

មេរៀនទី ១: Introduction to Web Development & PHP Basics (3 hours)

គោលបំណងមេរៀន

- យល់ដឹងពីមូលដ្ឋានគ្រឹះនៃការអភិវឌ្ឍគេហទំព័រ និងតួនាទីរបស់ PHP
- អាចរៀបចំបរិស្ថានអភិវឌ្ឍន៍ PHP ដោយជោគជ័យ (ដំឡើង XAMPP/WAMP)
- យល់ដឹងពីវាក្យសម្ព័ន្ធ (syntax) មូលដ្ឋាននៃ PHP
- អាចបង្កើត និងប្រើប្រាស់អថេរ និងប្រភេទទិន្នន័យក្នុង PHP
- អាចបង្កើតទំព័រ PHP ដំបូងដោយជោគជ័យ

១. ទិដ្ឋភាពទូទៅនៃ Web Development

តើអ្វីជា Web Development?

Web Development គឺជាដំណើរការនៃការបង្កើត និងថែរក្សាគេហទំព័រ ឬកម្មវិធីគេហទំព័រ។ វាគ្របដណ្តប់លើទិដ្ឋភាពផ្សេងៗជាច្រើន រួមមាន៖

- Front-end Development** - ផ្នែកដែលអ្នកប្រើប្រាស់អាចមើលឃើញ និងអន្តរកម្មបាន ផ្តោតលើ HTML, CSS និង JavaScript
- Back-end Development** - ផ្នែកដែលដំណើរការនៅលើម៉ាស៊ីនមេ រួមមានភាសាប្រកបដោយ PHP, Python, Java, Node.js ជាដើម
- Database Management** - ការរក្សាទុក និងគ្រប់គ្រងទិន្នន័យនៅពីក្រោយគេហទំព័រ

ការប្រើប្រាស់ធនធានអ៊ីនធឺណិត

គេហទំព័រ និងកម្មវិធីគេហទំព័រត្រូវបានដំណើរការតាមរយៈផ្នែកសំខាន់ៗពីរ៖

- Client-side** - សំដៅទៅលើកុំព្យូទ័រ ឬឧបករណ៍របស់អ្នកប្រើប្រាស់ ដែលប្រើប្រាស់ Browser ដើម្បីមើល និងអន្តរកម្មជាមួយគេហទំព័រ
- Server-side** - សំដៅទៅលើម៉ាស៊ីនមេដែលដំណើរការកូដ និងផ្តល់គេហទំព័រទៅកាន់ Browser

លំហូរនៃការស្នើសុំគេហទំព័រ

- អ្នកប្រើប្រាស់វាយអាសយដ្ឋានគេហទំព័រក្នុង Browser
- Browser ផ្ញើសំណើទៅម៉ាស៊ីនមេ
- ម៉ាស៊ីនមេដំណើរការសំណើ (ដូចជាការដំណើរការ PHP)
- ម៉ាស៊ីនមេប្រទានជូននូវ HTML ទៅកាន់ Browser
- Browser បង្ហាញគេហទំព័រជូនអ្នកប្រើប្រាស់

បច្ចេកវិទ្យា Web Development ដែលពេញនិយម

- **Front-end:** HTML, CSS, JavaScript, React, Angular, Vue.js
- **Back-end:** PHP, Python, Ruby, Java, Node.js, .NET
- **Databases:** MySQL, PostgreSQL, MongoDB, Oracle, SQL Server

២. តើអ្វីជា PHP? និងហេតុអ្វីត្រូវប្រើប្រាស់វា?

តើអ្វីជា PHP?

