

1. Front-end និង Back-end គឺជាពាក្យដែលប្រើដើម្បីពិពណ៌នាអំពីស្រទាប់ផ្សេងៗគ្នាដែលបង្កើតជាកម្មវិធីកុំព្យូទ័រ ឬគេហទំព័រ។ Front-end ត្រូវបានគេស្គាល់ថាជា client-side គឺជាអ្វីដែល user ឃើញ និងធ្វើអន្តរកម្មជាមួយ ដូចជា Graphical user interface(GUI) និង command line។

Back-end ត្រូវបានគេស្គាល់ផងដែរថាជា Server-Side គឺជាអ្វីគ្រប់យ៉ាងដែលកើតឡើង “Under the hood” ហើយសមាសធាតុរបស់វាមិនច្បាស់ភ្លាមៗចំពោះ User នោះទេ។

Back-end ទទួលខុសត្រូវចំពោះការរក្សាទុកទិន្នន័យ និង Business logic ។

- Front-end ផ្ដោតលើទិដ្ឋភាពនៃ Website ផ្នែកដែល User មើលឃើញ និងធ្វើអន្តរកម្មជាមួយ។

- Back-end ផ្ដោតលើរូបមាន structure, data and logic.

- Basic languages សម្រាប់ការអភិវឌ្ឍន៍ផ្នែកខាងមុខគឺ HTML, CSS, និង JavaScript។ ម្យ៉ាងវិញទៀត អ្នកអភិវឌ្ឍន៍ផ្នែកខាងក្រោយប្រើភាសាដូចជា Java, Python, Ruby on Rails, PHP និង .NET។

2. Static websites គឺជា basic Website ដែលមានមាតិកាថេរលេចឡើងដូចគ្នាសម្រាប់ User ទាំងអស់ដែលចូលប្រើ ហើយផ្លាស់ប្តូរតែនៅពេលដែល developer កែប្រែ source files ប៉ុណ្ណោះ។ ម្យ៉ាងវិញទៀត Dynamic websites គឺជា website កម្រិតខ្ពស់ដែលផ្តល់នូវមាតិកាខុសៗគ្នាទៅតាមសំណើរបស់អតិថិជន។ ពួកគេត្រូវបានគាំទ្រដោយ traditional server និង database ដែលធ្វើឲ្យពួកគេប្រសើរជាងមុនក្នុងការបម្រើមាតិកាផ្ទាល់ខ្លួន ឬតាមពេលវេលាជាក់ស្តែងជាង static counterparts របស់ពួកគេ។

3. នៅពេលយើងបើក website – web browser និងផ្ញើសំណើទៅកាន់ web server ដែលគ្រប់គ្រង website ។ បន្ទាប់មក web server នឹងដំណើរការសំណើ ហើយបញ្ជូនឯកសារដែលបានស្នើសុំត្រឡប់ទៅ web server របស់អ្នកវិញ។ បន្ទាប់មកឯកសារត្រូវបានបង្ហាញដោយ របស់ web browser អ្នករួចបង្ហាញនៅលើអេក្រង់របស់អ្នក។ Client-server model ត្រូវបានប្រើនៅក្នុង

process នេះ។ The client គឺជាកម្មវិធី web browser របស់អ្នក ហើយ server គឺជា web server ដែលបង្ហាញ website ។

4. PHP តំណាងឱ្យ Hypertext Preprocessor និងជាភាសាស្រ្តីបឋមម៉ាស៊ីនមេ។ វាត្រូវបានបង្កើតឡើងដោយ Rasmus Lerdorf ហើយបានបង្ហាញខ្លួននៅក្នុងឆ្នាំ 1995។ PHP គឺជាវេទិកាឯករាជ្យ ហើយដំណើរការលើគ្រប់វេទិកាទាំងអស់ មិនថាវាជា Mac, Windows ឬ Linux។

+ គុណសម្បត្តិក្នុងការប្រើប្រាស់ PHP ៖

- វាឥតគិតថ្លៃ។
- វាងាយស្រួលរៀននិងប្រើ។
- វាមានសហគមន៍អភិវឌ្ឍន៍ដ៏ធំ។
- វាជាប្រភពបើកចំហ។
- វាគាំទ្រមូលដ្ឋានទិន្នន័យជាច្រើន។

