



Introduction PHP

Lesson I



By : SAY PANHA

Tel: 089214054

គោលបំណងនៃមេរៀន

យល់ដឹងអំពី ប្រវត្តិ និង ការប្រើប្រាស់ PHP ព្រមទាំងមូលហេតុដែលវាជាជម្រើសដ៏ពេញនិយមសម្រាប់កម្មវិធី Server-Side ក្នុងការអភិវឌ្ឍវេបសាយ ។

PHP?

PHP (Hypertext Pre-processor) គឺជាភាសា script ដែលបើកចំហរនិងត្រូវបានគេប្រើយ៉ាងទូលំទូលាយ ជាពិសេសសម្រាប់ការអភិវឌ្ឍវេបសាយ ។ PHP ក្នុងដំណើរការនៅលើ Server ហើយបង្កើត HTML ដែលបញ្ជូនទៅ Client ដែលអាចប្រើសម្រាប់បង្កើតទំព័រវេបសាយ Dynamic ។

លក្ខណៈពិសេសនៃ PHP

- **Server-Side Execution:** ក្នុង PHP ត្រូវបានដំណើរការនៅលើ Server ហើយ HTML ត្រូវបានបញ្ជូនទៅ Browser ។
- **Embedded in HTML:** ក្នុង PHP អាចដាក់ចូលក្នុង HTML បាន ដែលអនុញ្ញាតអោយមានការរួមបញ្ចូលយ៉ាងងាយស្រួលជាមួយបច្ចេកវិទ្យា Front-End ។
- **Open Source and Platform-Independent:** PHP គឺអាចប្រើបាស់បានដោយឥតគិតថ្លៃ ហើយវាដំណើរការនៅលើ Platform ជាច្រើនរួមមាន Windows, Linux, និង macOS ។





១០ ៨១ ភ្លឺ PHP

PHP ត្រូវបានបង្កើតឡើងដំបូងក្នុងឆ្នាំ 1994 ដោយ Rasmus Lerdorf ដើម្បីតាមដានចំនួននៃការចូលមើលប្រវត្តិរូបរបស់គាត់នៅលើអ៊ីនធឺណិត ។ តាំងពីពេលនោះមក PHP បានវិវត្តន៍យ៉ាងខ្លាំង ហើយឥឡូវនេះត្រូវបានគ្រប់គ្រងដោយ PHP Group ។ PHP បានឆ្លងកាត់កំណែជាច្រើន ដែលក្នុងនោះកំណែចុងក្រោយគឺ PHP 8 ដែលមានការកែលម្អទំនើបនិងមុខងារថ្មីៗដូចជា JIT (Just-in-Time) Compilation ។





១. តើ អ្វីជា PHP?

PHP មានប្រជាប្រិយភាពសម្រាប់ការអភិវឌ្ឍវេបសាយដោយសារតែវាមានភាពបត់បែន ងាយស្រួលប្រើ និងទទួលបានការគាំទ្រជាច្រើនជុំវិញពិភពលោក ។ ហេតុផលសំខាន់ៗសម្រាប់ការប្រើ PHP រួមមាន៖

