулярних виразів ...
    return $isStrong;
}

if(isset($_POST['check'])) {
    // ... виклик функції та відображення результату ...
}
?>
```

Головний момент: Застосування регулярних виразів (preg_match()) для складного аналізу рядків та перевірки їх відповідності певним шаблонам.

1. Заміна символів

Текст:

Сьогодні буде тест

Знайти:

Сьогодні

Замінити:

завтра

Замінити

Результат: завтра буде тест

					ІПТР.420001.123-3Л	Арк.
						3
Змн.	Арк.	№ докум.	Підпис	Дата		

2. Сортуння міст

Введіть назви міст через пробіл:

Житомир Київ Вінниця Одеса

Сортувати

Відсортовані міста: Вінниця Житомир Київ Одеса

3. Виділення імені файлу

Повний шлях до файлу:

D:\WebServers\home\testsite\www\myfile.txt

Виділити

Ім'я файлу: myfile

4. Різниця між датами

Перша дата (DD-MM-YYYY):

27-04-2025

Друга дата (DD-MM-YYYY):

06-05-2025

Обчислити

Різниця: 9 днів

5. Генератор паролів

Довжина пароля:

12

Згенерувати

Згенерований пароль: cm-{kSeV7xh}

6. Перевірка надійності пароля

Введіть пароль:

1111111111

Перевірити

Пароль: 1111111111 - Ненадійний

Рис.1-6 Результат виконання програми

Завдання 2. Робота з масивами (4 завдання)

- 1) Написати функцію, яка приймає масив. Вивести елементи масиву, що повторюються.
 - 2) Написати функцію-генератор імен. Функція приймає масив з складів або символів. Функція має повертати згенероване ім'я кішки, собаки, хом'ячка або іншої тваринки
 - 3) Створіть 2 функції:
 - функцію **createArray()**, яка створює масив з випадковою довжиною (від 3 до 7) і випадковими значеннями (від 10 до 20)
 - функцію, яка приймає 2 масиви і виконує наступні дії:
 - З'єднує обидва масиви в один
 - Вилучає повторювані елементи з отриманого масиву
 - Відсортовує отриманий масив за зростанням і повертає його
- ❖ Створіть 2 масиви за допомогою функції **createArray()** та продемонструйте виконання функції з 2-го пункту

					ІПТР.420001.123-ЗЛ	Арк. 3
Змн.	Арк.	№ докум.	Підпис	Дата		

4) Створіть асоціативний масив, де ключами будуть імена користувачів, а значеннями - їхній вік. Реалізуйте функцію для сортування цього масиву в залежності від параметру, переданого у функцію: (віком користувачів або їх імен в алфавітному порядку).

У цьому завданні я створила веб-сторінку, яка демонструє різні способи обробки та маніпулювання масивами в PHP через набір інтерактивних прикладів.

Основні виконані частини:

- Пошук повторюваних елементів у масиві:** Я створила функцію `findDuplicates()`, яка приймає масив і повертає новий масив, що містить лише ті елементи, які зустрічаються вхідному масиві більше одного разу. Для цього я використала функцію `array_count_values()` для підрахунку кількості кожного елемента, а потім `array_filter()` для відбору елементів з кількістю більшою за 1. Насамкінець, `array_keys()` використовується для отримання самих значень елементів (ключів з масиву `counts`):

```
<?php
function findDuplicates($arr) {
    $counts = array_count_values($arr);
    $duplicates = array_filter($counts, function($count) {
        return $count > 1;
    });
    return array_keys($duplicates);
}

$testArray = [1, 2, 3, 2, 4, 3, 5, 6, 5];
$duplicates = findDuplicates($testArray);

echo "<div class='result'>";
echo "Тестовий масив: " . implode(" ", $testArray) . "<br>";
echo "Повторювані елементи: " . implode(" ", $duplicates);
echo "</div>";
?>
```

Головний момент: Ефективне використання комбінації функцій

`array_count_values()`, `array_filter()` та `array_keys()` для знаходження дублікатів у масиві.

- Генератор випадкових імен тварин:** Я розробила функцію `generatePetName()`, яка приймає масив складів і випадковим чином генерує ім'я тварини довжиною 2 або 3 склади, обираючи склади з наданого масиву. Функції `rand()`, `array_rand()`, `ucfirst()` та `strtolower()` використовуються для створення випадкового, але при цьому форматowanego імені:

```
<?php
function generatePetName($syllables) {
    $nameLength = rand(2, 3);
    $name = '';

    for ($i = 0; $i < $nameLength; $i++) {
        $name .= $syllables[array_rand($syllables)];
    }

    return ucfirst(strtolower($name));
}
```

					ІРТР.420001.123-ЗЛ	Арк.
						2
Змн.	Арк.	№ докум.	Підпис	Дата		

```

$syllables = ['pa', 'ma', 'li', 'ka', 'o', 'ta', 'mi', 'pa', 'vi',
'sя'];

echo "<div class='result'>";
echo "Згенеровані імена:<br>";
for ($i = 0; $i < 5; $i++) {
    echo generatePetName($syllables) . "<br>";
}
echo "</div>";
?>

```

Головний момент: Демонстрація створення випадкових комбінацій елементів масиву та форматування рядкових результатів.

3. **Операції з випадково згенерованими масивами:** Я створила дві функції: `createArray()` для генерації масиву випадкових чисел заданої довжини та `processArrays()` для об'єднання двох масивів (`array_merge()`), видалення дублікатів (`array_unique()`) та сортування результату (`sort()`):

```

<?php
function createArray() {
    // ... генерація масиву випадкових чисел ...
    return $array;
}

function processArrays($arr1, $arr2) {
    $result = array_merge($arr1, $arr2);
    $result = array_unique($result);
    sort($result);
    return $result;
}

$array1 = createArray();
$array2 = createArray();
$processed = processArrays($array1, $array2);

echo "<div class='result'>";
echo "Перший масив: " . implode(" ", $array1) . "<br>";
echo "Другий масив: " . implode(" ", $array2) . "<br>";
echo "Результат обробки: " . implode(" ", $processed);
echo "</div>";
?>

```

Головний момент: Ілюстрація послідовного застосування кількох важливих функцій для маніпулювання масивами.

4. **Сортування асоціативного масиву користувачів:** Я розробила функцію `sortUsers()`, яка приймає асоціативний масив користувачів (ім'я => вік) та параметр `$sortBy` для визначення критерію сортування ('name' або 'age'). Залежно від параметра використовуються функції `ksort()` (сортування за ключами) або `asort()` (сортування за значеннями):

```

<?php
function sortUsers($users, $sortBy = 'name') {
    if ($sortBy === 'name') {
        ksort($users);
    } else if ($sortBy === 'age') {
        asort($users);
    }
}

```

					ІРТР.420001.123-ЗЛ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		3

```

        return $users;
    }

    $users = [
        'Анна' => 25,
        'Петро' => 30,
        'Марія' => 22,
        'Іван' => 28,
        'Олена' => 24
    ];