+ គុណវិបត្តិនៃការប្រើប្រាស់ PHP ៖

- បញ្ហាសុវត្ថិភាព
- ដំណើរការមិនល្អបើប្រៀបធៀបទៅនឹងភាសាផ្សេងទៀត។
- កង្វះការគាំទ្រសម្រាប់ multithreading ។

5. web dynamic client គឺជាកម្មវិធីផ្នែក client side ដែលដំណើរការលើកុំព្យូទ័រ ឬឧបករណ៍របស់ user ហើយ interacts ជាមួយ web server ដើម្បីបង្ហាញខ្លឹមសាររបស់ web dynamic client ។

ឧទាហរណ៍: web dynamic client រួមមានកម្មវិធីរុករកតាមអ៊ីនធឺណិតដូចជា Google Chrome, Mozilla Firefox និង Microsoft Edge ។

-A web dynamic server is a server-side application that run on a web server and creates dynamic content in response to clients requests.

ឧទាហរណ៍: web dynamic client រួមាន Apache Tomcat, Microsoft IIS និង Nginx ។

ភាពខុសគ្នាសំខាន់រវាង web dynamic client និង web dynamic server គឺថា former ទទួលខុសត្រូវក្នុងការបង្ហាញ dynamic content នៅផ្នែក client-side ក្រោយមកទៀតបង្កើត dynamic content នៅលើផ្នែក server-side។

6. MVC តំណាងឱ្យ Model-View-Controller ។ វាគឺជាគំរូស្ថាបត្យកម្មដែលបំបែកកម្មវិធីទៅជាសមាសធាតុ logical សំខាន់ៗបី៖

Mode ៖ Back-end ដែលមាន data logic ទាំងអស់។

View៖ front-end ឬ graphical user interface (GUI)

Controller ៖ the brains of the application that controls how data is displayed.

7. Model-View-Controller (MVC) គឺជា software design pattern ដែលបំបែកកម្មវិធីទៅជាសមាសភាគសំខាន់ៗបីដែលទាក់ទងគ្នា៖ model, view, controller។ អត្ថប្រយោជន៍ចម្បងនៃការប្រើប្រាស់ស្ថាបត្យកម្ម MVC គឺវាមានភាពហ័សសារធាន និងដំណើរការអភិវឌ្ឍន៍ស្របគ្នា។ ឧទាហរណ៍ប្រសិនបើ developer ម្នាក់កំពុងធ្វើការលើ View នោះអ្នកផ្សេងទៀតកំពុងធ្វើការក្នុងពេលដំណាលគ្នានៅលើ Controller ។ វាជួយកាត់បន្ថយបន្ទុកការងារលើ developer ម្នាក់ៗ និងធ្វើឱ្យដំណើរការអភិវឌ្ឍន៍លឿន និងមានប្រសិទ្ធភាព។

8. នៅក្នុង PHP មានតែ scopes ពីរប៉ុណ្ណោះដែលអាចប្រើបាន៖ local and global scopes។

- Local scope ៖ អថេរដែលត្រូវបានកំណត់នៅក្នុង "body" នៃអនុគមន៍ (ឧ.i.e., the area between the curly brackets) មានសុពលភាពតែនៅក្នុងតំបន់នេះប៉ុណ្ណោះ។ នៅពេលមុខងារត្រូវបានបញ្ចប់ អថេរនៅក្នុង scope របស់វាត្រូវបានលុបដោយស្វ័យប្រវត្តិ។

- Global scope៖ អថេរដែលត្រូវបានកំណត់នៅក្រៅមុខងារមួយអាចប្រើបាន(ឧ i.e., accessed) ទាំងខាងក្នុង និងក្រៅមុខងារ។

```
<?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>";
```

```
?>
```

9. Global keyword ត្រូវបានប្រើដើម្បីចូលប្រើ global variable ពីក្នុងមុខងារមួយ។ ដើម្បីធ្វើវា ដូច្នេះប្រើ Global keyword មុនអថេរ (នៅក្នុង function)។ PHP ក៏រក្សាទុក global variable ទាំងអស់នៅក្នុង array ដែលហៅថា \$GLOBALS[index]។ Index រក្សាឈ្មោះរបស់អថេរ។

