

សំណួរ

🐞 _QUESTION_.php

1 ពន្យល់ត្រួសពី

- 2 - Front-end dynamic web site
- 3 - Back-end dynamic web site
- 4 - MVC parttern
- 5 - routing
- 6 - Model, View, and Controller interaction in ASP.NET MVC
- 7 - numeric array
- 8 - associative array
- 9 - include
- 10 - include_once
- 11 - include_require

12 រៀបរាប់ពីផលប្រយោជន៍និងផលលំបាកនៃ

- 13 - file
- 14 - \$_POST
- 15 - \$_GET

16 ប្រៀបធៀប

- 17 - \$_POST និង \$_GET
- 18 - Cookie និង Session
- 19 - file និង database

20 សរសេរកូដ

- 21 - បង្កើត Cookie

22 - ទាញយក Cookie មកប្រើ

23 - លុប Cookie ចោល

24 - បង្កើត Session

25 - ទាញយក Session មកប្រើ

26 - លុប Session ចោល

27 - បង្កើតអនុគមន៍ dbConnect ដើម្បី connect ទៅកាន់ MySQL Database

28 - បង្កើតអនុគមន៍ dbSelect ដើម្បីទាញទិន្នន័យពី MySQL Database

29 - បង្កើតអនុគមន៍ dbInsert ដើម្បី insert មួយ record ចូលក្នុង MySQL Database

30 - បង្កើតអនុគមន៍ dbUpdate ដើម្បី update មួយ record ដែលមានស្រាប់ក្នុង MySQL Database

31 - បង្កើតអនុគមន៍ dbDelete ដើម្បី delete មួយ record ចេញពី MySQL Database

27

ចម្លើយ

1- ពន្យល់ត្រួសៗពី

- Front-end dynamic web site គឺជា Code ដែល User មើលឃើញ និងអាច បត់បែន តាមឧបករណ៍ដែល User ប្រើសម្រាប់បើក web site ដែលយើងប្រើ HTML, JavaScript, JQuery, CSS ជាដើមសម្រាប់ Develop។

Ex. នៅពេលយើងបើកមើល Web site មួយយើងអាច View Code HTML, JavaScript, JQuery, CSS បានហើយអាច Copy ដាក់ក្នុង Website របស់យើងបាន ដោយគ្រាន់តែ Inspect element។

- Back-end dynamic web site គឺជា Code ដែលយើងសរសេរភ្ជាប់ទៅកាន់ Database ដើម្បីយកទិន្នន័យទៅរក្សាទុកក្នុង Database ឬទាញយកពី Database មកបង្ហាញនៅលើ web site ដោយប្រើ MySQL Database ដើម្បីអោយ User អាច កែប្រែ លុប បន្ថែម បាន ប៉ុន្តែ User មិនអាចមើលឃើញ Code របស់យើងបានទេ ហើយយើងប្រើប្រាស់ PHP, MySQL Database ជាដើមសម្រាប់ Develop។

Ex: គេមិនអាច View or inspect code របស់ Back-end មើលបានទេ

- MVC Pattern គឺជារបៀបដែលគេរៀបចំ program មួយដែលវាចែកចេញជា3ផ្នែក៖
 - + Data(model) ជាអ្នកគិត, អ្នកធ្វើការ, អ្នក Process ហើយផ្តល់ចំណើយ, អ្នករៀបចំ, វាទទួលព័ត៌មានពី View ហើយនិងផ្តល់ចំណើយទៅអោយ View ។
 - + Interface(view) ជាអ្នកផ្តល់និងទទួលព័ត៌មាន , User Interface, វាធ្វើការជិតស្និទ្ធ ជាមួយ Model , វាផ្តល់ព័ត៌មានទៅអោយ Model ។
 - + Application Logic (Controller) ជាអ្នកគ្រប់គ្រងវាគ្រប់គ្រង View and Model, តែ ភាគច្រើនវាធ្វើការជិតស្និទ្ធជាមួយនិង View ។

- Routing គឺជាផ្លូវដឹកនាំ HTTP request ទៅកាន់ Controller អោយត្រូវនិង Action របស់វា។