    echo "<div class='result'>";
    // ... виведення оригінального та відсортованих масивів ...
    echo "</div>";
    ?>

```

Головний момент: Демонстрація різних способів сортування асоціативних масивів залежно від потреби (за ключами чи за значеннями).

Робота з масивами

1. Пошук повторюваних елементів

Тестовий масив: 1, 2, 3, 2, 4, 3, 5, 6, 5
Повторювані елементи: 2, 3, 5

2. Генератор імен тварин

Згенеровані імена:
мара
опася
вівімі
мапао
тапара

3. Операції з масивами

Перший масив: 14, 13, 12, 19, 11
Другий масив: 15, 16, 16, 12, 12, 18
Результат обробки: 11, 12, 13, 14, 15, 16, 18, 19

					ІПТР.420001.123-ЗЛ	Арк.
						3
Змн.	Арк.	№ докум.	Підпис	Дата		

4. Сортуння користувачів

Оригінальний список:

Анна: 25 років
Петро: 30 років
Марія: 22 років
Іван: 28 років
Олена: 24 років

Сортуння за іменем:

Іван: 28 років
Анна: 25 років
Марія: 22 років
Олена: 24 років
Петро: 30 років

Сортуння за віком:

Марія: 22 років
Олена: 24 років
Анна: 25 років
Іван: 28 років
Петро: 30 років

[Назад](#)

Рис.7-10 Результат виконання програми

Завдання 3. Робота з формою (1 завдання)

1) Створити файл .htaccess, написати у ньому рядок:

AddDefaultCharset utf-8

2) Створити файл **index.php**

3) На головній сторінці створіть форму з такими елементами:

Логін:	<input type="text"/>
Пароль:	<input type="password"/>
Пароль (ще раз):	<input type="password"/>
Стать:	<input checked="" type="radio"/> чоловік <input type="radio"/> жінка
Місто:	<input type="text" value="Житомир"/>
Улюблені гри:	<input type="checkbox"/> футбол <input type="checkbox"/> баскетбол <input type="checkbox"/> волейбол <input type="checkbox"/> шахи <input type="checkbox"/> World of Tanks
Про себе:	<div></div>
Фотографія:	<div>Выберите файл Файл не выбран</div> <div>Зареєструватися</div>

Не забувайте, щоб форма могла передавати файли, у тезі форм потрібно вказувати атрибуту **enctype="multipart/form-data" method="post"**

4) На іншій сторінці реалізуйте прийом даних цієї форми та відображення переданої інформації, у т.ч. фотографії. Наприклад так:

Логін: morozov.andriy@gmail.com

Пароль: не співпадає (перший - 5 символів, другий - 7 символів)


Стать: чоловік

Місто: Житомир

Улюблені ігри: футбол
шахи

Про себе: Тут можна ввести фрагмент тексту.
Реалізуйте відображення текст з нового рядка
там, де користувач ставить Enter.
Це можна зробити функцією nl2br(\$str).

Фотографія:



[Повернутися на головну сторінку](#)

5) Запишіть усі передані дані у сесію. Реалізуйте заповнення значень форми на головній сторінці значеннями, які зберігаються у формі. Тобто, щоб при переході на головну сторінку елементи форми заповнилися тими значеннями, які збережені у сесії (окрім фотографії):

Логін: morozov.andriy@gmail.com

Пароль:

Пароль (ще раз):

Стать: ☒ чоловік ☐ жінка

Місто: Житомир ▾

Улюблена гра: ☒ футбол
☐ баскетбол
☐ волейбол
☒ шахи
☐ World of Tanks

Про себе: Тут можна ввести фрагмент тексту.
Реалізуйте відображення текст з нового рядка там, де користувач ставить Enter.
Це можна зробити функцією nl2br(\$str).

Фотографія: Файл не выбран

6) На головній сторінці додайте іконки для вибору мови:



					ІПТР.420001.123-ЗЛ	Арк.
						3
Змн.	Арк.	№ докум.	Підпис	Дата		

7) Кожну іконку оберніть у посилання вигляду: **index.php?lang=ukr** і реалізуйте у файлі **index.php** обробник GET-запиту з вибраною мовою. Вибрану мову запишіть у кукі-змінну з часом життя півроку і при відображенні файла **index.php** реалізуйте читання кукі-змінної і виведення фрази вигляду:

Вибрана мова: Українська

index.php (Форма реєстрації):

Цей файл відображає HTML-форму для реєстрації користувача. Форма містить поля для введення логіна, пароля (та його підтвердження), вибору статі, міста, улюблених ігор, інформації "про себе" та завантаження фото. Також реалізовано вибір мови інтерфейсу через cookies. При натисканні кнопки "Зареєструватися" дані форми відправляються на скрипт `process.php`.

Головні моменти:

- **HTML-форма для збору даних користувача.**
- **Реалізація вибору мови інтерфейсу** через cookies та GET-параметр `lang`.
- **Використання масиву `$translations`** для зберігання текстових значень різними мовами.
- **Функція `getTranslation()`** для отримання перекладеного тексту на основі вибраної мови.

process.php (Обробка форми):