+EX :

```
<?php
```

```
$x = 5;
```

```
$y = 10;
```

```
function myTest() {
    global $x, $y;
    $y = $x + $y;
}
```

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

- static keyword ត្រូវបានប្រើដើម្បីប្រកាសលក្ខណសម្បត្តិ និងវិធីសាស្ត្រនៃថ្នាក់ជាប្រភេទ  
នៃវត្ថុ។ លក្ខណៈសម្បត្តិ និងវិធីសាស្ត្រប្រភេទអាចប្រើដោយមិនចាំបាច់បង្កើតឧទាហរណ៍នៃថ្នាក់។  
ពាក្យគន្លឹះប្រភេទត្រូវបានប្រើដើម្បីប្រកាសអថេរនៅក្នុងមុខងារដែលរក្សាតម្លៃរបស់ពួកគេបន្ទាប់ពី  
មុខងារបានបញ្ចប់។

+ EX :

```
<?php
```

```
function testStatic() {
    static $x = 0;
    echo $x;
    $x++;
}
```

```
testStatic();
testStatic();
testStatic();
?>
```

10. នៅក្នុង PHP, echo និង print ត្រូវបានប្រើដើម្បីបញ្ជូនទិន្នន័យទៅកាន់អេក្រង់។ ភាពខុសគ្នា គឺ small ៖ echo មិនមានតម្លៃត្រឡប់មកវិញទេ ខណៈពេលដែល print មានតម្លៃត្រឡប់ 1។ ដូច្នេះ វាអាចប្រើក្នុងកន្សោមបាន។ echo អាចយកប៉ារ៉ាម៉ែត្រជាច្រើន (ទោះបីជាការប្រើប្រាស់បែបនេះក៏ម្រ មានក៏ដោយ) ខណៈពេលដែល print អាចយក argumentecho មួយគឺលឿនជាង print បន្តិ ច។

```
+EX : <?php
$name = "John";
$age = 25;
echo "My name is $name and I am $age years old.";
print "<br>";
print "My name is " . $name . " and I am " . $age . " years old.";
?>
```

11. នៅក្នុង PHP, \$\_POST និង \$\_GET ទាំងពីរត្រូវបានប្រើដើម្បីប្រមូល data ពីទម្រង់ HTML និង URLs ។ ភាពខុសគ្នារវាងពួកវាគឺថា \$\_POST បញ្ជូនទិន្នន័យដែលត្រូវដំណើរការ (ឧ. ពីទម្រង់ HTML) ទៅ resource ដែលបានកំណត់។ data ត្រូវបានរួមបញ្ចូលនៅក្នុង request អាចបណ្តា លឱ្យមានការបង្កើត resource ថ្មី ឬការ updates resource ដែលមានស្រាប់ ឬទាំងពីរ។ ម្យ៉ាង វិញទៀត \$\_GET ទាញយក data ពីចម្ងាយ។ វាត្រូវបានប្រើដើម្បីស្នើសុំ data ពី resource ដែល បានបញ្ជាក់។ data ត្រូវបានផ្ញើជាខ្សែសំណួរដែលភ្ជាប់ទៅនឹង URL។

-example \$\_POST

```
<form method="post" action="process.php">
  <label for="name">Name:</label>
  <input type="text" id="name" name="name"><br><br>
  <label for="email">Email:</label>
  <input type="text" id="email" name="email"><br><br>
```

```

        <input type="submit" value="Submit">
    </form>

    -example $_GET

    <a href="process.php?name=John&email=john@example.com">Submit</a>

```

12. Write code PHP to allow upload multiple files (only image) to server

```

<form method="post" action="" enctype="multipart/form-data" id="frmImgUpload">

    <input name="multiple_uploaded_files[]" type="file" multiple="true" />

</form>

```

```

foreach ($_FILES['multiple_uploaded_files']['name'] as $file) {
    print_r($file);

    //use the move_uploaded_file() to move your file on your server directory.

    //fire an insert query that inserts all the file names with comma separated value
}

