

UNIT III

1. Which of the following is not a valid example of variable declaration?
 - a. \$name
 - b. \$address2
 - c. \$colour_30.
 - d. \$2name
2. Give the output for the following code

```
$first_name = 'Chhaya';  
$greeting1 = "Hello, my first name is  
$first_name. ";echo $greeting1;
```

 - a. Hello, my first name is Chhaya.
 - b. Hello, my last name is \$last_name.
 - c. \$greeting1
 - d. Hello, my first name is
3. To concatenate 2 strings which of the following character is used
 - a. .
 - b. ,
 - c. \$
 - d. *
4. _____ are named and indexed collections of other values.
 - a. Strings
 - b. Arrays
 - c. Objects
 - d. Resources
5. _____ hold references to resources external to PHP
 - a. Strings
 - b. Arrays
 - c. Objects
 - d. Resources
6. Which of the following is not a —special datatype?
 - a. Boolean
 - b. Null
 - c. Array
 - d. Object
7. Which of the following is used to print the next character as a dollar not as a part of variable?
 - a. \d
 - b. \$
 - c. \\\$
 - d. \\$
8. Which of the following represents conditional operator
 - a. ?:
 - b. ??

- c. ?;
 - d. ?.
9. Which keyword is used to halt the current iteration of a loop but it does not terminate the loop
- a. **Continue**
 - b. Break
 - c. Halt
 - d. Stop
10. Which Letter is used to represent —AM|| in Date
- a. **A**
 - b. a
 - c. D
 - d. D
11. Which of the following is used to Set the internal pointer of an array to its last element?
- a. count
 - b. current
 - c. next
 - d. **end**
12. Which of the following determines if a variable is declared and is different than **NULL**
- a. **Isset**
 - b. Empty
 - c. Current
 - d. Set
13. An array with string index is called as
- a. Numeric array
 - b. **Associative array**
 - c. Multidimensional array
 - d. Indexed Array
14. PHP's numerically indexed array begin with position _____
- a. 1
 - b. 2
 - c. **0**
 - d. -1
15. Which of the following function is used to get the value of the previous element in an array?
- a. last()
 - b. before()
 - c. **prev()**
 - d. previous()
16. Multidimensional arrays are simple arrays that have
- a. One dimensional
 - b. **Many arrays stored in them**
 - c. No indexes
 - d. 1 element

17. For finding nonempty elements in array we use
- is_array () function
 - sizeof () function
 - array_count () function
 - count () function**
18. When we simply want iteration through looping an array values we can use
- current ()
 - foreach ()**
 - next ()
 - prev()
19. Count () function is identical to
- is_array () function
 - in_array () function
 - sizeof ()**
 - isset () function
20. How does the identity operator === compare two values?
- It converts them to a common compatible data type and then compares the resulting values**
 - It returns True only if they are both of the same type and value
 - If the two values are strings, it performs a lexical comparison
 - It bases its comparison on the C strcmp function exclusively
21. Which of the following is not a global variable?
- \$_POST
 - \$_GLOBALS
 - \$_REQUEST
 - \$NUM**
22. ____loop will execute at least once.
- While
 - Do while**
 - For
 - Foreach
23. Which among the following has highest precedence over others?
- ==
 - ++
 - &&
 - +
24. Which of the following is unary operator?
- -
 - *
 - &&
25. ____are used in PHP to performs what is known as variable expansion
- Double quotes**
 - Single quotes
 - Comma

d. Semicolon

Long Answer Questions (Application)

1. Explain the following with example

a. Assignment operator

b. Logical operator

Ans.: Assignment operators

You can use the Assignment Operator to assign a value to a variable. Often a variable is assigned a value of another variable. In this case assignment operators are used. The equal character (=) is used here. Look at the following expression:

Example:

```
$first_var = 5;
```

```
$second_var = $first_var;
```

Here the values of both '\$first_var' and '\$second_var' variables are assigned the same value i.e.