PHP (PHP: Hypertext Preprocessor) គឺជាភាសាស្រ្តីបម៉ាស៊ីនមេបើកចំហសម្រាប់ការអភិវឌ្ឍគេហទំព័រ។ វាត្រូវបានបង្កើតឡើងដោយ Rasmus Lerdorf នៅឆ្នាំ ១៩៩៤។

លក្ខណៈពិសេសចម្បងនៃ PHP រួមមាន៖

- វាត្រូវបានដំណើរការនៅលើម៉ាស៊ីនមេ (មិនមែននៅលើកុំព្យូទ័រអ្នកប្រើប្រាស់ទេ)
- វាបង្កើត HTML ដែលត្រូវបានផ្ញើទៅកាន់ Browser
- វាអាចចាក់បញ្ចូលទៅក្នុង HTML ដោយប្រើស្លាក `<?php ?>`
- វាអាចទាញយក និងរក្សាទុកទិន្នន័យក្នុងមូលដ្ឋានទិន្នន័យ
- វាអាចដំណើរការលើប្រព័ន្ធប្រតិបត្តិការផ្សេងៗគ្នា (Windows, Linux, Mac)

ហេតុអ្វីត្រូវប្រើ PHP?

1. **ងាយស្រួលរៀន** - PHP មានវាក្យសម្ព័ន្ធដែលងាយស្រួលយល់សម្រាប់អ្នកចាប់ផ្តើម
2. **ប្រសិទ្ធភាព** - វាមានដំណើរការលឿនជាងភាសាជាច្រើនផ្សេងទៀត
3. **ភាពអាចប្តូរបាន** - វាអាចប្រើជាមួយប្រព័ន្ធប្រតិបត្តិការ និងម៉ាស៊ីនមេវេបភាគច្រើន
4. **Community Support** - PHP មានសហគមន៍ធំ និងមានធនធានជាច្រើន
5. **ការប្រើប្រាស់ទូលំទូលាយ** - គេហទំព័រល្បីៗជាច្រើនប្រើ PHP (Facebook, WordPress, Wikipedia)
6. **ការគាំទ្រ Database** - PHP ធ្វើការបានយ៉ាងល្អជាមួយ MySQL និងមូលដ្ឋានទិន្នន័យផ្សេងទៀត

ឧទាហរណ៍មូលដ្ឋាននៃកូដ PHP

```
php
<?php
    echo "Hello, World!";
?>
```

៣. ការរៀបចំបរិស្ថានអភិវឌ្ឍន៍ (XAMPP/WAMP)

ដើម្បីអភិវឌ្ឍជាមួយ PHP អ្នកត្រូវការកញ្ចប់ម៉ាស៊ីនមេវេបមួយដែលមាន៖

- PHP
- ម៉ាស៊ីនមេវេប (ជាទូទៅ Apache)
- មូលដ្ឋានទិន្នន័យ (ជាទូទៅ MySQL)

កញ្ចប់ដែលពេញនិយមបំផុតគឺ៖

XAMPP

XAMPP គឺជាកញ្ចប់បើកចំហសម្រាប់ប្រព័ន្ធប្រតិបត្តិការផ្សេងៗគ្នា (Windows, Linux, macOS) ដែលរួមមាន៖

- **X:** Cross-platform
- **A:** Apache
- **M:** MariaDB/MySQL
- **P:** PHP
- **P:** Perl

ការដំឡើង XAMPP

1. ទាញយក XAMPP ពីគេហទំព័រ <https://www.apachefriends.org>
2. ដំណើរការឯកសារដំឡើង និងធ្វើតាមការណែនាំ
3. បន្ទាប់ពីដំឡើងរួច ចូលទៅកាន់ XAMPP Control Panel ហើយចាប់ផ្តើមម៉ូឌុល Apache និង MySQL
4. សាកល្បងដោយចូលទៅកាន់ <http://localhost> ក្នុង Browser របស់អ្នក
5. ឯកសារ PHP គួរត្រូវបានរក្សាទុកក្នុងថតឯកសារ `htdocs` ក្នុងថតឯកសារដំឡើង XAMPP

WAMP (សម្រាប់ Windows)