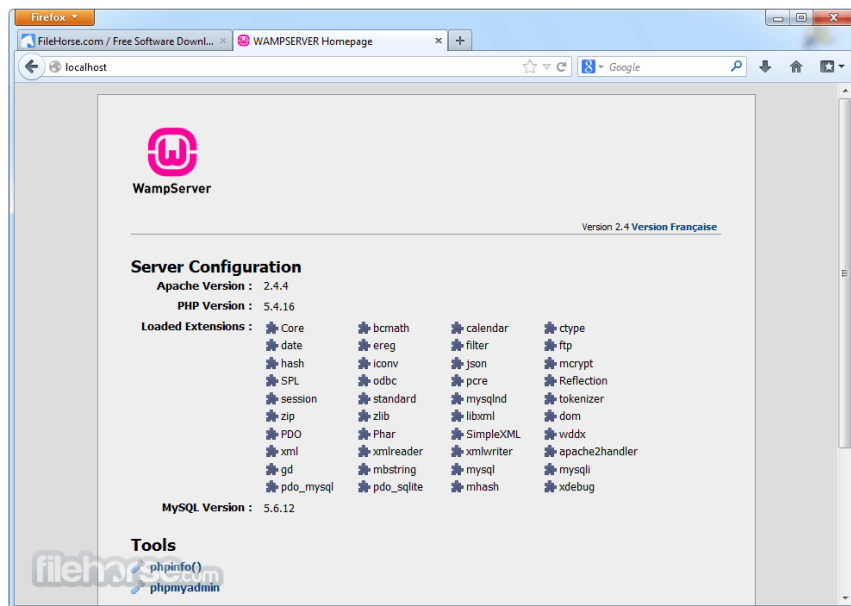
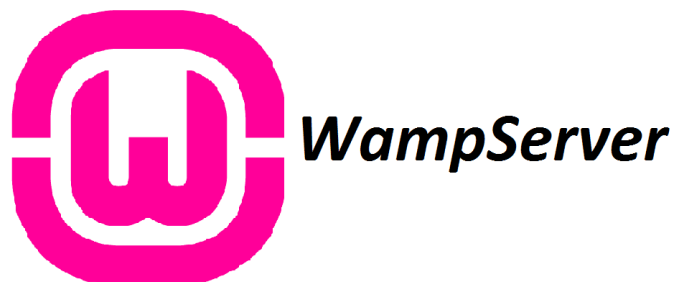
- ងាយស្រួលក្នុងការរៀន និងប្រើប្រាស់: Syntax របស់ PHP សាមញ្ញ ដែលធ្វើអោយវាងាយស្រួលសម្រាប់អ្នកទើបចាប់ផ្តើម ។
- មានការគាំទ្រ និងសហគមន៍ដ៏ទូលំទូលាយ: PHP មានសហគមន៍ធំ ដែលផ្តល់ជូនបណ្ណាល័យ, Frameworks, និងការគាំទ្រ ។
- មានការរួមបញ្ចូលជាមួយ Databases: PHP អាចភ្ជាប់ជាមួយ Database ដូចជា MySQL, PostgreSQL និង SQLite ដែលសំខាន់សម្រាប់ការ ងើតវេបសាយ Dynamic ។