```

13. ភាពខុសគ្នារវាង cookies និង session៖

- cookies គឺជាឯកសារភាគី client-side ដែលមានព័ត៌មាន user នៅលើកុំព្យូទ័រមូលដ្ឋាន។ ពួកវាត្រូវបានប្រើដើម្បីរក្សាទុកទិន្នន័យមួយចំនួនតូចដែលអាចចូលប្រើបាននៅទំព័រផ្សេងៗគ្នានៃ website ។ cookies មិនអាស្រ័យលើ session ទេ។

+EX៖ ការប្រើ cookies គឺនៅពេលដែលអ្នកចូលទៅកាន់ website ទិញទំនិញអនឡាញ ហើយបន្ថែមរបស់របរទៅក្នុង cart របស់អ្នក។ website រក្សាទុកព័ត៌មាននេះនៅក្នុង cookies ដូច្នេះនៅពេលដែលអ្នកចូលទៅកាន់ pages ផ្សេងទៀតនៅលើ website នោះ cart របស់អ្នកនៅតែមាន items ដែលអ្នកបានបន្ថែមពីមុន។

-Sessions គឺជាឯកសារខាង server ដែលរក្សាទុកព័ត៌មានរបស់ user ។ ពួកគេប្រើការកំណត់ session identifier ដើម្បីកំណត់ទីតាំង session data របស់អ្នកប្រើប្រាស់ជាក់លាក់។ Session គឺអាស្រ័យលើ cookies ។

+EX : ការប្រើប្រាស់ sessions គឺនៅពេលដែលអ្នកចូលទៅក្នុង website ។ website រក្សាទុកព័ត៌មានសម្ងាត់នៃការចូលរបស់អ្នកនៅក្នុង session មួយ ដូច្នេះអ្នកមិនចាំបាច់ចូលម្តងទៀតនៅពេលអ្នកចូលមើល pages ផ្សេងទៀតនៅលើ website ។

14. cookies និង session មិនអាចរក្សាទុក binary file ដោយផ្ទាល់នៅក្នុង cookies ឬ session នោះទេ។ ទោះយ៉ាងណាក៏ដោយ អ្នកអាច encode binary file ជា string ហើយបន្ទាប់មករក្សាទុកវានៅក្នុង cookies ឬ session ។

15. Write PHP code to generate the random number with 8 digit (only number)

```
$digits_needed=8;
$random_number="";
$count=0;
while ($count < $digits_needed) {
    $random_digit = mt_rand (0, 9);
    $random_number.= $random_digit;
    $count++;
}
echo "The random $digits_needed digit number is $random_number";
```

16. Write PHP code to write “Hello World!” 1000 line in a file named myfile.txt

```
<?php
```



```
$file = fopen("myfile.txt", "w");

for ($i = 0; $i < 1000; $i++) {
    fwrite($file, "Hello World!\n");
}

fclose($file);
?>
```

17. Write 3 functions to connect, insert, update, and delete data from mysql database respectively :

```
import mysql.connector

def connect_to_database(host, user, password, database):
    mydb = mysql.connector.connect(
        host=host,
        user=user,
        password=password,
        database=database
    )
    return mydb

def insert_data(mydb, table, data):
    mycursor = mydb.cursor()
    columns = ', '.join(data.keys())
```

```
values = tuple(data.values())
placeholders = ', '.join(['%s'] * len(data))
sql = f"INSERT INTO {table} ({columns}) VALUES ({placeholders})"
mycursor.execute(sql, values)
mydb.commit()
```

```
def update_data(mydb, table, data, condition):
    mycursor = mydb.cursor()
    set_clause = ', '.join([f'{key} = %s' for key in data.keys()])
    values = tuple(data.values()) + tuple(condition.values())
    where_clause = ' AND '.join([f'{key} = %s' for key in condition.keys()])
    sql = f"UPDATE {table} SET {set_clause} WHERE {where_clause}"
    mycursor.execute(sql, values)
    mydb.commit()
```

```
def delete_data(mydb, table, condition):
    mycursor = mydb.cursor()
    where_clause = ' AND '.join([f'{key} = %s' for key in condition.keys()])
    values = tuple(condition.values())
    sql = f"DELETE FROM {table} WHERE {where_clause}"
    mycursor.execute(sql, values)
    mydb.commit()
```

18. ភាពខុសគ្នារវាង \$var1 និង \$\$var1:

