

1. គេត្រូវការ Web Dynamic Server-side សម្រាប់ Hosting files លើ Web Server និងប្រើប្រាស់ Server-side Technologies ដើម្បីបង្កើត Scripts ដែល Dynamically generate content ។

* ដើម្បី Develop Web Dynamic Server-side គេត្រូវការ

2. Web Server ជា Computer software និង Hardware ពាក់ព័ន្ធ សម្រាប់ ទទួលសំណើរជា HTTP ដើម្បីរក្សាទុក ដំណើរការ និង deliver web pages ទៅកាន់ users

* Database Server ជា Server ដែលប្រើប្រាស់ database application សម្រាប់ផ្តល់ database services ទៅកាន់ Computer programs ឬ Computer ដទៃទៀត

3. PHP ជា Script language run ខាង Server Side សម្រាប់លាយ ជាមួយ HTML ដើម្បីបង្កើត Dynamic and Interactive Web Page (ឆ្លើយតបទៅនឹង User's Input) ផ្នែក Server ។

របៀប Embed PHP in HTML ៖

<pre><!DOCTYPE html> <html> <head> <?php ... ?> </head></pre>	<pre><body> <?php ... ?> </body> </html></pre>
---	---

4. លក្ខណៈពិសេសរបស់ អថេរ (Variable) ក្នុង PHP គឺ

- ប្រកាស ដោយប្រើសញ្ញា \$
- Data Type នៃ អថេរ អាស្រ័យលើ តម្លៃដែលផ្តល់ឱ្យ
- ប្រើជា Container សម្រាប់ផ្ទុកទិន្នន័យ
- មិនមាន Command សម្រាប់ Declare

- កើតមានឡើង នៅពេលយើង Assign តម្លៃដំបូងឱ្យវា

Syntax: \$varName = value

Ex:

```
<php ?  
  
    $name = "Khavy SR";    $age = 21;  
  
    echo $name;  
  
?>
```

5. Array in PHP មាន 3

-Indexed Array: គឺជា Array ដែលមាន index ជាលេខ

```
$x = array ( "A", "B", "C" );  
  
foreach ( $x as $value )  
{ echo $value . "<br/>"; }
```

-Associative Array: គឺជា Array ដែលមាន index ជា អក្សរ

```
$x = array ( "name"=> "sok", "age"=>25 );  
  
foreach ( $x as $key => $value )  
{ echo $key . ":" . $value. "<br/>"; }
```

-Multidimensional Array: គឺជា Array ដែលមានច្រើនវិមាត្រ

```
$x = array (
    array( "name"=> "sok", "age"=>25 ),
    array( "name"=> "chan", "age"=>18 )
);
foreach ( $x as $row ) {
    foreach ( $row as $key => $value )
    { echo $key . ":" . $value. "<br/>"; }
}
```

6. Cookie ជាវិធីសាស្ត្រមួយ ដែលមានប្រយោជន៍សម្រាប់រក្សាទិន្នន័យទុកក្នុងរយៈពេលកំណត់ជាក់លាក់ មួយអាចគិតជា វិនាទី នាទី ម៉ោង ថ្ងៃ ខែ ឆ្នាំ ផ្ទុកខាង client-side ក្នុង web browser ។

Ex:

```
< ?php setcookie( "name", "EX1", time( ) + ( 60 * 60 ) ); ?>

<html><body>

< ?php print_r( $_COOKIE ); ?>

</body></html>
```

7. Session វិធីសាស្ត្រមួយ ដែលមានប្រយោជន៍សម្រាប់រក្សាទិន្នន័យក្នុងកំឡុងពេលដែលគេចូលមក កាន់ Web Application រហូតដល់គេចាកចេញ ផ្ទុកខាង server-side ។

Ex:

```
< ?php
session_start( );    // Create Session

$_SESSION [ 'name' ] = "Sok San" // Put data

?>
```

```
<html><body>

< ?php print_r ( $_SESSION );    // Read data ?>

</body></html>
```

8. Upload File វិធីសាស្ត្រមួយ ដែលមានប្រយោជន៍សម្រាប់រក្សាទិន្នន័យទុកជាអចិន្ត្រៃយ៍ ខាង Server ។

Ex:

```
<html>

<body>

<form action= “upload_file.php” method= “POST” enctype= “multipart/form-data”>

    <input type= “file” name= “file” />

    <input type= “submit” value= “submit” />

</form>

</body></html>

< ?php

    if ( isset ( $_POST [‘submit’] ) )

    { echo “Stored in: ” . $_FILE [‘file’][‘tmp_name’]; }

?>
```

12. របៀបភ្ជាប់ Database ជាមួយ PHP

1. Procedure:

```
mysqli_connect ( $servername, $username, $password, $dbname );
```

Ex:

```

<?php

$conn = mysqli_connect ( "localhost", "root", "", "mydb" )

if ( !$conn )

{
    die ( "Connection failed: " . mysqli_connect_error ( ) ); }

echo "Connected successfully";

mysqli_close ( $conn );

?>

```

2. Object-oriented

```

new mysqli ( $servername, $username, $password, $dbname )

```

Ex:

```

<?php

$conn = new mysqli ( "localhost", "root", "", "mydb" );

if ( $conn -> connect_error )

{
    die ( "Connection failed: " . $conn -> connect_error ); }

echo "Connected successfully";

$conn->close( );

<?