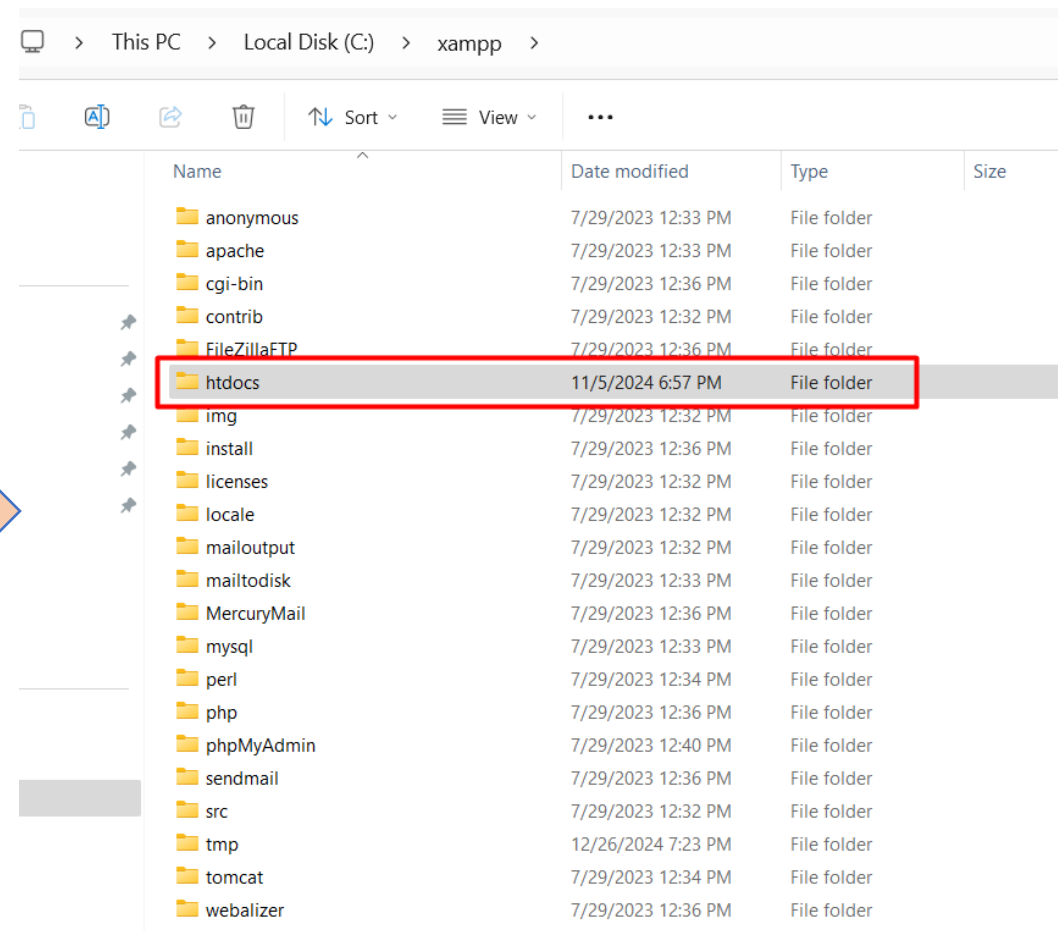
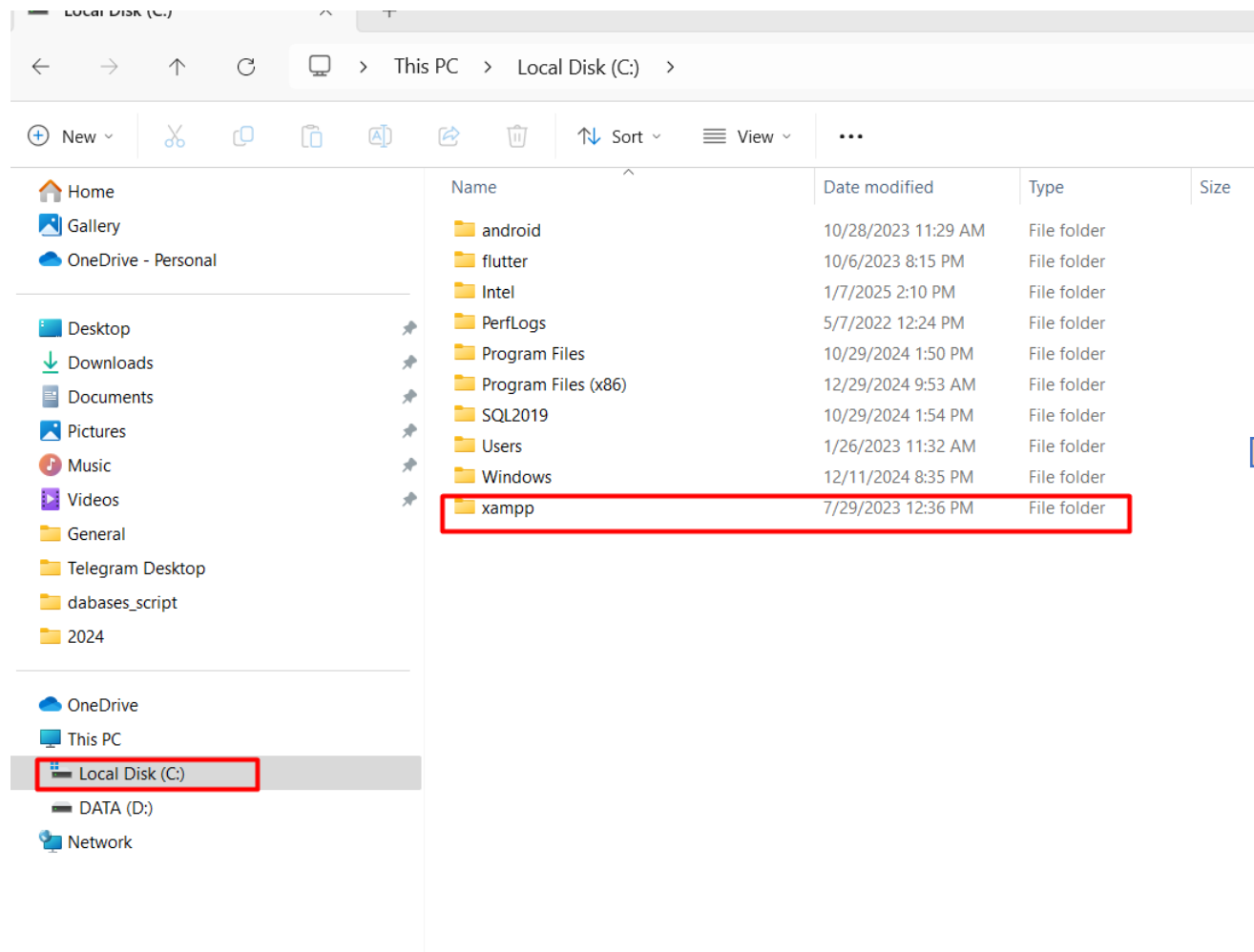
១. តើ អ្វីជា ប្រព័ន្ធ PHP ឬ អ្វីជា ប្រព័ន្ធ ប្រតិបត្តិការ ប្រព័ន្ធ ប្រតិបត្តិការ