\$var1 គឺជា variable reference ខណៈដែល \$\$var1 គឺជា reference ទៅកាន់លេខសម្គាល់ដំណើរការនៃ shell បច្ចុប្បន្ន។ ភាពខុសគ្នារវាង \$var1 និង \${var1} គឺក្រោយមកទៀតវាត្រូវបានប្រើនៅពេលដែលអ្នកចង់ពង្រីកអថេរ ហើយភ្ជាប់វាជាមួយតួអក្សរ ឬ string ផ្សេងទៀត។ ឧទាហរណ៍ ប្រសិនបើអ្នកមានអថេរឈ្មោះ filename អ្នកអាចប្រើ \${filename}\_backup ដើម្បីបង្កើតអថេរថ្មីមួយដែលមានឈ្មោះថា filename\_backup។

19. In PHP, there are two ways to define a constant: using the define() function and using the const keyword

+ EX : -use the define() function:

```
define('CONSTANT_NAME', 'constant_value');
```

-use the const keyword:

```
const CONSTANT_NAME = 'constant_value';
```

20. isset() ត្រូវបានប្រើដើម្បីពិនិត្យមើលថាតើអថេរត្រូវបានកំណត់ឬអត់។ ប្រសិនបើអថេរមិនត្រូវបានកំណត់ជាមួយនឹងមុខងារ unset() នោះវានឹងលែងត្រូវបានកំណត់ទៀតហើយ។ អនុគមន៍ isset() នឹងត្រឡប់មកវិញមិនពិត ប្រសិនបើអថេរសាកល្បងមានតម្លៃ NULL ។

21.PHP, \$\_SERVER គឺជា superglobal variable ដែលផ្ទុកព័ត៌មានអំពី headers, paths, script locations។ វាជា array ដែលមានព័ត៌មានដូចជា headers, paths, និង script locations.

- អថេរ \$\_SERVER មានប្រយោជន៍សម្រាប់គោលបំណងបំបាត់កំហុសព្រោះវាផ្តល់ព័ត៌មានអំពីបរិស្ថានដែលស្ត្រីបកំពុងដំណើរការ។ វាអាចត្រូវបានប្រើដើម្បីទទួលបានអាសយដ្ឋាន IP របស់ user ដែលចូលប្រើ website ឈ្មោះ SERVER ដែលកំពុងដំណើរការស្ត្រីប និងព័ត៌មានមានប្រយោជន៍ផ្សេងទៀត។

22. ភាពខុសគ្នារវាង require, require\_once , include, include\_once:

require, require\_once, include, include\_once ត្រូវបានប្រើដើម្បីរួមបញ្ចូលឯកសារនៅក្នុង PHP scripts ។ ភាពខុសគ្នារវាងពួកវាគឺ require និង require\_once ត្រូវបានប្រើដើម្បីរួមបញ្ចូលឯកសារដែលចាំបាច់សម្រាប់មុខងាររបស់ scripts ចំណែក include, include\_once ត្រូវបានប្រើដើម្បីរួមបញ្ចូលឯកសារដែលមិនចាំបាច់។

23. mysqli និង PDO គឺជាផ្នែកបន្ថែមរបស់ PHP ដែលអនុញ្ញាតឱ្យ PHP ទាក់ទងជាមួយ databases ។ ភាពខុសគ្នាសំខាន់ទាំងពីរគឺ mysqli ត្រូវបានប្រើសម្រាប់តែ MySQL databases ហើយ PDO supports multiple databases .

- Mysqli អាចត្រូវបានប្រើទាំង object-oriented style ឬ procedural style ខណៈពេលដែល PDO គឺតែជា object-oriented ។ Mysqli គាំទ្រសេចក្តីថ្លែងការណ៍ដែលបានរៀបចំជាមួយ? placeholders for parameters ខណៈពេលដែល PDO បានដាក់ឈ្មោះប៉ារ៉ាម៉ែត្រ។

- PDO មានសុវត្ថិភាពជាង mysqli ព្រោះវាមានការការពារប្រឆាំងនឹងការវាយប្រហារ SQL ។ PDO ល្បឿនជាង mysqli procedural និង mysqli object-oriented ។