Operator	Example	Equivalent to
=	<code>\$j = 15</code>	<code>\$j = 15</code>
+=	<code>\$j += 5</code>	<code>\$j = \$j + 5</code>
-=	<code>\$j -= 3</code>	<code>\$j = \$j - 3</code>
*=	<code>\$j *= 8</code>	<code>\$j = \$j * 8</code>
/=	<code>\$j /= 16</code>	<code>\$j = \$j / 16</code>
.=	<code>\$j .= \$k</code>	<code>\$j = \$j . \$k</code>
%=	<code>\$j %= 4</code>	<code>\$j = \$j % 4</code>

Logical operator

Logical Operators	Functions
And	Checks if two or more statements are true
&&	Same as And
Or	Checks if at least one of two statements is true
	Same as Or
Xor	Checks if only one of two statements is true
!	Checks if a statement is not true

2. What is Escaping Characters? Write brief notes on Escaping Characters.

Ans.: Escape Sequences:

You can achieve the same effect in double-quoted strings by using the escape character, which, in PHP, is a backslash \. Escape sequences, the combination of the escape character \ and a letter, are used to signify that the character after the escape character should be treated specially. For example, if you wanted to have the string "And then he said, "That is amazing!", which was true", you would need escape characters because you have double quotes inside double quotes. Here is a list of the escape sequences in PHP:

Escape Sequences	Functions
\"	Used to print the next character as a double quote, not as a string closer
\'	Used to print the next character as a single quote, not a string closer
\n	Used to print a new line character (remember our print statements?)
\t	Used to print a tab character
\r	Used to print a carriage return (not used very often)
\\$	Used to print the next character as a dollar, not as part of a variable
\\	Used to print the next character as a backslash, not an escape character

3. With appropriate example differentiate post increment and pre increment.

Ans.: Pre-increment operator

++\$var (Pre-increment : Increments \$var by 1, then returns \$var)

Post-increment operator

\$var++ (Post-increment : Returns \$var, then increments \$var by one)

Example 1: Pre-increment operator

```
<?php
$i=2;
//Increments $i by 1, then returns $i
echo ++$i."<br>";
echo $i;
?>
```

Output

3
3

Example 2: Post-increment operator

```
<?php
$i=2;
//Returns $i, then increments $i by one echo $i++."<br>";
echo $i;
?>
```

Output

2
3

4. **Explain date () function with its syntax. List and explain any eight format which can be used with it.**

Ans.: Date function()

In PHP the date() function is used to format a timestamp or a date. It arranges a timestamp into a readable date and time. Using the date/ time functions, you can format date and time on the server.

However, these functions entirely depend on the server settings.

Here you can use the syntax,

date (format, timestamp).

Look at the table below:

Format: This parameter is essential. It assigns the timestamp format.

Timestamp: This Parameter is optional. This takes the Date or/and time that you want to format.

If no value is provided then the current time is used for formatting.

In the date function, the first parameter specifies about formatting date and time. Several letters are used to represent date and time formats. Some commonly used letters are given below:

- d - Represents day of a month (01-31)
- D - Represents day in three letter text format
- m - Represents month, as a number (01-12)
- M - Represents month in three letter text format
- Y - Represents year in four digits
- y - Represents year in two digits
- a – “am” or “pm”
- A – “AM” OR “PM”
- F – Full name of the month (January- December)
- g – Hours in 12- hour format without leading zero (1-12)
- G – Hours in 24- hour format without leading zero (0-23)
- h – Hours in 12- hour format with leading zero (01-12)

- H – Hours in 24- hour format with leading zero (00-23)
- i– minutes with leading zero(00-59)
- j – day of month without leading zero
- l – the full name of the day (Monday- Sunday)
- n – the month as a number, without leading zero(1-12)
- s – the seconds with leading zero(00-59)
- t – the number of days in a given month (28-31)

```
<?php
// Assuming today is March 10th, 2001, 5:16:18 pm, and that we are in the
// Mountain Standard Time (MST) Time Zone

$today = date("F j, Y, g:i a");           // March 10, 2001, 5:16 pm
$today = date("m.d.y");                   // 03.10.01
$today = date("j, n, Y");                 // 10, 3, 2001
$today = date("Ymd");                     // 20010310
$today = date('h-i-s, j-m-y, it is w Day'); // 05-16-18, 10-03-01, 1631 1618 6 Satpm01
$today = date('\i\t \i\s \t\h\e jS \d\ay.'); // It is the 10th day.
$today = date("D M j G:i:s T Y");         // Sat Mar 10 17:16:18 MST 2001
$today = date('H:m:s \m \i\s\ \m\o\n\t\h'); // 17:03:18 m is month
$today = date("H:i:s");                   // 17:16:18
$today = date("Y-m-d H:i:s");             // 2001-03-
10 17:16:18 (the MySQL DATETIME format)
?>
```