Ex: យើងចូលទៅកាន់ URL បន្ទាប់មកត្រូវ សរសេរ ឈ្មោះ controller/ឈ្មោះ method /
ឈ្មោះ parameter => url:"{controller}/{action (method)}/{id}"

- Model គឺជាស្រទាប់ដែលមានតួនាទីក្នុងការគ្រប់គ្រង Data ទាំងអស់ ដែលធ្វើការ Transfer នៅក្នុង Control និង View។ Model ក៏មានតួនាទីធ្វើការភ្ជាប់ទៅកាន់ Database ផងដែរ។

- View គឺជាស្រទាប់ដែលផ្តល់ User Interface Logic សម្រាប់ Application តាមការ Request ណាមួយ។

- Contronller គឺជាស្រទាប់ដែលផ្តល់អោយនូវ Interface រវាង View និង Model ឬ Business Logic ដែលមាននាទីគ្រប់គ្រង Process នៅលើ Application ទាំងអស់រួមទាំងទទួល Request និង Respond ណាមួយត្រឡប់ពី View វិញ។

- Numeric Array គឺជា Array មួយដែលអាចផ្ទុក Number String និង Object ប៉ុន្តែ

Index របស់វា និងតំណាងដោយលេខ ហើយ Default Array Index ចាប់ផ្តើមពីលេខសូន្យ។

Ex: ដើម្បីប្រើ Numeric Array យើងត្រូវ បង្កើត Variable មួយ អោយផ្ទុកធាតុរបស់យើង

ហើយ ដាក់ index អោយវា។

Numeric Arrays ដោយប្រើ index

```
<h2 class="ps-4">Numeric Arrays</h2>
<?php $student_name = ["My Daly","Em Pey","Bi Dyna"]?>
<h3 class="ps-4">Access Numeric Arrays -index</h3>
<ul class="ps-5">
  <li><?=$student_name[2]?></li>
  <li><?=$student_name[1]?></li>
  <li><?=$student_name[0]?></li>
</ul>
```

- Associative array គឺវាស្រដៀងទៅនិង Numeric Array ដែលនៅក្នុង Term of Functionality ប៉ុន្តែវាខុសគ្នាត្រង់ Term of Index។ Associative array វាអាចមាន index

String ដូចនេះ វាអាចបង្កើត Strong Associative រវាង Key និង value នៅក្នុង Associative របស់វា។

Ex: - Access Numeric Arrays ដោយប្រើ key

- ដើម្បីប្រើ យើងត្រូវ បង្កើត Variable មួយ អោយផ្ទុកធាតុ ហើយធាតុនីមួយៗមានសម្គាល់ដោយដាក់ Key មិនអោយស្មនគ្នា រួចបង្ហាញវាដោយ variable ដែលយើងបានប្រកាស រួចបញ្ជាក់នូវ Key ដែលយើងបានកំណត់។

```
<h2 class="ps-4">Associative Arrays</h2>
<?php $student_name = ["ND"=>"My Daly","EP"=>"Em Pey","BD"=>"Bi Dyna"]?>
<h3 class="ps-4">Access Associative Arrays -key</h3>
<ul class="ps-5">
    <li><?=$student_name["ND"]?></li>
    <li><?=$student_name["EP"]?></li>
    <li><?=$student_name["BD"]?></li>
</ul>
```

- Include គឺជា Keyword ប្រើសម្រាប់ បង្គប់ File ដែលមានរួចហើយ និង Copy file ដែលមានរួចហើយ ទៅក្នុង File ផ្សេងទៀត ដោយប្រើ Keyword Include។
Ex: <?php include 'footer.php';?>
- Include_once គឺជា Keyword ប្រើសម្រាប់ បង្គប់ File ដែលមានរួចហើយ និង Copy file ដែលមានរួចហើយ ទៅក្នុង File ផ្សេងទៀត ដោយប្រើ Keyword Include_once ប្រសិនបើវាបាន include ម្តងរួចហើយ នោះវាមិន include ម្តងទៀតទេ ប្រសិនបើវាស្វែងរក File មិនឃើញនោះ វានឹងបង្ហាញ Error ហើយ Program ដំណើរធម្មតា។
Ex: <?php include_once 'footer.php';?>
- Require គឺជា Keyword ប្រើសម្រាប់ បង្គប់ File ដែលមានរួចហើយ និង Copy file ដែលមានរួចហើយ ទៅក្នុង File ផ្សេងទៀត ដោយប្រើ Keyword require ប្រសិនបើវាស្វែងរក File មិនឃើញ នោះ វានឹងបង្ហាញ Error ហើយ Program បញ្ឈប់ដំណើរធម្មតា។
Ex: <?php require 'footer.php';?>