ទាញយកឯកសារ Installer សម្រាប់ XAMPP, MAMP ឬ WAMP សម្រាប់ដំឡើងការ Code PHP



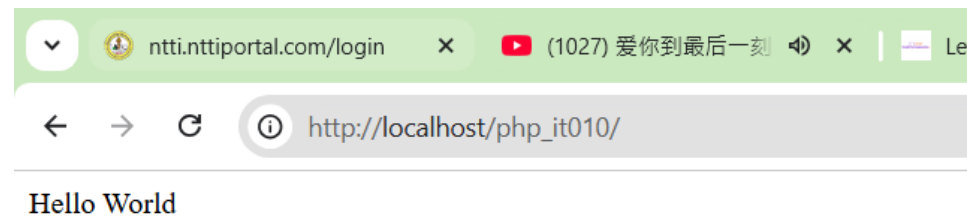
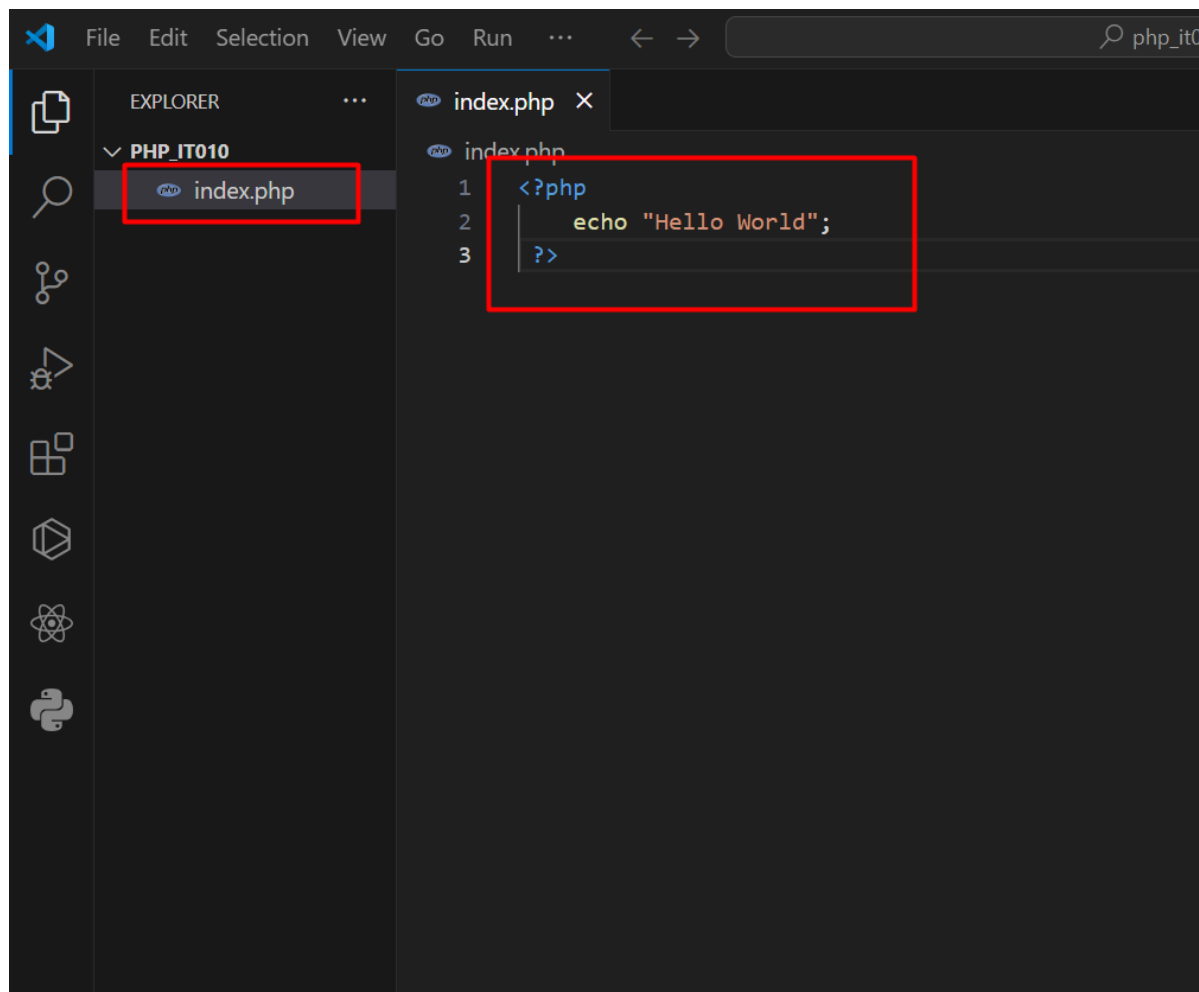