24/ Object-oriented programming (OOP) គឺជាគំរូនៃការសរសេរកម្មវិធីដែលពឹងផ្អែកលើគោលគំនិតនៃ class និង objects ។

- អ្នកអាចបង្កើត class ដោយប្រើ class keyword អមដោយឈ្មោះ class ។ ឧទាហរណ៍ដើម្បីបង្កើត class មួយឈ្មោះថា Person អ្នកនឹងសរសេរ៖

```
class Person {
    // properties and methods go here
}
```

- អ្នកអាចបង្កើត object មួយ (ត្រូវបានគេស្គាល់ផងដែរថាជាឧទាហរណ៍) នៃថ្នាក់នោះ ដោយប្រើពាក្យគន្លឹះថ្មីដែលតាមដោយឈ្មោះនៃថ្នាក់។ ឧទាហរណ៍:

```
$person = new Person();
```

25. អ្នកអាចបង្កើត database មួយដែលមានឈ្មោះថា "mydatabase" ដោយប្រើ PHP និង MySQL ដោយបង្កើតការភ្ជាប់ទៅកាន់ MySQL server ដោយប្រើមុខងារ mysql\_connect()។ បន្ទាប់មកវាពិនិត្យមើលថាតើការតភ្ជាប់បានជោគជ័យឬអត់។ ប្រសិនបើវាកើតឡើង វាបង្កើតមូលដ្ឋានទិន្នន័យថ្មីមួយដែលមានឈ្មោះថា "mydatabase" ដោយប្រើមុខងារ mysql\_query() និង function និង SQL បង្កើតទិន្នន័យ។ ចុងក្រោយ វាបិទការតភ្ជាប់ដោយប្រើ mysql\_close() function។

26. Write a PHP code to send an email :

```
<?php
$to = "example@example.com";
$subject = "Test mail";
$message = "Hello! This is a simple email message.";
$from = "example@example.com";
$headers = "From:" . $from;
mail($to,$subject,$message,$headers);
echo "Mail Sent.";
?>
```

27. ដើម្បីប្រតិបត្តិសំណួរ SQL ក្នុង PHP អ្នកអាចប្រើមុខងារ `mysqli\_query()` ឬ `PDO::query()` ។ ទោះជាយ៉ាងណាក៏ដោយ វាមានសារៈសំខាន់ណាស់ក្នុងការទប់ស្កាត់ការវាយប្រហារដោយ SQL ដោយការបញ្ជាក់ភាពត្រឹមត្រូវនៃការ input របស់ user និងការប្រើប្រាស់ prepared statements ។ នេះជាការអនុវត្តល្អបំផុតមួយចំនួនសម្រាប់ដោះស្រាយការ input របស់ user ដើម្បីការពារការវាយប្រហារដោយ SQL ៖

1. Validate input ៖ Validate the data supplied by the user against a whitelist.
2. Prepare a query ៖ Create your query using parameter names preceded with colons as placeholders.
3. បង្កើត prepared statement ។
4. Bind the parameters to the prepared statement.
5. ប្រតិបត្តិសំណួរ។
6. ទាញយកលទ្ធផល។
7. ធ្វើឱ្យកម្មវិធីមានសុពលភាព។

30. Write to search for a student based on their name and date of birth in a single php file. If the student is found, “Found!” message will be display while if not, “Not Found!” message is shown in the page:

```
<?php
// Connect to the database
$servername = "localhost";
$username = "username";
$password = "password";
```

```
$dbname = "myDB";
```

```
$conn = new mysqli($servername, $username, $password, $dbname);
```

```
// Check connection
```

```
if ($conn->connect_error) {
```

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

```
}
```

```
// Get the name and date of birth from the user
```

```
$name = $_POST["name"];
```

```
$date_of_birth = $_POST["date_of_birth"];
```

```
// Prepare the SQL statement
```

```
$sql = "SELECT * FROM students WHERE name='$name' AND  
date_of_birth='$date_of_birth'";
```

```
// Execute the SQL statement
```

```
$result = $conn->query($sql);
```

```
// Check if any rows were returned
```

```
if ($result->num_rows > 0) {
```

```
    echo "Found!";
```

```
} else {
```

```
    echo "Not Found!";
```

```
}
```

```
// Close the database connection
```

```
$conn->close();
```

```
?>
```