# Лабораторна робота №2. Основи мови програмування PHP.

**Мета**: отримати практичні навички використання типів даних та основних керуючих конструкцій мови програмування PHP.

**Теоретичні відомості**

PHP – це широко використовувана мова сценаріїв загального призначення з відкритим кодом. Абревіатура PHP означає Hypertext Preprocessor (Препроцесор Гіпертексту). Важливою перевагою мови PHP перед такими мовами, як мов Perl і C, є можливість створення HTML документів з впровадженими командами PHP.

РНР щось середнє між компілятором та інтерпретатором і обробляє сценарії за таким принципом:

 на вхід РНР подається сценарій;

 він перекладається (транслюється у спеціальний байт-код (внутрішнє уявлення);

 виконується байт-код (а не код самої програми!), не створюючи при цьому файл, що виконується.

Байт-код значно компактніший за звичайний код програми, тому він інтерпретується (виконується) швидше.

Змінні PHP визначаються за такими правилами:

 ім'я змінної має починатися зі знака долара $;

 ім'я змінної не повинно містити жодних інших символів, крім символів латинського алфавіту, цифр та знака підкреслення;

 імена змінних у РНР чутливі до рецистру;

 оголошувати змінну можна у будь-якому місці програми, але до місця першого використання.

Змінні РНР можуть містити дані будь-якого типу. Виняток становлять константи, які можуть містити лише число чи рядок. Вибір типу здійснюється самим інтерпретатором. Однак іноді РНР може помилитися та зіставити змінною неправильний тип. У таких випадках потрібно задавати тип явно.

**Таблица 2.1 – Типи змінних у РНР**

|  |  |
| --- | --- |
| **Тип переменной** | **Описание** |
| **Integer** | Целое число. Диапазон значений: от -2 147 483 648 до  -2 147 483 647 (32 бита). |

|  |  |
| --- | --- |
| **double (или float)** | Вещественное число очень большой |
| **String** | Строка. |
| **Array** | Массив. |
| **Object** | Объект. |
| **Bool** | Логический тип данных. Переменные этого типа могут принимать только одно из двух значений: true (истина) или false |

Мова РНР надає багато засобів для визначення типу змін-

 is\_int($x) або is\_integer ($x) – повертає true, якщо передана

змінна - ціле число;

 is\_double($x) або is\_float ($x) – повертає true, якщо передана змінна – дійсне число;

 is\_string ($x) – повертає true, якщо переданий змінний рядок;

 is\_array ($x) – повертає true, якщо передана змінна масив;

 is\_object ($x) – повертає true, якщо змінна є об'єктом;

 is\_bool ($x) – повертає true, якщо змінна оголошена як логічна;

gettype($x) – повертає рядок, що описує тип змінної (integer, double, string, object, array, bool або unknown type, якщо неможливо визначити тип).

Якщо РНР неправильно визначив тип змінної, то можна вказати його явно. Для цього використовується функція settype ($x, $type), де $type - це один із рядків, що повертаються функцією gettype().

Основні оператори мови PHP та їх пріоритет представлені у таблиці 2.2.

**Таблица 2.2 – Пріоритет операцій.**

|  |  |
| --- | --- |
| **Приоритет** | **Оператор** |
| 13 | (постфикс)++ (постфикс ) — |
| 12 | ++(префикс) --(префикс) |
| 11 | \*/% |
| 10 | + - |
| 9 | « » |
| 8 | «= »= |
| 7 | == != |
| 6 | & |
| 5 | « |
| 4 | ! |
| 3 | && |
| 2 | II |
| 1 | = += .= \*=/=%= »=«==&="= | = |

Основные управляющие конструкции языка PHP приоритет представлены в таблице 2.3.

**Таблица 2.3 – Оператори.**

|  |  |
| --- | --- |
| **Название** | **Оператор** |
| Условный оператор | *if (условие) { … }*  *if (условие) { … } else { … }* |
| Цикл с предусловием | *while (условие) { … }* |
| Цикл с постусловием | *do {…} while (условие)* |
| Цикл со счетчиком | f*or($i;$i<10;$i++){…} foreach($items as $val){…}*  *foreach($items as $key=>$val){…}* |
| Оператор switch | *switch ($n) { case “1”: {…} break;*  *…*  *default: {…}*  *}* |

Перевірка існування змінної - дуже зручна можливість мови PHP. Завдяки їй можна перевірити, чи передано сценарію певний параметр. Для перевірки використовують функцію isset ($name).

Щоб не займати пам'ять, можна видалити непотрібні нам змінні.

Це робиться за допомогою функції unset($name).

Виведення даних та значень змінних здійснюється з використанням оператора echo та функції print.

**приклад. Вивести таблицю множення числа 2.**

<?php

$n = 2;

echo “Таблица умножения числа $n <br>”; for($i = 0; $i<=9; $i++){

echo $n.” x “.$i.” = “.$n\*$i.”<br>”;

}

?>

**Завдання**

1. Використовуючи середовище розробки веб-додатків2 створити проект PHP у який додати:
   * папку для зберігання css файлів та файл styles.css;

2 NetBeans IDE, Exclipse IDE або іншу.

* + скрипт index.php з шаблоном документу у форматы HTML5.

1. Використовуючи мову програмування PHP у тілі документа організувати відображення таблиці множення для чисел від 1 до 9.
2. Кожен запис таблиці множення розташувати у окремій комірці таблиці та мати класову помітку про парний та не парний номер комірки; кожен наступний рядок таблиці відокремити пустим рядком (рис. 2.1).
3. Описати у файлі styles.css стилі для придання таблиці необхідного вигляду та під’єднати його до скрипта index.php.
4. Розмістити файли проекту у папці «Lab1» кореневої директорій веб-серверу.
5. Протестувати результати роботи та оформити звіт.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Таблиця множення | | | | | | | | | | |
| 1 | | 1 х 0 = 0 | | | 2 х 0 = 0 | | | 3 х 0 = 0 | | |
| 2 | |  | 1 х 1 = 1 |  |  | … |  |  | … |  |
| 3 | | … | | |  | | |  | | |
| 4 | |  | | |  | | |  | | |
| 5 | |  | | |  | | |  | | |
| 6 | |  | | |  | | |  | | |
| 7 | |  | | |  | | |  | | |
| 8 | |  | | |  | | |  | | |
| 9 | |  | | |  | | |  | | |
| 10 |  |  | | |  | | |  | | |
|  | | | | | | | | | | |
| 11 | | … | | | … | | | … | | |

**Контрольні питання**

1. Які правила визначення змінних у мові програмування PHP?
2. Яким чином можливо примусово встановити тип даних змінної прогрими?
3. Які особливості оголошення та використання масивів?
4. Який масив називають асоціативним?
5. Чи можливо знищувати змінні та перевіряти їх існування?