, ។ ៧ ៦៧ ៦៨ ៦៩ ៦ Code





០១ ថ្ងៃ ១២ Folder & File Code PHP





• ជំហានទី១ ៖ ការដាក់ PHP ក្នុង HTML

ដើម្បីយល់កាន់តែច្បាស់អំពី Syntax និងមុខងារមូលដ្ឋាន ដូចជា ការដាក់ PHP ក្នុង HTML ការប្រើ Comments និង Echoing Text ។

• ជំហានទី២ ៖ ការប្រើ echo ដើម្បីបង្ហាញការស្វាគមន៍

បន្ថែម PHP Code Block មួយដែលបង្ហាញ៖

- ការដាក់ PHP ក្នុង HTML ។
- ការប្រើ echo ដើម្បីបង្ហាញការស្វាគមន៍ ។
- ការសរសេរ Single-line និង Multi-line Comments នៅក្នុង PHP ។

• ជំហានទី៣ ៖ ការត្រួតពិនិត្យលទ្ធផល

- Step 1: ចាប់ផ្តើមដោយ tag <!DOCTYPE html> ហើយបង្កើតរចនាសម្ព័ន្ធ HTML មូលដ្ឋាន ។
- Step 2: នៅក្នុង tag <body> បន្ថែម PHP block ដែលមាន <?php ... ?> ។
- Step 3: ប្រើ echo ដើម្បីបង្ហាញការស្វាគមន៍ ដូចជា "Welcome to PHP Basics!" ។
- Step 4: បញ្ចូលទាំង Single-line (//) និង Multi-line (/* ... */) Comments នៅក្នុង PHP Code ។
- Step 5: រក្សាទុកហើយដំឡើងការងារនៅ <http://localhost/index.php> ក្នុង Browser របស់អ្នក ។



PHP Variables ~ ទិន្នន័យ Data Types

Lesson II



By : SAY PANHA

Tel: 089214054



• ៧ ០ ១៥ ១១ ៩៤ ៩៦

យល់អំពីការប្រើប្រាស់ Variables, Data Types, និង Constants ក្នុង PHP ដែលជាគំនិតមូលដ្ឋានសម្រាប់រក្សានិងគ្រប់គ្រងទិន្នន័យក្នុងកម្មវិធីរបស់អ្នក ។ សរសេរកូដ PHP មូលដ្ឋាន ។

I, PHP Variables

Variable អថេរ គឺត្រូវបានប្រើដើម្បីរក្សាទិន្នន័យដែលអាចចូលប្រើ និងដំណើរការបាននៅក្នុងស្រ្តីប ។ អថេរត្រូវបានចាប់ផ្តើមដោយ សញ្ញា \$ តម្លៃជាប្រភេទផ្សេងៗដូចជា អក្សរតួរ, ចំនួនគត់, អាល្លេ, វ៉ត, និងផ្សេងៗទៀត ។
យើងអាចធ្វើការ Define a Variable ជាមួយនឹង ទំរង់ដូចខាងក្រោម:

```
$variable_name = value;
```