Цей файл обробляє дані, надіслані з форми реєстрації (`index.php`). Він отримує значення полів, проводить базову валідацію (перевірка на порожні поля, довжина логіна та пароля, співпадіння паролів, вибір статі та міста). Також обробляється завантаження фото: перевіряється наявність файлу, створюється папка `uploads`, якщо її немає, визначається ім'я файлу та його розширення, перевіряється тип файлу (дозволені `jpg`, `jpeg`, `png`, `gif`) та відбувається спроба переміщення завантаженого файлу до папки `uploads`. У разі успішного завантаження ім'я файлу зберігається в сесії. Якщо під час валідації виникають помилки, вони зберігаються в сесії, і користувач перенаправляється назад на форму. У разі успішної валідації всі введені дані (включаючи шлях до завантаженого фото) зберігаються в сесії, і користувач перенаправляється на сторінку `result.php`.

Головні моменти:

- **Отримання та очищення даних форми** (`trim()`, `htmlspecialchars()`).
- **Базова валідація введених даних** (логін, пароль, статі, місто).
- **Обробка завантаження фото:**
 - Створення папки `uploads`.
 - Перевірка типу файлу (зображення).
 - Переміщення завантаженого файлу до папки `uploads`.
 - Збереження шляху та імені файлу в сесії.
- **Збереження даних користувача в сесії** у разі успішної валідації.
- **Перенаправлення користувача** на сторінку `result.php` у разі успіху або назад на форму з повідомленнями про помилки.

result.php (Відображення введених даних):

					ІПТР.420001.123-ЗЛ	Арк.
						4
Змн.	Арк.	№ докум.	Підпис	Дата		

Цей файл відображає дані, які користувач ввів у форму реєстрації та які були збережені в сесії на попередньому кроці (process.php). Він отримує значення змінних сесії (логін, стать, місто, ігри, інформація "про себе", шлях до фото) та виводить їх на сторінку. Також відображається фото користувача, якщо воно було завантажено та його шлях є в сесії. Якщо користувач не авторизований (відсутня змінна \$_SESSION['login']), його перенаправляють назад на форму реєстрації (index.php).

Головні моменти:

- **Перевірка авторизації** (наявності \$_SESSION['login']).
- **Отримання та відображення даних** з сесійних змінних.
- **Відображення завантаженого фото**, якщо воно є.
- **Використання функції getTranslation()** для відображення текстових значень різними мовами.
- **Посилання для повернення на форму реєстрації**.

Вибрана мова: Українська

Логін:

Yulia188

Пароль:

.....

Повторіть пароль:

.....

Стать:

☐

Чоловік

☒

Жінка

Місто:

Житомир

Місто:

Житомир

Улюблені ігри:

- ☐ Футбол
☒ Баскетбол
☒ Волейбол
☐ Шахи

Про себе:

шугіліоаи цпищкпр дітшгшар укмзгпзйртіои мз9цкер парзпрп тто ткпкешот оиті кдшн
ойктмаііке

Фото:

Вибір файлу 1-4.png

Зареєструватися



Wybrany język: Polski

Login:

Hasło:

Powtórz hasło:

Płeć:

☐

Mężczyzna

☐

Kobieta

Miasto:

Zytomierz

					IPTP.420001.123-3Л	Арк.
						4
Змн.	Арк.	№ докум.	Підпис	Дата		

Введені дані:

Логін: Yulia188

Стать: Жіноча

Місто: Zhytomyr

Ігри: basketball, volleyball

Про себе: шугиліоаи цпищцкпр дітщгшар укмзгпзйртіои мз9цкер парзпрп тто тпкешот оиті кдшн ойктмаіке

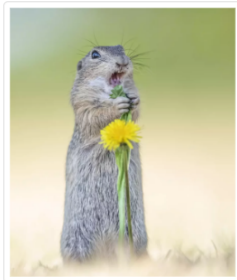


Фото:

[Повернутись до форми](#)

Завдання 4. Робота з функціями (1 завдання)

- 1) Створіть папку з назвою "Function" і в ній створіть файл **func.php** з власними функціями (\sin , \cos , \tan , $\text{my_tg}(x)$, x^y , $x!$). Потім цей файл підключити за допомогою конструкцій `include`, `require`
- 2) Створіть файл **index.php**, який буде головною сторінкою і міститиме форму для вводу даних та виклик функцій:

x	y	
<input type="text" value="4"/>	<input type="text" value="3"/>	<input type="button" value="="/>

- 3) Створіть файл **calculate.php**, який буде обробляти введені дані, викликати функції та виводити результати:

x^y	$x!$	$\text{my_tg}(x)$	$\sin(x)$	$\cos(x)$	$\tan(x)$
64	24	1.15782128235	-0.756802495308	-0.653643620864	1.15782128235

У цьому завданні я створила веб-сторінку, яка дозволяє користувачеві вводити два числа (x та y) та обчислює для x кілька математичних функцій: степінь (x^y), факторіал ($x!$), тангенс ($\tan(x)$), синус ($\sin(x)$), косинус ($\cos(x)$), а також мою власну реалізацію тангенса ($\text{my_tg}(x)$). Результати обчислень відображаються у вигляді таблиці.

Основні виконані частини:

1. **Визначення математичних функцій у файлі `func.php`:** Я створила окремий файл `Function/func.php`, який містить реалізації необхідних математичних функцій: `factorial()`, `power()`, `tangent()`, `my_sin()`, `my_cos()`, та `my_tg()`.