- ពន្យល់ពី query method ក្នុង mySQLiConnection។

ចំ៖ query method វាដំណើរការបាននៅពេលដែលយើងភ្ជាប់ជាមួយ Database បានរួចរាល់ជោគជ័យ ហើយ

យើងប្រើ query method ដើម្បីធ្វើការស្នើ ដោយប្រើសំណើដែលយើងបានរៀប ឬ ដើម្បីអនុវត្តការស្នើ ដោយប្រើសំណើដែលយើងបានរៀប។

➢ ករណីប្រើជាមួយ Insert Update Delete

នោះ query method គឺវាអោយលទ្ធផលពីរគឺ error=fail និង Success=true

- ប្រសិនបើលទ្ធផលវាអោយមក failure មានន័យថាការ Insert Update Delete គឺបរាជ័យ

យើងត្រូវបញ្ជាក់ប្រាប់គេថា Error លើអ្វី ហើយ error មានពីរដែលយើងត្រូវបញ្ជាក់

- ត្រូវបញ្ជាក់ Error ID អ្វី ដោយប្រើ Errno
- ហើយបញ្ជាក់ Error នៅដំណើរការមួយណាដូចជា Insert, Update, Delete ណាមួយ ដោយប្រើ Error

Echo "<p>Error: \$cn->errno, Description: \$cn->error </p>"

➢ ករណីប្រើជាមួយ Select

- query method គឺវាអោយលទ្ធផលជា result object (សម្រាប់បញ្ជាក់ពីលទ្ធផល ដែលបានទាញចេញពីតារាង នៃ Database) ហើយដើម្បីដឹងពីលទ្ធផលវា យើងត្រូវប្រើ property num row

- num_rows គឺជា property ដែលរាប់ចំនួនជួរដេកនៅក្នុង result object

✓ ប្រសិនបើ num_row ស្មើសូន្យ មានន័យថា ទាញមិនបានទេ ពីព្រោះនៅក្នុងតារាងគ្មាន data

✓ ប្រសិនបើ num_row ធំជាងសូន្យ មានន័យថា ទាញបាន នោះយើងត្រូវប្រើ method

fetch_assoc ប្រើសម្រាប់ ទាញយកមួយជួរដេក ចេញពី Result Object ដាក់ចូល associate array ហើយវាមានKey សម្រាប់សម្គាល់ធាតុ Array

2- រៀបរាប់ពីផលប្រយោជន៍ និងផលវិបាក

I រៀបរាប់ពីវគ្គប្រឡង និង ផលសំខាន់

វគ្គប្រឡង	ផលសំខាន់
- file.	<ul style="list-style-type: none"> - file មានក្នុង Upload-server. - Upload ឯកសារប្រឡង - file អាចកំណត់បាន -
- \$-post :	<ul style="list-style-type: none"> - មានសុវត្ថិភាព - ផ្ញើ Data លាប ឡូកាលី, number of Variable. និង មានសំណើ ប្រើប្រាស់ ។
- \$-Get :	<ul style="list-style-type: none"> - អាច retrieve Data. និង Identifies លេខ request URL - អាចដើរលើក្នុង Browser History. - អាចស្រួល request data. -
	<ul style="list-style-type: none"> - low performance. - មិនអាច Bookmarked - Parameter មិនត្រូវបានកម្រិតក្នុង Web Browser History. - មិនអាច cacheable. - មិនអាច use. space tap, carriage, return...។ - មិនអាច ប្រើប្រាស់ send, word, document or image. - មិនអាច ប្រើប្រាស់ retrieve Data. - មិនអាច ប្រើប្រាស់កូដ Username - Password. - ប្រើប្រាស់កំណត់ URL

