Name: Abdul Munim Khan

ID: K15-2897

WriteUp:

**PHP:**

like Javascript tag php tag is <?php \_\_\_\_\_ ?>

1. Correct opening tag <?php and closing tag is ?>

2. To output we use echo

3. For single line comment we use // and for multiple line /\* \_\_\_ \*/

4. Line Terminator we use ‘;’

Server side scripting language, dynamic and interactive website. Competitors are Microsoft ASP.net.

All variable names are case sensitive, run on server and return simple HTML page on text.

To type of Variable:

Global keyword (static variable), and Local within function or under different aliases {}.

Global is used to define at the top, but nor excess in method. We need to use Global.

PHP stores all Global variables name under $GLOBALS[‘Y’].

PHP Data Types:

1. String

2. Integer

3. Float

4. Boolean

5. Array

6. Object

7. Null

8. Resources

String define by “ ” or by ‘ ’

Integer has three formats. Decimal, hexadecimal and octal.

var\_dump() function use to get floatValues.

Array = $car.array(“BMW”, “Audi”);

Object not need var\_dump(), but float and array is.

**String:**

To check string length we use strlen.(“helloworld”)

String Operations:

str\_word\_count.(“Return Word Count”) //output 3 words

str\_rev.(“Alpha”) //output ahplA

str\_pos.(“HelloWorld”, “World”) //give 6 (starting pointer position)

str\_replace.(“from”, “with” , “fromStirng”) // withString

**PHP Array:**

count($ArrayName) //gives elements in array

arrayAccess using arrayName[index]

Associative Array:

foreach($age=> as $y)

{

echo “Key”. $x . “value:” $y

}

Array Sort Functions:

Sort($arrayname”) → ascending

rSort($arrayname”) → descending

incase of value

aSort($arrayname”) → ascending by value

arSort($arrayname”) → descending by value

incase of key

kSort($arrayname”) → ascending by key

krSort($arrayname”) → descending by key

**PHP Forms:**

PHP Form Handling:

<form> inputs and other activity </form>

we can use PHP for displaying data on backend, like $\_POST["name"]

using <?php ?>

GET and POST Methods:

Both create an array like (Key1 => value1, Key2 => value2, …, key3 => value3)

|  |  |  |
| --- | --- | --- |
|  | GET | POST |
| 1. | Visible to every one | Invisible to others |
| 2. | All variable and values are displayed in URL | All variable and values could be pass through Body tag |
| 3. | Specific amount of information could be pass through URL | Has no limit to send information, under body |
| 4. | Use for non-sensitive data | Support advance functionality of multi-part binary input while uploading files to server. |
| 5. | GET should never be used for sending passwords or other sensitive information. | For secure data passing, POST should be use because information is hidden in body |

PHP Form Validation:

On form validation usually prevent server hits, bandwidth and improve performance.

Rather than defining direct action its better to give php backend to redirect page where it programmer want, but problem exist with htmlspecialchars() to prevent $php form to being exploit.

Do two things when the user submits the form:

1. Strip unnecessary characters (extra space, tab, newline) from the user input data (with the PHP trim() function)

2. Remove backslashes (\) from the user input data (with the PHP stripslashes() function)

PHP Form Required:

Before page hits server check all inputs by user, is it correct else show error message in start.

check all valid thing enter under preg\_match(matching statement)

$name = test\_input($\_POST["name"]);

if (!preg\_match("/^[a-zA-Z ]\*$/",$name)) {

$nameErr = "Only letters and white space allowed";

}

PHP Form URL/E-mail:

As discussed above code:

we can simply check email

$email = test\_input($\_POST["email"]);  
if (!filter\_var($email, FILTER\_VALIDATE\_EMAIL)) {  
 $emailErr = "Invalid email format";   
}

we can simply check URL

$website = test\_input($\_POST["website"]);  
if (!preg\_match("/\b(?:(?:https?|ftp):\/\/|www\.)[-a-z0-9+&@#\/%?=~\_|!:,.;]\*[-a-z0-9+&@#\/%=~\_|]/i",$website)) {  
 $websiteErr = "Invalid URL";   
}

and Furthers things can be checked with little changes in above code snippet.

PHP Form Complete:

## Keep The Values in The Form.

In text box we can show value or hint using Value.

For Reference: Value=”<?php echo $name;?>”

**PHP Advance:**

PHP Arrays Multi:

Multi-dimension Array is array of array.

Sample Two Dimension Array:

$cars = array

(

array("Volvo",22,18),

array("BMW",15,13),

array("Saab",5,2),

array("Land Rover",17,15)

);

And we can access array using array[row\_num][column\_num]

Simple Dates:

we can access dates using date key word with y as year, m as month and d as day.

Different position, different style.

date("Y/m/d")

date("Y.m.d")

date("Y-m-d")

and similarly if you only pass m it only prints month.

We can create time format as required using mktime

sample: mktime(hour,minute,second,month,day,year)

The PHP **strtotime()** function is used to convert a human readable string to a Unix time.

Include vs require:

require must be given else code execution halt, but incase of include system give warning and you may continue.

File Operations:

ReadFile() is used to read file with the start of file.

Fopen() is used to openFile read and write, more options then ReadFile();

Copied and Pasted from W3Schools.

r

Open a file for read only. File pointer starts at the beginning of the file

w

Open a file for write only. Erases the contents of the file or creates a new file if it doesn't exist. File pointer starts at the beginning of the file

a

Open a file for write only. The existing data in file is preserved. File pointer starts at the end of the file. Creates a new file if the file doesn't exist

x

Creates a new file for write only. Returns FALSE and an error if file already exists

r+

Open a file for read/write. File pointer starts at the beginning of the file

w+

Open a file for read/write. Erases the contents of the file or creates a new file if it doesn't exist. File pointer starts at the beginning of the file

a+

Open a file for read/write. The existing data in file is preserved. File pointer starts at the end of the file. Creates a new file if the file doesn't exist

x+

Creates a new file for read/write. Returns FALSE and an error if file already exists

fClose() is used to close file.

Fopen() create file automatically.

To upload file, we should set file php.in for fileUpload = on;

We can check file size and other conditions before uploading file.

Database:

We have two ways to connect from Database.

1. MySQLi (procedural & Object Oriented)

2. PDO

To connect with MySQLi

<?php

$servername = "localhost";

$username = "username";

$password = "password";

// Create connection

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

// Check connection

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

echo "Connected successfully";

?>

To Connect With PDO

<?php

$servername = "localhost";

$username = "username";

$password = "password";

try {

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

// set the PDO error mode to exception

$conn->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

echo "Connected successfully";

}

catch(PDOException $e)

{

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

}

?>

To run Query on MySQLi:

$sql = "INSERT INTO MyGuests (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;

}