WAMP ត្រូវបានប្រើជាទូទៅលើ Windows ដែលរួមមាន៖

- **W:** Windows
- **A:** Apache
- **M:** MySQL
- **P:** PHP

ការដំឡើង WAMP

1. ទាញយក WAMP ពីគេហទំព័រ <https://www.wampserver.com>
2. ដំណើរការឯកសារដំឡើង និងធ្វើតាមការណែនាំ
3. បន្ទាប់ពីដំឡើងរួច WAMP នឹងចាប់ផ្តើមអនុវត្ត
4. ឯកសារ PHP គួរត្រូវបានរក្សាទុកក្នុងថតឯកសារ `www` ក្នុងថតឯកសារដំឡើង WAMP

៤. PHP syntax មូលដ្ឋាន និងរចនាសម្ព័ន្ធ

ការសរសេរកូដ PHP

កូដ PHP ត្រូវសរសេរនៅក្នុងស្លាកចាប់ផ្តើម `<?php` និងស្លាកបញ្ចប់ `?>`:

php

```
<?php
    // កូដ PHP នៅទីនេះ
?>
```

អ្នកអាចលាយបញ្ចូល PHP ជាមួយ HTML:

php

```
<!DOCTYPE html>
<html>
<head>
    <title>ទំព័រ PHP របស់ខ្ញុំ</title>
</head>
<body>
    <h1>ស្វាគមន៍មកកាន់គេហទំព័រ PHP របស់ខ្ញុំ!</h1>

    <?php
        echo "<p>នេះជាកូដបង្កើតដោយ PHP</p>";
    ?>

    <p>នេះជា HTML ធម្មតា</p>
</body>
</html>
```

ការសម្គាល់ (Comments)

php

```
<?php
    // ការសម្គាល់មួយបន្ទាត់

    /*
    ការសម្គាល់
    ច្រើនបន្ទាត់
    */
?>
```

ការបង្ហាញលទ្ធផល

PHP មានពាក្យបញ្ជាច្រើនសម្រាប់បង្ហាញលទ្ធផល៖

php

```
<?php
    echo "Hello World!"; // បង្ហាញអក្សរ

    print "ប្រើប្រាស់ print"; // ដូចគ្នានឹង echo

    // បង្ហាញព័ត៌មានលម្អិតអំពីអថេរ
    $number = 123;
    var_dump($number); // បង្ហាញប្រភេទទិន្នន័យ និងតម្លៃ

    // បង្ហាញព័ត៌មានតាមទម្រង់អានបានងាយស្រួល
    print_r($number);
?>
```

ប្រយោគ (Statements)

ប្រយោគនីមួយៗក្នុង PHP បញ្ចប់ដោយសញ្ញាក្បឿងស (:):

php

```
<?php
    $a = 1;
    $b = 2;
    $c = $a + $b;
    echo $c;
?>
```

៥. អថេរ និងប្រភេទទិន្នន័យក្នុង PHP

ការប្រកាសអថេរ

អថេរនៅក្នុង PHP ចាប់ផ្តើមដោយសញ្ញា \$ រួចតាមដោយឈ្មោះអថេរ:

php

```
<?php
    $name = "Dara"; // អថេរ string
    $age = 25;       // អថេរ integer
    $height = 1.75; // អថេរ float
    $isStudent = true; // អថេរ boolean
?>
```

ក្នុងន័យសម្រាប់ការបង្កើតឈ្មោះអថេរ:

- ត្រូវចាប់ផ្តើមដោយអក្សរ ឬសញ្ញា underscore (_)
- អាចមានអក្សរ លេខ និងសញ្ញា underscore

- គិតគូរករណីអក្សរធំតូច (case-sensitive)
- ប្រើឈ្មោះដែលពណ៌នាពីខ្លឹមសារ

ប្រភេទទិន្នន័យមូលដ្ឋាន

PHP គាំទ្រប្រភេទទិន្នន័យដូចខាងក្រោម៖

1. **String** - អត្ថបទ (ដាក់ក្នុងសញ្ញាដកសម្រង់ទោល ' ' ឬដកសម្រង់ទ្វេ " ")

php

```
<?php
    $firstName = 'Sokha';
    $lastName = "Kim";

    // ការភ្ជាប់ strings
    echo $firstName . ' ' . $lastName; // Sokha Kim

    // ការប្រើអថេរក្នុង double quotes
    echo "ឈ្មោះគឺ $firstName $lastName"; // ឈ្មោះគឺ Sokha Kim
?>
```