5. With example, explain multi-dimensional array.

Ans.: **Multidimensional Arrays**

A multi-dimensional array each element in the main array can also be an array. And each element

in the sub-array can be an array, and so on. Values in the multi-dimensional array are accessed using multiple index.

Example

In this example we create a two dimensional array to store marks of three students in three subjects –

This example is an associative array, you can create numeric array in the same fashion

```
<html>
<body>
<?php
$marks = array(
    "mohammad" => array (
        "physics" => 35,
        "maths" => 30,
        "chemistry" => 39
    ),
    "qadir" => array (
        "physics" => 30,
        "maths" => 32,
        "chemistry" => 29
    ),
    "zara" => array (
        "physics" => 31,
        "maths" => 22,
        "chemistry" => 39
    )
);
/* Accessing multi-dimensiona l array values */
echo "Marks for mohammad in physics : " ;
echo $marks['mohammad']['physics'] . "<br />";
```

```

echo "Marks for qadir in maths : ";
echo $marks['qadir']['maths'] . "<br />";
echo "Marks for zara in chemistry : ";
echo $marks['zara']['chemistry'] . "<br />";
?>
</body>
</html>

```

Marks for mohammad in physics : 35
 Marks for qadir in maths : 32
 Marks for zara in chemistry : 39

6. Explain the following with example, and give respective output.

a. substr_count()

b. strcasecmp()

Ans.: **substr_count()**

The substr_count() function is used to count the number of times a substring appears in a string. This function was introduced in PHP4.

```

<html>
<body>
<?php print substr_count("Hello how are you and what are you doing now?", "are"); ?>
</body>
</html>

```

Output

2

strcasecmp()

The strcasecmp() function is used to compare two case sensitive strings. This function was introduced in PHP3. Look at the example below:

```

<html>
<body>
<?php
$text1="Good morning";
$text2="Good morning";
if (strcasecmp($text1, $text2) == 0) {
echo '$text1 is equal to $text2 in a case-insensitive string comparison'; }
?>
</body>
</html>

```

Output

\$text1 is equal to \$text2 in a case-insensitive string comparison

Long Answer Questions (Understanding)
--

7. What are the different data types available in PHP

Ans.: PHP has a total of eight data types which we use to construct our variables:

- Integers: are whole numbers, without a decimal point, like 4195.
- Doubles: are floating-point numbers, like 3.14159 or 49.1.
- Booleans: have only two possible values either true or false.
- NULL: is a special type that only has one value: NULL.
- Strings: are sequences of characters, like 'PHP supports string operations.'
- Arrays: are named and indexed collections of other values.

- Objects: are instances of programmer-defined classes, which can package up both other kinds of values and functions that are specific to the class.
- Resources: are special variables that hold references to resources external to PHP (such as database connections).

8. With syntax and example explain, else if and switch statements in PHP.

Ans.: **elseif/else if:**

It is a combination of 'if' and 'else' Control Structure. If the 'if' Control Structure returns a 'false' value, then a different statement is executed by using the 'else' Control Structure'.

Example 1:

```
<?php
$t = date("H");
if ($t < "10") {
    echo "Have a good morning!";
} elseif ($t < "20") {
    echo "Have a good day!";
} else {
    echo "Have a good night!";
}
?>
```

switch:

This Control Structure is similar to a series of 'if' statements.

If you want to select one of many blocks of code to be executed, use the Switch statement.

The switch statement is used to avoid long blocks of if..elseif..else code.