- > PHP Variables អាចកំណត់ឈ្មោះដោយមានអក្សរជា a-z, A-Z, 0-9 និង Underscore (_) តែប៉ុណ្ណោះ ។
- > PHP Variables មិនអាចសរសេរ value-name, \$user-data ។
- > Variable មានលក្ខណៈជា PHP (មានន័យថា \$enter ខុសពី \$Enter) ។
- > Variable ត្រូវបាន Assign ជាមួយនឹង សញ្ញា (=) និងបញ្ចប់ Statement ជាមួយនឹង semicolon (;) ។



- symbols or signs in PHP
 - Semicolon(;)
 - Dollar Sign (\$)
 - Period (.)
 - Comma (,)
 - Colon (:)
 - Arrow (->)
 - Equal Sign (=)
 - Single quotes (')
 - Double Quotes(")

.	Period, decimal, point
'	Apostrophe, prime
\$	Dollar sign
,	Comma
/	Slash
%	Percent
{ }	Brace
<	Less than
>	Greater than
"	Quotation mark
()	Parenthesis
?	Question mark
!	Exclamation mark
:	Colon
;	Semicolon
[]	Square bracket
#	Number sign
*	Asteriks
_	Underscore
&	Ampersand



```
Go Run ... < > php_it010

index.php X

index.php > ...
1 <?php
2     $name = "Dara"; // A string variable
3     $age = 20;      // An integer variable
4     ?>
```

```
index.php X

index.php > ...
1 <?php
2     $price = 10;
3     $amount = 20;
4     echo "price $price<br />";
5     echo "amount $amount<br />";
6     ?>
7
```



. PHP Data Types

PHP គាំទ្រ Data Types ច្រើន ដើម្បីរក្សាប្រភេទទិន្នន័យផ្សេងៗ៖

- String: លំដាប់នៃតួអក្សរ ប្រើសម្រាប់រក្សាអត្ថបទ។

```
$greeting = "Hello, World!";
```

- Integer: លេខគត់ដែលមិនមានខ្ទង់ទសភាគ។

```
$year = 2024;
```

- Float (Double): លេខដែលមានខ្ទង់ទសភាគ។

```
$price = 19.99;
```

- Boolean: តំណាងឱ្យតម្លៃពិតដែលអាចមាន គឺ ពិត (true) ឬ មិនពិត (false)។

```
$is_admin = true;
```

- Array: ការរួមបញ្ចូលតម្លៃច្រើនក្នុង Variable មួយ ឬ Indexed ឬ Associative Array។

```
$fruits = array("apple", "banana", "cherry");
```

- NULL: Data Type ពិសេសដែលបង្ហាញថា Variable មិនមានតម្លៃទេ។

```
$empty = NULL;
```

- Object: ជា Instance មួយនៃ Class ដែលអាចមាន Methods និង Properties (ត្រូវបានណែនាំនៅក្នុងផ្នែក Object-Oriented Programming)។



. Operators ក្នុង PHP

សិក្សាអំពីប្រភេទ Operators ក្នុង PHP និងវិធីប្រើវាសម្រាប់អនុវត្តការងារផ្សេងៗ ដូចជាការគណនា ការកំណត់ ការប្រៀបធៀប ការធ្វើប្រតិបត្តិការបែប logical និងការងារផ្សេងៗទៀត ។

I.១០ ១៣៖ Operators ក្នុង PHP

PHP គាំទ្រប្រភេទ Operators ជាច្រើន ដែល Operators នីមួយៗត្រូវបានរចនាឡើងសម្រាប់ដំណើរការជាក់លាក់ ៖