2. **Integer** - លេខគត់

php

```
<?php
    $age = 25;
    $negativeNumber = -10;

    echo $age + 5; // 30
?>
```

3. **Float** (ឬ Double) - លេខទសភាគ

php

```
<?php
    $price = 19.99;
    $pi = 3.14159;

    echo $price * 2; // 39.98
?>
```

4. **Boolean** - តម្លៃពិត (true) ឬមិនពិត (false)

php

```
<?php
    $isLoggedIn = true;
    $hasPermission = false;

    // Booleans ត្រូវបានបង្ហាញជា "1" សម្រាប់ true និងមិនបង្ហាញអ្វីទេសម្រាប់ false
    echo $isLoggedIn; // បង្ហាញ 1
    echo $hasPermission; // មិនបង្ហាញអ្វីទេ
?>
```

5. Array - ការប្រមូលផ្តុំនៃតម្លៃ

php

```
<?php
    // Indexed array
    $fruits = array("Apple", "Banana", "Orange");
    // ឬ ទម្រង់សរសេរថ្មី៖
    $fruits = ["Apple", "Banana", "Orange"];

    echo $fruits[0]; // Apple

    // Associative array (key-value)
    $person = [
        "name" => "Bopha",
        "age" => 30,
        "city" => "Phnom Penh"
    ];

    echo $person["name"]; // Bopha
?>
```

6. NULL - តម្លៃទំនេរ

php

```
<?php
    $var = NULL;

    // ពិនិត្យថាតើអថេរមានតម្លៃ NULL ឬអត់
    if(is_null($var)) {
        echo "អថេរមានតម្លៃ NULL";
    }
?>
```

7. Object - វត្ថុ

php

```
<?php
class Person {
    public $name;
    public $age;

    function __construct($name, $age) {
        $this->name = $name;
        $this->age = $age;
    }

    function introduce() {
        return "ខ្ញុំឈ្មោះ " . $this->name . " និងមានអាយុ " . $this->age . " ឆ្នាំ";
    }
}

$person = new Person("Thida", 22);
echo $person->introduce(); // ខ្ញុំឈ្មោះ Thida និងមានអាយុ 22 ឆ្នាំ
?>
```

ការប្លែងប្រភេទទិន្នន័យ (Type Casting)

PHP អនុញ្ញាតឱ្យប្លែងប្រភេទទិន្នន័យពីមួយទៅមួយ:

php

```
<?php
$stringNumber = "42";
$number = (int)$stringNumber; // ប្លែងពី string ទៅ integer

$floatNumber = 3.14;
$intNumber = (int)$floatNumber; // ប្លែងពី float ទៅ integer

$numericValue = 123;
$stringValue = (string)$numericValue; // ប្លែងពី integer ទៅ string
?>
```

៦. Lab: ការបង្កើតទំព័រ PHP ដំបូង

សេចក្តីណែនាំ:

ក្នុង Lab នេះ អ្នកនឹងបង្កើតទំព័រ PHP ដំបូងដែលបង្ហាញព័ត៌មានផ្ទាល់ខ្លួនរបស់អ្នក និងកាលបរិច្ឆេទបច្ចុប្បន្ន។

ជំហានទី ១: បង្កើតឯកសារ PHP

1. បើក Text Editor ដែលអ្នកចូលចិត្ត (Notepad++, VS Code, ជាដើម)
2. បង្កើតឯកសារថ្មីហើយរក្សាទុកជា `index.php` ក្នុងថតឯកសារ `htdocs` (XAMPP) ឬ `www` (WAMP)