3- ប្រៀបធៀប

- ប្រៀបធៀប

* post

- តម្លៃប៉ង Visible លើ URL
- ប៉ងបាត់កំណត់សម្គាល់ប្រព័ន្ធនៃតម្លៃ
- បាត់ប្រភេទ performance នាឈរ
ចំណាយពេលវេលាច្រើននៅក្នុង Include
Post លើក្នុង Body HTTP.
- Support ច្រើនប្រភេទនៃ data type.
- មិនអាចកាច់ Cacheable.
- ប៉ងរក្សាទុកក្នុង Browser History.

COOKIES.

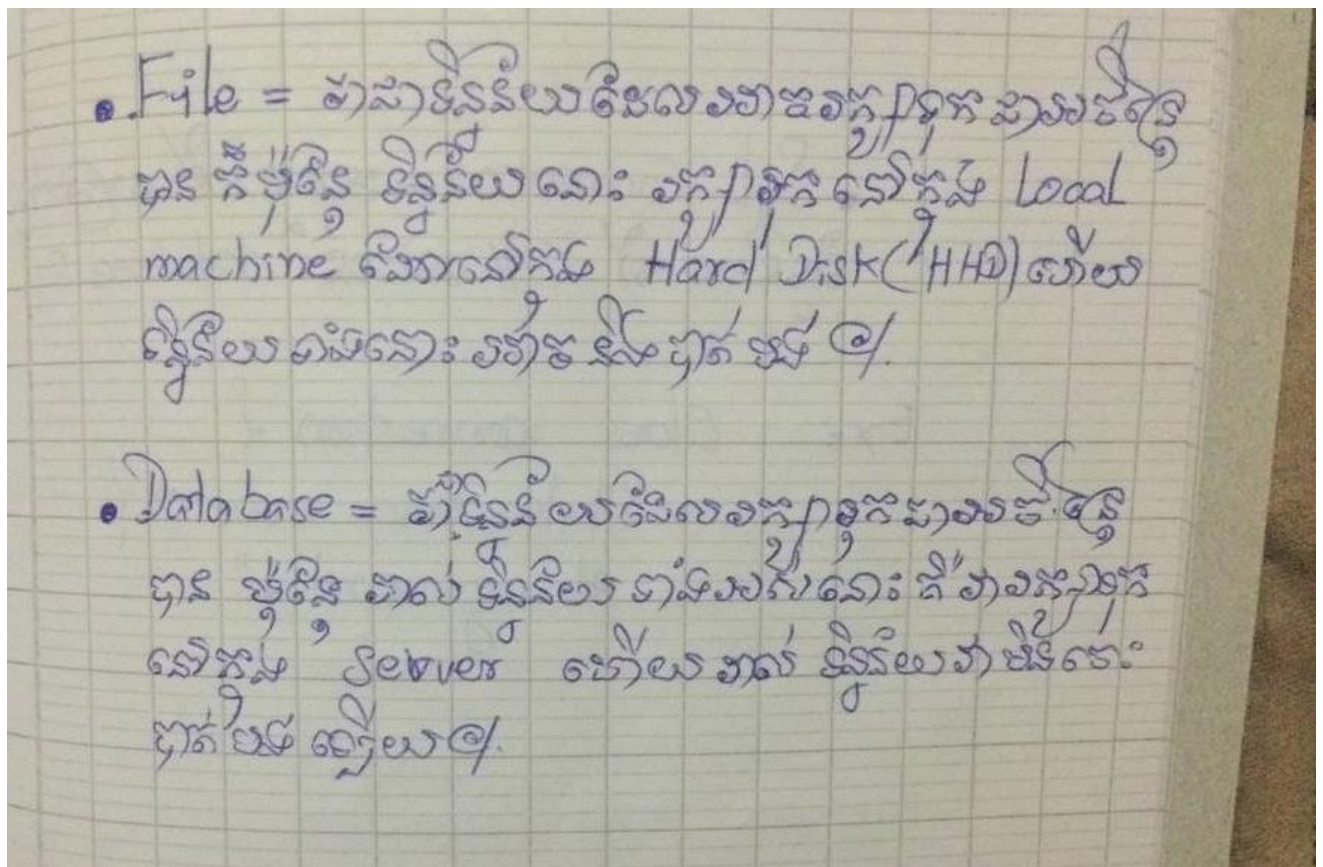
- + រក client-side file លើ local.
- + Set By user.
- + អាចកំណត់ into ទិន្នន័យ Data.
- + Cookies មាន capacity 4KB
- + មិនអាចកាច់ local. ប៉ងរក្សាទុក.
- run Script ដើម្បី start