• Arithmetic Operators: ប្រើសម្រាប់ការគណនាបែបមូលដ្ឋាន ។

• + (Addition): $\$a + \b

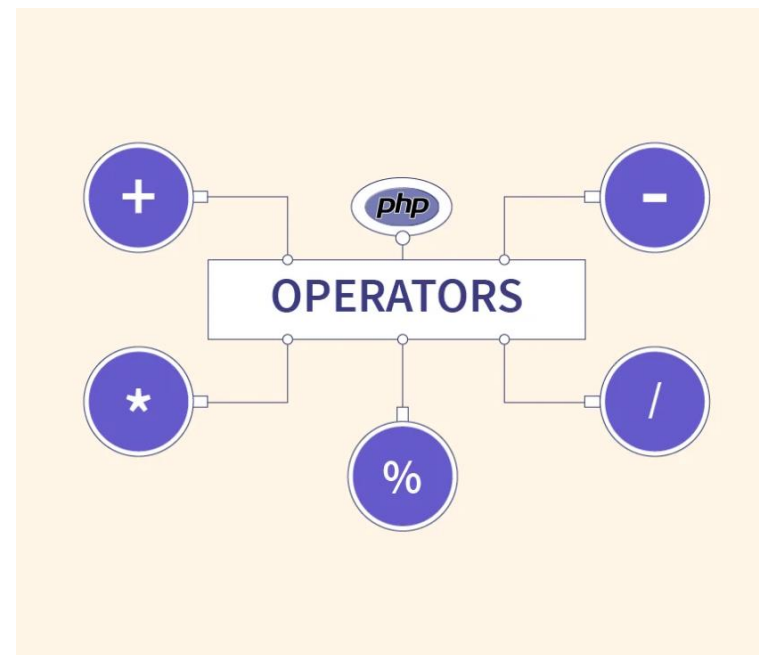
• - (Subtraction): $\$a - \b

• * (Multiplication): $\$a * \b

• / (Division): $\$a / \b

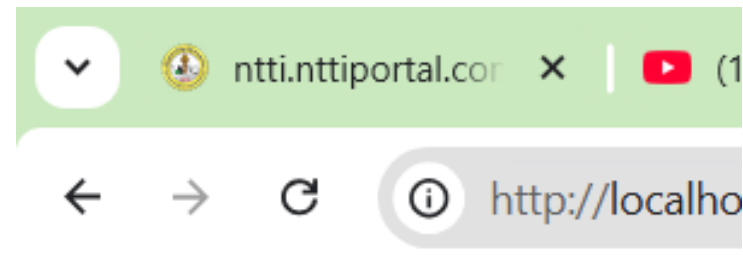
• % (Modulus): $\$a \% \b (សំណល់ពីការចែក)

• ** (Exponentiation): $\$a ** \b (លើកជាស្វ័យគុណ)





```
view  GO  Run  /  php_ko...
...  index.php  X
index.php > ...
1  <?php
2      $result = 5 + 3; // Outputs 8
3      echo $result;
4  ?>
```



8



•Assignment Operators: ប្រើសម្រាប់ផ្តល់តម្លៃទៅ Variables ។

- = (Simple assignment): $\$a = \b
- += (Addition assignment): $\$a += \b (ស្មើនឹង $\$a = \$a + \$b$)
- = (Subtraction assignment): $\$a -= \b
- *= (Multiplication assignment): $\$a *= \b
- /= (Division assignment): $\$a /= \b
- %= (Modulus assignment): $\$a \% = \b

```
... index.php X
index.php > ...
1  <?php
2      $a = 5;
3      $a += 3; // $a is now 8
4      echo $a;
5  ?>
```



•Comparison Operators: ប្រើសម្រាប់ប្រៀបធៀបតម្លៃពីររួមគ្នា ។

•== (Equal): \$a == \$b (true ប្រសិនបើ \$a ស្មើ \$b)

•=== (Identical): \$a === \$b (true ប្រសិនបើ \$a និង \$b ស្មើគ្នា និង មានប្រភេទដូចគ្នា)

•!= ឬ <> (Not equal): \$a != \$b

•!== (Not identical): \$a !== \$b

•< (Less than): \$a < \$b

•> (Greater than): \$a > \$b

•<= (Less than or equal to): \$a <= \$b

•>= (Greater than or equal to): \$a >= \$b

```
index.php
index.php > ...
1  <?php
2      $isEqual = (5 == "5"); // true
3      $isIdentical = (5 === "5"); // false
4
5  ?>
```



- Logical Operators: ប្រើសម្រាប់បញ្ចូលលក្ខខណ្ឌច្រើនរួមគ្នា ។
- && (AND): True ប្រសិនបើលក្ខខណ្ឌទាំងពីរត្រឹមត្រូវ ។
- || (OR): True ប្រសិនបើមានយ៉ាងហោចណាស់លក្ខខណ្ឌមួយត្រឹមត្រូវ ។
- ! (NOT): បញ្ចូលផ្ទុយនៃលក្ខខណ្ឌមួយ

```
... index.php ●  
index.php > ...  
1  <?php  
2      $a = true;  
3      $b = false;  
4      $result = $a && $b; // false  
5  ?>
```