```

3. PDO – PHP Data Object

```

new PDO ( "mysql:host=$servername", $username, $password );

```

Ex:

```

< ?php

    try {

        $conn = new PDO (

            "mysql:host=localhost;dbname=mydb", "root", "" );

        $conn -> setAttribute

            ( PDO::ATTR_ERRORMODE, ERRORMODE_EXCEPTION );

        echo "Connected successfully";

    } catch ( PDOException $e )

    {

        die ( "Connection failed:" . $e -> getMessage( ) );

    }

    $conn = null

< ?

```

9. ASP.NET ជា Development Framework មួយរបស់ Microsoft សម្រាប់លាយជាមួយ HTML បង្កើត Dynamic Server-Side Web Page ដោយប្រើ MVC Pattern ។

10. MVC Pattern បែងចែកការងារនៃ Components ក្នុង User Interface Layer ជាទម្រង់ Model-View-Controller ។ M, C ជាសំណុំ Classes ឯ V ជាសំណុំ Files សម្រាប់ Output

- Controller: សម្រាប់ឱ្យ User Run

Ex: HomeController.cs

```

public ActionResult About( )

{

    ViewBag.id = 1;

    ViewBag.name = "A A";

    return View( );
}

```

```
}
```

- View: សម្រាប់ឱ្យ User មើលឃើញ Output

Ex: **About.cshtml**

```
id=@ViewBag.id
```

```
name=@ViewBag.name
```

- Model: សម្រាប់ភ្ជាប់ Database

Ex: **Book.cs**

```
public class Book {  
  
    public int Code { set; get; }  
  
    public string Title { set; get; }  
  
    public Book( int c, string t ) { Code = c; Title = t; }  
  
}
```

HomeController.cs

```
public ActionResult ListBook( )  
  
{  
  
    Book[] b = new Book[] { new Book( 1, "Java" );  
  
    return View( b );  
  
}
```

ListBook.cshtml

```
@Html.DisplayNameFor( model => model.Code )
```

```
@Html.DisplayNameFor( model => model.Title )
```

11. Entity Framework គឺជា Object-relational Mapper Framework របស់ Microsoft ដែលអាចឱ្យ Developers ធ្វើការជាមួយ Database ដោយប្រើ .NET objects និងកាត់បន្ថយការសរសេរកូដ ក្នុងការ access data ។

13. ប្រៀបធៀប PHP និង ASP

PHP

- ជា Scripting language run ខាង Server Side
- លាយជាមួយ HTML ដើម្បីបង្កើត Dynamic Server-side Web Page
- សម្រាប់បង្កើត Web ដែលមានទំហំ តូចទៅមធ្យម
- ខ្វះមុខងារសុវត្ថិភាព
- ផ្ដោតលើ User Interfaces
- មិនល្អសម្រាប់ដំណើរការ Desktop Applications
- ឥតគិតថ្លៃគ្រប់ Web Applications (Open Source)
- ងាយស្រួលរៀន និងស្វែងយល់

ASP.NET

- ជា Web Application Framework Technology របស់ Microsoft
- លាយជាមួយ HTML បង្កើត Dynamic Server Side ដោយប្រើ MVC Pattern
- សម្រាប់បង្កើត Applications ដែលមានទំហំ មធ្យមទៅធំ
- ផ្ដោតជាចម្បងលើ សុវត្ថិភាព និងមុខងារ
- មានសុវត្ថិភាពខ្ពស់
- កម្មវិធី Desktop Applications ដំណើរការលឿន
- ត្រូវការចំណាយលើ License (Paid Framework)
- ពិបាកក្នុងការរៀន សម្រាប់ Beginner

14. List 5 of Server-Side Technology

- PHP
- ASP
- ASP.NET
- JSP
- Node.js
- Python

លំហាត់

1. បង្កើត Form សម្រាប់ Insert ទិន្នន័យ

```
<html>
  <body>
    <h1>Please Enter Data</h1>
    <form action="../../php/insert_book.php" method="POST">
      ID<input name="id" /><br />
      Name<input name="name" /><br />
      Email<input name="email" /><br />
      Password<input name="pass" /><br />
      <input type="submit" value="Add" />
    </form>
  </body>
</html>
```

```
<?php
$id = $_POST['id'];
$name = $_POST['name'];
$email = $_POST['email'];
$pass = $_POST['pass'];

$server = "exam2021";
$user = "rupp";
$password = "123456";
$dbname = "dbexam2021";
$tbname = "tbUser";

$con = new mysqli($server, $user, $password, $dbname);
```

```

$sql = "INSERT INTO $tbname(id, name, email, pass) VALUES (
    '$id', '$name', '$email', '$pass'
)";
$con->query($sql);
$con->close();

```

2. បង្ហាញទិន្នន័យ

```

<h1>List of Users</h1>
<?php
$con = new mysqli($server, $user, $password, $dbname);
if($con->connect_error){
    die("can not connect to backend database." . $con->connect_error);
}
else{
    $sql = "SELECT * FROM $tbname";
    $rows = $con->query($sql);
    echo "<table border=1>";
    while ($row = $rows->fetch_assoc()) {
        echo "<tr>";
        echo "<td>" . $row["id"];
        echo "<td>" . $row["name"];
        echo "<td>" . $row["email"];
        echo "<td>" . $row["pass"];
        echo "<td><a>Edit</a>";
    }
    echo "</table>";
?>
<a href="add_book.html">Add</a>

```