Syntax

```
switch (expression)
{
    case label1: code to be executed if expression = label1;
    break;
    case label2: code to be executed if expression = label2;
    break;
    default: code to be executed if expression is different from both label1 and label2;
}
```

Example

```
<?php
$favcolor= "red";
switch ($favcolor) {
    case "red":
        echo "Your favoritecolor is red!";
        break;
    case "blue":
        echo "Your favoritecolor is blue!";
        break;
    case "green":
        echo "Your favoritecolor is green!";
        break;
    default:
        echo "Your favoritecolor is neither red, blue, nor green!";
}
?>
```

9. Why do we need break and continue statements, explain with example?

Ans.: **The break statement**

The PHP break keyword is used to terminate the execution of a loop prematurely.

The break statement is situated inside the statement block. It gives you full control and whenever you want to exit from the loop you can come out. After coming out of a loop immediate statement

to the loop will be executed.

Example

In the following example condition test becomes true when the counter value reaches 3 and loop terminates.

```
<html>
<body>
<?php
$i = 0;
while( $i< 10)
{
    $i++;
    if( $i == 3 )break;
}
echo ("Loop stopped at i = $i" );
?>
</body>
</html>
```

This will produce following result:

Loop stopped at i = 3

The continue statement

The PHP continue keyword is used to halt the current iteration of a loop but it does not terminate the loop.

Just like the break statement the continue statement is situated inside the statement block containing the code that the loop executes, preceded by a conditional test. For the pass encountering continue statement, rest of the loop code is skipped and next pass starts.

Example

In the following example loop prints the value of array but for which condition becomes true it just skip the code and next value is printed.

```
<?php
$x=1;
echo 'List of odd numbers between 1 to 10 <br />';
while ($x<=10)
{
    if (($x%2)==0)
    {
        $x++;
        continue;
    }
    else
    {
        echo $x.'<br />';
        $x++;
    }
}
```

List of odd numbers between 1 to 10

1
3
5
7
9

10. What is an array give its syntax? Explain Reset () and Sizeof () function with example.

Ans. **Arrays**

In PHP, arrays are ordered data maps and are used to store, manage and operate a set of variables. To put it simply, an array is a data structure that holds multiple data within a single identifier. There are two parts in an Array - Values and Keys. While Values contain information to be stored, Keys are used to identify these values. It is allocated to a single variable. It holds significant information, popularly termed as Array Elements. This information can be used for a number of times in a program. Either non-negative Integers or Strings are used as Keys. The arrays that use non-negative Integers as Keys are termed as Scalar Arrays. These are Associative Arrays that use Strings as keys. An Array may contain different Array(s) popularly known as Multidimensional Arrays.

The syntax of an Array is as follows:

```
$array[key] = value;
```

sizeof()

The sizeof() function returns the number of elements in an array.

The sizeof() function is an alias of the count() function.

```
<?php  
$cars=array("Volvo","BMW","Toyota");  
echo sizeof($cars);  
?>
```

Output:

3

Reset()

The reset() function moves the internal pointer to the first element of the array.

```
<?php  
$people = array("Peter", "Joe", "Glenn", "Cleveland");  
echo next($people) . "<br>";  
echo reset($people);  
?>
```

Output

Joe

Peter

11. Define numeric and associative array with example.

Ans. **Numeric Array**

These arrays can store numbers, strings and any object but their index will be represented by numbers. By default array index starts from zero.

Example

Following is the example showing how to create and access numeric arrays.

Here we have used **array()** function to create array. This function is explained in function

Reference

```
<html>
<body>
<?php
$numbers[0] = "one";
$numbers[1] = "two";
$numbers[2] = "three";
$numbers[3] = "four";
$numbers[4] = "five";
foreach( $numbers as $value ) {
echo "Value is $value <br />";
}
?>
```

```
</body>
```

```
</html>
```

Output

Value is one

Value is two

Value is three

Value is four

Value is five

Associative Arrays

The associative arrays are very similar to numeric arrays in term of functionality but they are different in terms of their index. Associative array will have their index as string so that you can establish a strong association between key and values.

Example

```
<!DOCTYPE html>
<html>
<body>
<?php
$page = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");
echo "Peter is " . $age['Peter'] . " years old.";
?>
</body>
</html>
```

OUTPUT

Peter is 35 years old.