* get

- តម្លៃ Visible លើ URL
- បាត់កំណត់សម្គាល់ប្រព័ន្ធនៃ Value. បាត់ប្រភេទ
មាន 255 តួ.
- បាត់ប្រភេទ performance ទាំងស្រុង.
- Support តែ String
- Cacheable
- មិនអាចកាច់ Browser History.

Session.

- + រក server-side file. លើ user.data
server.
- + Set នៅលើ programming
- + ប៉ងកំណត់ប្រព័ន្ធនៃ
- + មាន 128 MB.
- + ដើម្បី start នៅក្នុង method
start().



4- លំហាត់

សំណួរ១៖ បង្កើតអនុគមន៍ dbConnect ដើម្បី connect ទៅកាន់ MySQL Database

ចំ៖

```

<?php
$servername = "localhost";
$username = "username";
$password = "";
$dbname = "myDB";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

echo "Connected successfully";

mysqli_close($conn);

?>

```


សំណួរ២៖ បង្កើតអនុគមន៍ dbSelect ដើម្បីទាញយកទិន្នន័យពី MySQL Database

ចំ៖

```
<?php
```

```
$servername = "localhost";
$username = "username";
$password = "";
$dbname = "myDB";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$sql = "SELECT * FROM tb";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
    // output data of each row
    while($row = $result->fetch_assoc()) {
        echo "id: " . $row["id"]. " - Name: " . $row["firstname"]. " " .
        $row["lastname"]. "<br>";
    }
} else {
    echo "0 results";
}
$conn->close();
?>
```

សំណួរ៣៖ បង្កើតអនុគមន៍ dbInsert ដើម្បី Insert មួយ Record ចូលទៅក្នុង MySQL Database

ចំ៖

```
<?php
```

```
$servername = "localhost";
$username = "username";
$password = "";
$dbname = "myDB";
```

```

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$sql = "INSERT INTO tb (firstname, lastname, email)
VALUES ('John', 'Doe', 'john@example.com')";

if ($conn->query($sql) === TRUE) {
    echo "New record created successfully";
} else {
    echo "Error: " . $sql . "<br>" . $conn->error;
}

$conn->close();

```

?>

សំណួរ៤៖ បង្កើតអនុគមន៍ dbUpdate ដើម្បី Update មួយ Record ដែលមានស្រាប់ក្នុង MySQL Database

ចំ៖

<?php

```

$servername = "localhost";
$username = "username";
$password = "";
$dbname = "myDB";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$sql = "UPDATE MyGuests SET lastname='Doe' WHERE id=2";

if ($conn->query($sql) === TRUE) {
    echo "Record updated successfully";
}

```

```

    } else {
        echo "Error updating record: " . $conn->error;
    }

    $conn->close();

```

?>

សំណួរ៥៖ បង្កើតអនុគមន៍ dbUpdate ដើម្បី Update មួយ Record ដែលមានស្រាប់ក្នុង MySQL Database

ចំ៖

<?php

```

$servername = "localhost";
$username = "username";
$password = "";
$dbname = "myDB";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// sql to delete a record
$sql = "DELETE FROM MyGuests WHERE id=3";

if ($conn->query($sql) === TRUE) {
    echo "Record deleted successfully";
} else {
    echo "Error deleting record: " . $conn->error;
}

$conn->close();

```

?>