- **Increment and Decrement Operators:** បន្ថែមឬដកតម្លៃនៃ Variable ដោយ 1 ។
- **++\$a (Pre-increment):** បន្ថែម \$a មួយ ហើយបន្ទាប់មកស្តារ \$a ។
- **\$a++ (Post-increment):** ស្តារ \$a ហើយបន្ថែមតម្លៃដោយមួយ ។
- **--\$a (Pre-decrement):** ដក \$a មួយ ហើយបន្ទាប់មកស្តារ \$a ។
- **\$a-- (Post-decrement):** ស្តារ \$a ហើយដកតម្លៃដោយមួយ ។

```
... index.php X
index.php > ...
1  <?php
2      $x = 10;
3      echo ++$x; // 11
4      echo $x--; // 11, then $x becomes 10
5  ?>
```



- String Operators: ប្រើសម្រាប់ភ្ជាប់ String ។
- . (Concatenation): ភ្ជាប់ String ពីរ ។
- .= (Concatenation assignment): បន្ថែម String ទៅ Variable ដែល

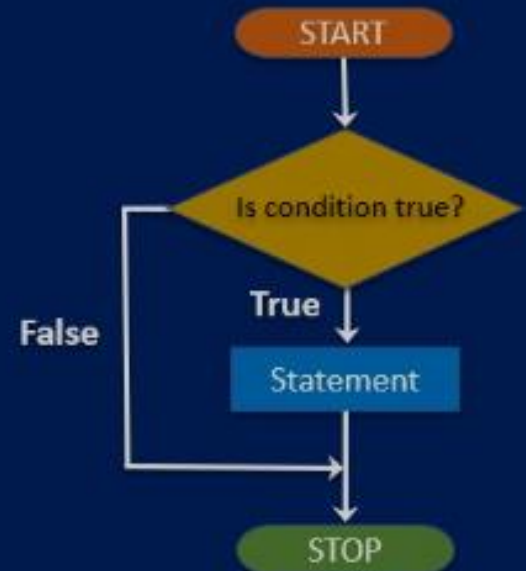
```
.. index.php X
index.php > ...
1  <?php
2      $str = "Hello !";
3      $str .= " World"; // "Hello World"
4      echo $str. "Hello World";
5  ?>
```



Conditionals and loops

Lesson III

if else Statement in PHP



By : SAY PANHA

Tel: 089214054



យល់អំពីរចនាសម្ព័ន្ធនៃការគ្រប់គ្រង(Control Structures) ក្នុង PHP រួមមាន Conditionals និង Loops ដើម្បីគ្រប់គ្រងលំហូរនៃការអនុវត្តកម្មវិធី ដោយផ្អែកលើលក្ខខណ្ឌ និងកិច្ចការដដែលៗ ។

1. Conditional Statements

Conditionals អនុញ្ញាតឱ្យអ្នកដំណើរការកូដផ្សេងៗដោយផ្អែកលើលក្ខខណ្ឌ ជាមួយ Statements ដូចជា if, else, elseif, និង Ternary Operator ។

- if Statement: ដំណើរការកូដប្រសិនបើលក្ខខណ្ឌត្រឹមត្រូវ ។
- else Statement: ផ្តល់ជម្រើសផ្សេងទៀតប្រសិនបើលក្ខខណ្ឌមិនត្រឹមត្រូវ ។

```
View Go Run ... < >
...
index.php X
index.php > ...
1 <?php
2     $age = 18;
3     if ($age >= 18) {
4         echo "Yes";
5     }
6 ?>
```

```
index.php X
index.php > ...
1 <?php
2     $age = 17;
3     if ($age >= 18) {
4         echo "Yss";
5     } else {
6         echo "No";
7     }
8 ?>
```




- elseif Statement: បន្ថែមលក្ខខណ្ឌច្រើន ។

```
index.php X
index.php > ...
1  <?php
2      $score = 85;
3      if ($score >= 90) {
4          echo "Grade: A";
5      } elseif ($score >= 80) {
6          echo "Grade: B";
7      } else {
8          echo "Grade: C";
9      }
10  ?>
```



- Ternary Operator: តំណាងដោយ if-else ។

```
.. index.php X
index.php > ...
1  <?php
2  $status = ($age >= 18) ? "Yes" : "No";
3  ?>
```