12. With syntax and example explain the following:

- a. String length
- b. Count the Number of Words in a String
- c. Reverse a String
- d. Search for a Specific Text Within a String

Ans.: The PHP strlen() function

The strlen() function returns the length of a string, in characters. The example below returns the length of the string "Hello world!": strlen() is often used in loops or other functions, when it is important to know when a string ends. (i.e. in a loop, we might want to stop the loop after the last character in a string)

```
<?php
echostrlen("Hello world!"); // outputs 12
```

```
?>
```

Output:

12

Count The Number of Words in a String

The PHP `str_word_count()` function counts the number of words in a string:

Example

```
<?php
echostr_word_count("Hello world!"); // outputs 2
?>
```

Outputs:

2

Reverse a String

The PHP `strrev()` function reverses a string:

Example

```
<?php
echostrrev("Hello world!"); // outputs !dlrowolleH
?>
```

Outputs:

!dlrowolleH

Search For a Specific Text Within a String

The PHP `strpos()` function searches for a specific text within a string.

If a match is found, the function returns the character position of the first match. If no match is found, it will return `FALSE`.

The example below searches for the text "world" in the string "Hello world!":

Example

```
<?php
echostrpos("Hello world!", "world"); // outputs 6
?>
```

Outputs:

6

UNIT IV

1. Which predefined variables are used to retrieve information from forms?
 - a) \$GET
 - b) \$_GET**
 - c) \$_SET
 - d) GET
2. When you use the \$_GET variable to collect data, the data is visible to _____
 - a) none
 - b) only you
 - c) everyone**
 - d) selected few
3. When you use the \$_POST variable to collect data, the data is visible to _____
 - a) none
 - b) only you**
 - c) everyone
 - d) selected few
4. Which variable is used to collect form data sent with both the GET and POST methods?
 - a) \$BOTH
 - b) \$_BOTH
 - c) \$REQUEST
 - d) \$_REQUEST**
5. Which one of the following should not be used while sending passwords or other sensitive information?
 - a) GET**
 - b) POST
 - c) REQUEST
 - d) NEXT
6. There is a superglobal variable that stores information passed into the script through HTTP cookies.
 - A) \$COOKIE
 - B) \$_COOKIE**
 - C) \$SET_COOKIE
 - D) \$ADD_COOKIE
7. What is the default type of 'type' attribute of <input> element?
 - a) Text**
 - b) Password
 - c) Numerals
 - d) Special Characters
8. In PHP, cookies are set by using the
 - a. setcookie () function**
 - b. set () function
 - c. cookie_set () function
 - d. cookie()
9. _____ variable is used to set session

- a. **\$_SESSION**
 - b. **\$_SESSION**
 - c. **SESSION**
 - d. **SESSION**
10. How do we remove all session variables
- a. **session_unset();**
 - b. **session_destroy();**
 - c. **session_delete();**
 - d. **session_stop();**
11. Which symbol is used for **Pass by Reference**
- a. **&**
 - b. **|**
 - c. **%**
 - d. **\$**
12. How do you get information from a form that is submitted using the "get" method?
- a. **\$_GET[];**
 - b. **Request.Form;**
 - c. **Request.QueryString;**
 - d. **\$_POST[];**
13. Which of the following is the right one to define the function
- a. **function {function body}**
 - b. **datatype functionname(parameters){function body}**
 - c. **function {function body}**
 - d. **function functionname (parameters){function body}**
14. Maximum length of get method
- a. **2048**
 - b. **2000**
 - c. **2100**
 - d. **3000**
15. In PHP default behavior for user defined functions is
- a. **Call-by-value**
 - b. **Call-by-reference**
 - c. **Call-by-type**
 - d. **Call-by-loc**
16. Which of the following methods would you use for sending an email?
- a. **mail(\$to,\$subject,\$body)**
 - b. **sendmail(\$to,\$subject,\$body)**
 - c. **mail(to,subject,body)**
 - d. **sendmail(to,subject,body)**
17. In your PHP application you need to open a file. You want the application to issue a warning and continue execution, in case the file is not found. The ideal function to be used is:
- a. **include()**
 - b. **require()**