ជំហានទី ២៖ សរសេរកូដ PHP

php

```
<!DOCTYPE html>
<html>
<head>
  <title>ទំព័រ PHP ដំបូងរបស់ខ្ញុំ</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      max-width: 800px;
      margin: 0 auto;
      padding: 20px;
      background-color: #f5f5f5;
    }
    .container {
      background-color: white;
      padding: 20px;
      border-radius: 5px;
      box-shadow: 0 0 10px rgba(0,0,0,0.1);
    }
    h1 {
      color: #4285f4;
    }
    .info-item {
      margin-bottom: 10px;
    }
    .label {
      font-weight: bold;
      width: 150px;
      display: inline-block;
    }
  </style>
</head>
<body>
  <div class="container">
    <h1>ទំព័រ PHP ដំបូងរបស់ខ្ញុំ</h1>

    <?php
      // កំណត់ព័ត៌មានផ្ទាល់ខ្លួន
      $name = "ឈ្មោះរបស់អ្នក";
      $age = 20;
      $city = "ទីក្រុងរបស់អ្នក";
      $hobbies = ["ចំណង់ចំណូលចិត្ត 1", "ចំណង់ចំណូលចិត្ត 2", "ចំណង់ចំណូលចិត្ត 3"];
      $isStudent = true;

      // បង្ហាញព័ត៌មានផ្ទាល់ខ្លួន
      echo "<div class='info-item'><span class='label'>ឈ្មោះ៖</span> $name</div>";
      echo "<div class='info-item'><span class='label'>អាយុ៖</span> $age ឆ្នាំ</div>";
```

```

echo "<div class='info-item'><span class='label'>ទីក្រុង៖</span> $city</div>";

// បង្ហាញចំណង់ចំណូលចិត្ត
echo "<div class='info-item'><span class='label'>ចំណង់ចំណូលចិត្ត៖</span> ";
echo $hobbies[0] . ", " . $hobbies[1] . ", " . $hobbies[2];
echo "</div>";

// បង្ហាញស្ថានភាពនិស្សិត
echo "<div class='info-item'><span class='label'>ជានិស្សិត៖</span> ";
echo $isStudent ? "បាទ/ចាស" : "ទេ";
echo "</div>";

// បង្ហាញកាលបរិច្ឆេទ និងពេលវេលាបច្ចុប្បន្ន
echo "<div class='info-item'><span class='label'>កាលបរិច្ឆេទ៖</span> " . date("");
echo "<div class='info-item'><span class='label'>ពេលវេលា៖</span> " . date("H:

// គណនាឆ្នាំកំណើត
$birthYear = date("Y") - $age;
echo "<div class='info-item'><span class='label'>ឆ្នាំកំណើតប្រហាក់ប្រហែល៖</span>

?>

<h2>ការគណនាងាយៗ</h2>
<?php
// ធ្វើការគណនាងាយៗ
$num1 = 10;
$num2 = 5;

echo "<div class='info-item'><span class='label'>លេខទី១៖</span> $num1</div>";
echo "<div class='info-item'><span class='label'>លេខទី២៖</span> $num2</div>";
echo "<div class='info-item'><span class='label'>ផលបូក៖</span> " . ($num1 + $
echo "<div class='info-item'><span class='label'>ផលដក៖</span> " . ($num1 - $
echo "<div class='info-item'><span class='label'>ផលគុណ៖</span> " . ($num1 *
echo "<div class='info-item'><span class='label'>ផលចែក៖</span> " . ($num1 /

?>
</div>
</body>
</html>

```

ជំហានទី ៣៖ ដំណើរការទំព័រ PHP

1. ចាប់ផ្តើម XAMPP/WAMP បើសិនជាមិនទាន់ដំណើរការ
2. បើក Browser របស់អ្នក
3. ចូលទៅកាន់ <http://localhost/index.php>
4. ពិនិត្យមើលថាទំព័ររបស់អ្នកដំណើរការដោយជោគជ័យ

ជំហានទី ៤៖ កែសម្រួលឯកសារ

1. ជំនួសតម្លៃរបស់អថេរដោយព័ត៌មានផ្ទាល់ខ្លួនរបស់អ្នក
2. រក្សាទុកឯកសារ និងផ្ទុកឡើងវិញ
3. សង្កេតមើលការផ្លាស់ប្តូរនៅលើគេហទំព័ររបស់អ្នក