```
<?php
// Function/func.php
function factorial($n) {
    if ($n == 0) return 1;
    return $n * factorial($n - 1);
}
```

```

}

function power($x, $y) {
    return pow($x, $y);
}

function tangent($x) {
    return sin($x) / cos($x);
}

function my_sin($x) {
    return sin($x);
}

function my_cos($x) {
    return cos($x);
}

function my_tg($x) {
    return tan($x);
}
?>

```

Головний момент: Організація коду за функціональністю шляхом винесення математичних функцій в окремий файл для кращої структури проекту.

2. **Форма введення даних (index.php):** Я створила головну сторінку index.php з HTML-формою, яка запитує у користувача значення для x та y . Форма використовує метод POST для передачі даних на сторінку calculate.php:

Головний момент: Створення простого інтерфейсу для користувача для введення необхідних вхідних даних.

3. **Обробка даних та обчислення функцій (calculate.php):** На сторінці calculate.php я спочатку підключаю файл з визначеннями функцій (func.php). Потім я отримую значення x та y з POST-запиту, перетворюю x на число з плаваючою комою, а y на ціле число. Далі я викликаю відповідні функції для обчислення результатів:

```

<?php
require_once "Function/func.php";

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $x = floatval($_POST["x"]);
    $y = intval($_POST["y"]);

    $xy = power($x, $y);
    $x_factorial = factorial(round($x));
    $tg_x = tangent($x);
    $sin_x = my_sin($x);
    $cos_x = my_cos($x);
    $tg_x_php = my_tg($x);
}
?>

```

Головний момент: Отримання даних від користувача та застосування раніше визначених функцій для виконання необхідних обчислень. Важливо відзначити перетворення типів вхідних даних для коректної роботи функцій.

					ІРТР.420001.123-ЗЛ	Арк.
						4
Змн.	Арк.	№ докум.	Підпис	Дата		

4. **Відображення результатів у таблиці (calculate.php):** Після обчислення результатів я створюю HTML-таблицю для їх відображення. У кожному рядку таблиці виводяться назва функції та її обчислене значення. Для умовного виведення значень я використовую тернарний оператор `?:` для перевірки, чи була вже обчислена відповідна змінна:

Головний момент: Зручне та структуроване відображення обчислених результатів за допомогою HTML-таблиці.

x: y:

[Назад](#)

x^y	$x!$	my_tangent(x)	sin(x)	cos(x)	tangent(x)
64	24	1.1578212823496	-0.75680249530793	-0.65364362086361	1.1578212823496

[Назад](#)

Висновок: На сьогоднішньому занятті я навчилась використовувати функції, рядки, масиви та форми в PHP.

					ІРТР.420001.123-ЗЛ	Арк.
						4
Змн.	Арк.	№ докум.	Підпис	Дата		