

១.តើអថេរជាអ្វីក្នុង PHP ?

- Variable(អថេរ) ត្រូវបានប្រើដើម្បីផ្ទុកទិន្នន័យ ដូចជា:
 - តួអក្សរ(String)
 - លេខ(Numbers)
- តម្លៃរបស់ អថេរ អាចធ្វើការផ្លាស់ប្តូរបានតាម ស្រ្តីប(Script)
- រឿងសំខាន់ៗដែលយើងត្រូវដឹង អំពីអថេរគឺ:
 - អថេរមិនចាំបាច់ប្រកាសមុនពេលប្រើប្រាស់ទេ ព្រោះ php8 អាចធ្វើការបម្លែងអថេរហ្នឹងអោយត្រូវគ្នានឹងទិន្នន័យដែលគេបានផ្តល់អោយវា។
 - បន្ទាប់ពីប្រកាសអថេរ វាអាចប្រើឡើងវិញបាន។
 - assignment operator (=) ប្រើដើម្បីកំណត់តម្លៃទៅអថេរ។
- នៅក្នុងអថេរ PHP អាចត្រូវបានប្រកាសថាជា: \$var_name = តម្លៃ;

ឧទាហរណ៍ ៤.១ : 04_ex01_variable.php

```
<?php
// Declaring variables : ប្រកាសអថេរ
$txt = "Hello World!";
$number = 10;
// Displaying variables value បង្ហាញតម្លៃអថេរ
echo $txt; // Output: Hello World!
echo $number; // Output: 10
?>
```

ក្នុងឧទាហរណ៍ខាងលើ យើងបានបង្កើតអថេរពីរ ដែលទីមួយបានកំណត់ដោយតម្លៃខ្សែអក្សរ (string) ហើយទីពីរបានកំណត់ដោយលេខ(number)។ ក្រោយមក យើងបានបង្ហាញតម្លៃអថេរក្នុងកម្មវិធីរុករក(browser) ដោយប្រើ **echo** statement។ **echo** statement ក្នុង php8 ត្រូវបានគេប្រើជាញឹកញាប់ដើម្បីបញ្ចេញទិន្នន័យទៅកាន់កម្មវិធីរុករក(browsers)។

២. ការដាក់ឈ្មោះអោយអថេរ (Variables)

ការដាក់ឈ្មោះអោយអថេរយើងត្រូវគោរពតាមគោលការណ៍ដូចខាងក្រោម៖

- អថេរទាំងអស់ត្រូវបង្កើតដោយប្រើសញ្ញា \$ នៅពីមុខឈ្មោះអថេរ
- ឈ្មោះអថេរត្រូវជាប់ជាមួយអក្សរពី a-z, A-Z ឬជាមួយសញ្ញា Underscore(_)
- អថេរមិនអាចចាប់ផ្តើមដោយលេខបានទេ(0-9)
- អថេរក្នុង php អាចផ្ទុកបានតែ alpha numeric character ហើយនឹង Underscore (a-z, A-Z, 0-9)
- ឈ្មោះអថេរមិនអាចមានដកហ្នា(No Space)
- អថេរគឺ case-sensitive បានន័យថាអក្សរតូច និងធំនិដូចគ្នាទេ \$x មិនដូចគ្នានឹង \$X នេះទេ វាគឺខុសគ្នា។ ដូច្នេះយើងត្រូវប្រុងប្រយ័ត្នពេលប្រកាសអថេរ។

ការបង្កើតអថេរជា String

ឧទាហរណ៍ ៤.២ : 04_ex02_variable.php

```
<?php
$txt = "www.rpitsr.edu.kh";
echo "I love $txt!";
?>

<?php
$txt = "www.rpitsr.edu.kh";
echo "I love " . $txt . "!";
?>
```

ឧទាហរណ៍ខាងក្រោមនឹងបង្ហាញផលបូកនៃអថេរពីរ៖

```
<?php
$x = 5;
$y = 4;
echo $x + $y;
?>
```

៣. Constant ក្នុង PHP8

Constant គឺជាឈ្មោះអថេរមួយប្រភេទដែលគេបង្កើតមកដើម្បីកំណត់តម្លៃថេរ។ បានន័យថា តម្លៃនោះមិនអាចធ្វើការប្តូរតម្លៃជាលើកទីពីរបានទេ។

Constant គឺមានប្រយោជន៍ខ្លាំងណាស់សម្រាប់ការរក្សាទុកទិន្នន័យដែលមិនផ្លាស់ប្តូរខណៈពេលដែលស្ត្រីបកប្រែកំពុងដំណើរការ។

ឧទាហរណ៍ទូទៅនៃទិន្នន័យបែបនេះរួមមានការកំណត់រចនាសម្ព័ន្ធដូចជាមូលដ្ឋានទិន្នន័យ (Database) ឈ្មោះអ្នកប្រើ (Username) និងពាក្យសម្ងាត់ (Password) URL មូលដ្ឋានរបស់គេហទំព័រ (url base) ឈ្មោះក្រុមហ៊ុន (Company Name) ។ល។

Constants ត្រូវបានកំណត់ដោយប្រើ function **define()** របស់ PHP ដែលទទួលយកអាគុយម៉ង់ពីរ (arguments) ៖ ឈ្មោះនៃថេរ (variable name) និងតម្លៃ (value) របស់វា។ នៅពេលកំណត់តម្លៃថេរអាចចូលបានគ្រប់ពេលដោយគ្រាន់តែ ហៅឈ្មោះរបស់វា។ ខាងក្រោមនេះគឺជាឧទាហរណ៍ដ៏សាមញ្ញមួយ៖