គម្រោងសិក្សាបន្ថែម

គម្រោងទី១: បង្កើតទំព័រណែនាំខ្លួនជាមួយ PHP

បង្កើតទំព័រណែនាំខ្លួនពេញលេញដែលបង្ហាញព័ត៌មានដូចខាងក្រោម៖

- ព័ត៌មានផ្ទាល់ខ្លួន (ឈ្មោះ អាយុ ទីក្រុង)
- ការអប់រំ (ឯកទេស សាលា ឆ្នាំបញ្ចប់)
- ជំនាញ (បង្ហាញជា array)
- បទពិសោធន៍ការងារ (បង្ហាញជា associative array)
- ប្រើប្រាស់ CSS ដើម្បីធ្វើឱ្យទំព័ររបស់អ្នកទាក់ទាញ

គម្រោងទី២: បង្កើតកម្មវិធីគណនាមូលដ្ឋាន

បង្កើតកម្មវិធីគណនាមូលដ្ឋានដែលអាច៖

- បូក ដក គុណ និងចែកលេខពីរ
- គណនាផ្ទៃក្រឡា និងបរិមាត្ររូបធរណីមាត្រ (ការ៉េ រង្វង់ ត្រីកោណ)
- បង្ហាញលទ្ធផលជាទម្រង់តារាង
- ប្រើប្រាស់ CSS ដើម្បីធ្វើឱ្យកម្មវិធីគណនាមានរូបរាងទាក់ទាញ

គម្រោងទី៣: បង្កើតទំព័របង្ហាញព័ត៌មានសម្រាប់ប្រទេសក្នុងតំបន់អាស៊ាន

បង្កើតទំព័រ PHP ដែលបង្ហាញព័ត៌មានដូចខាងក្រោម៖

- បង្កើត array ជាមួយព័ត៌មានអំពីប្រទេសក្នុងតំបន់អាស៊ាន (ឈ្មោះ ទីក្រុងរដ្ឋធានី ប្រជាជន ផ្ទៃដី)
- បង្ហាញព័ត៌មានជាទម្រង់តារាង
- គណនាចំនួនប្រជាជនសរុប និងផ្ទៃដីសរុបរបស់តំបន់អាស៊ាន
- ប្រើប្រាស់ condition ដើម្បីបង្ហាញព័ត៌មានផ្សេងៗគ្នាសម្រាប់ប្រទេសដែលមានប្រជាជនច្រើនជាង ៥០ លាននាក់

សេចក្តីសន្និដ្ឋាន

នៅក្នុងមេរៀនដំបូងនេះ យើងបានរៀនអំពីមូលដ្ឋានគ្រឹះនៃការអភិវឌ្ឍគេហទំព័រ និងភាសា PHP។ យើងបានស្វែងយល់ពីអ្វីជា PHP និងហេតុអ្វីបានជាវាមានប្រជាប្រិយភាព។ យើងបានរៀនពីរបៀបដំឡើង និងរៀបចំបរិស្ថានអភិវឌ្ឍន៍ដូចជា XAMPP និង WAMP។ លើសពីនេះ យើងបានមើលរក្សាសម្ព័ន្ធមូលដ្ឋាន និងរចនាសម្ព័ន្ធនៃ PHP រួមជាមួយអថេរ និងប្រភេទទិន្នន័យផ្សេងៗ។ ជាចុងក្រោយ យើងបានបង្កើតទំព័រ PHP ដំបូងរបស់យើង។

នៅមេរៀនបន្ទាប់ យើងនឹងពិនិត្យមើលលម្អិតបន្ថែមលើ PHP Control Structures និង Functions ដែលនឹងអនុញ្ញាតឱ្យយើងសរសេរកូដកាន់តែស្មុគស្មាញ និងមានប្រសិទ្ធភាព។

ឯកសារយោង

1. ឯកសារផ្លូវការរបស់ PHP: <https://www.php.net/docs.php>
2. W3Schools PHP Tutorial: <https://www.w3schools.com/php/>
3. XAMPP: <https://www.apachefriends.org>
4. WAMP: <https://www.wampserver.com>