ឧទាហរណ៍ ៤.៣ : 04_ex03_constants.php

```
<?php
// Defining constant
define("SITE_URL", "https://www.rpitsr.edu.kh/");
// Using constant
echo '<br>Thank you for visiting - ' . SITE_URL;

define("databse_name", "sourng_db");
define("user_name", "root");
define("password", "Pwd123");

echo "<br>My Databse name is :". databse_name;
// define("databse_name", "my_db");

echo "<br>My Databse name is :". databse_name;
?>
```

៤. វិសាលភាពអថេរ PHP

- នៅក្នុង PHP អថេរអាចត្រូវបានប្រកាសគ្រប់ទីកន្លែងនៅក្នុងស្រ្តីប។
- វិសាលភាពនៃអថេរគឺជាផ្នែកនៃស្រ្តីប ដែលអថេរអាចត្រូវបានReferenced / Used។
- PHP មានវិសាលភាពអថេរបីផ្សេងគ្នា៖
 - ក្នុងស្រុក (local)
 - សកល (global)
 - បីតិវន្ត (static)

៤. Global and Local Scope

អថេរបានប្រកាស **outside(ខាងក្រៅ)** អនុគមន៍(function) មាន GLOBAL SCOPE ហើយនឹងអាចត្រូវបានចូលប្រើប្រាស់ខាងក្រោមអនុគមន៍បាន :

ឧទាហរណ៍ ៤.៤ : 04_ex04_global.php

ការប្រើប្រាស់អថេរជាមួយវិសាលភាពសកល(global scope)៖

```
<?php
    $x = 5; // global scope

    function myTest() {
        // using x inside this function will generate an error
        echo "<p>Variable x inside function is: $x</p>";
    }
    myTest();

    echo "<p>Variable x outside function is: $x</p>";
?>
```

អថេរដែលបានប្រកាស ខាងក្នុង(within) អនុគមន៍មាន LOCAL SCOPE និងអាចត្រូវបានចូលប្រើប្រាស់ខាងក្នុងអនុគមន៍នោះ៖

ឧទាហរណ៍ ៤.៥ : 04_ex05_local_scope.php

ការប្រើប្រាស់អថេរជាមួយ local scope:

```
<?php
function myTestLocal() {
    $x = 5; // local scope
    echo "<p>Variable x inside function is: $x</p>";
}
myTestLocal();

// using x outside the function will generate an error
echo "<p>Variable x outside function is: $x</p>";
?>
```

៥. ប្រើប្រាស់ global Keyword ក្នុង PHP

ចំពោះ **global** keyword គឺត្រូវបានប្រើដើម្បីចូលដើម្បីចូលប្រើនូវអថេរ global ពីខាងក្នុងអនុគមន៍(function)។

ដើម្បីធ្វើរឿងនេះ, ប្រើ **global** keyword ពីមុខអថេរ (ខាងក្នុងអនុគមន៍):

ឧទាហរណ៍ ៤.៦ : 04_ex06_global_keyword.php

```
<?php
$x = 5;
$y = 10;

function myTestGlobal() {
    global $x, $y; //បង្កើតអថេរ global
    $y = $x + $y;
}

myTestGlobal();
echo $y; // outputs 15
?>
```

PHP ក៏រក្សាទុកអថេរសកលទាំងអស់នៅក្នុងអារេមួយហៅថា `$GLOBALS[index]` ដែល `index` រក្សាទុកឈ្មោះរបស់អថេរ។ អារេ(array)នេះក៏អាចចូលប្រើបានពីក្នុងអនុគមន៍(function) និងអាចត្រូវបានប្រើដើម្បីធ្វើបច្ចុប្បន្នភាពអថេរសកល(global variable)ដោយផ្ទាល់ផងដែរ។

ឧទាហរណ៍ខាងលើអាចសរសេរឡើងវិញដូចខាងក្រោម៖

ឧទាហរណ៍ ៤.៧ : 04_ex07_global_index.php

```
<?php
    $x = 5;
    $y = 10;

    function myTestGobal2() {
        $GLOBALS['y'] = $GLOBALS['x'] + $GLOBALS['y'];
    }

    myTestGobal2();
    echo $y; // outputs 15
?>
```

៦. ប្រើប្រាស់ static Keyword

ជាធម្មតា នៅពេលដែលអនុគមន៍មួយត្រូវបានបញ្ចប់/ប្រតិបត្តិ(completed/executed) អថេរទាំងអស់របស់វាត្រូវបានលុប។ ទោះយ៉ាងណាក៏ដោយ ពេលខ្លះយើងចង់ឱ្យអថេរមូលដ្ឋាន(local variables)មិនត្រូវបានលុប។ យើងត្រូវការវាសម្រាប់ការងារបន្ថែម។

ដើម្បីធ្វើដូចនេះសូមប្រើ `static` keyword នៅពេលអ្នកប្រកាសអថេរដំបូង៖

ឧទាហរណ៍ ៤.៨ : 04_ex08_static.php

```
<?php
    function myTest() {
        static $x = 0;
        echo $x;
        $x++;
    }

    myTest();
    myTest();
    myTest();
?>
```