- c. nowarn()
 - d. getFile(false)
18. Sessions doesn't allow you to
- A) store persistent user preference on a site
 - B) save user authentication information from page to page
 - C) create multipage forms
 - d) **Doesn't save user authentication information from page to page**
19. ____ will produce a fatal error
- a. **Require**
 - b. Include
 - c. Get
 - d. Post
20. Which of the following is a mandatory parameter of setcookie
- a. expiry_time
 - b. cookie_path
 - c. **cookie_name**
 - d. domain
21. ____ can be used to define the cookie access hierarchy
- a. expiry_time
 - b. cookie_path
 - c. cookie_name
 - d. **domain**
22. A definition of function starts with ____.
- a. **Function**
 - b. Datatype
 - c. Function name
 - d. Void
23. ____ things are stored in the \$_FILES array when a file is uploaded?
- a. **5**
 - b. 8
 - c. 7
 - d. 9
24. \$_FILES is an ____ array
- a. **Associative**
 - b. Numeric
 - c. Indexed
 - d. Boolean

Long Answer Questions (Application)

1. Explain the use of any 2 form elements in php with appropriate example.

Ans.

First, we'll introduce the first HTML element you'll need: form . It delimits the form's area in the page and holds the fields you want your web site users to fill in.

```
< form action="formprocess1.php" method="post" >
```

```
<!--form controls go here-->
</form>
```

Notice that the form element has an ending tag and two attributes. The first attribute (action) is the recipient page address (the form - processing script). The second attribute (method) is the way in which you will send the data to the recipient. You may recall that there are two separate ways of sending a form to its processing script: the POST and the GET methods.

The POST method takes the data from the form fields and sends it through an HTTP header. In this case, the data cannot be seen in the URL. The GET method gets the data from the form fields, encodes it, and appends it to the destination URL.

2. Explain function declaration in PHP. Give example.

Ans. In all programming and scripting languages, a function is a block of code that is used repetitively in a program. It saves time while developing a web page. In PHP, the concept of function is the same as in other languages. There are some in-built functions in PHP. Besides that, we can define functions as per our requirements. These are called ‘User Defined Functions’.

Look at the elements of a function:

function: all function declarations begin with the word ‘function’.

Name of the function: names to a function are usually assigned in accordance with its utility.

Opening and Closing parentheses (): the opening and closing parentheses are an integral part of a function and you can insert both the opening and closing parentheses together, just after the name of the function. As the dollar sign (\$) indicates the existence of a variable, these parentheses indicate the existence of a function.

Opening and Closing curly braces {}: the opening curly brace {} indicates the beginning of the function code and the closing curly brace marks the termination of a function.

Example:

```
<html>
<body>
<?php
function DisplayTitle()
{
echo "Learning Function";
}
DisplayTitle();
?>
</body>
</html>
```

In this example, PHP codes are embedded in HTML. Here, we have used a function ‘DisplayTitle()’. This function starts with the word function and indicates that the character inserted just after this word is a function. It displays the title of the tutorial. Any one who will go through this will understand the purpose of this function from its name.

3. Write a note on scope of variable.

Ans. The origin from where a function can be accessed is called the function scope. A function, once declared, can be accessed from any section of a program. A variable scope will be local to a function, if defined within a function. Use the global key word while using a variable defined in the body part of the program

- 1) **Local variables** A variable declared within a PHP function is local and can only be accessed within that function. (the variable has local scope):

```
<?php
$a = 5; // global scope
function myTest()
{
```



```
echo $a; // local scope
}
myTest();
?>
```

The script above will not produce any output because the echo statement refers to the local scope variable \$a, which has not been assigned a value within this scope.

You can have local variables with the same name in different functions, because local variables are only recognized by the function in which they are declared. Local variables are deleted as soon as the function is completed

- 2) **Global variables** Global scope refers to any variable that is defined outside of any function. Global variables can be accessed from any part of the script that is not inside a function. To access a global variable from within a function, use the global keyword:

```
<?php
$a = 5; $b = 10;
function myTest()
{
    global $a, $b;
    $b = $a + $b;
}
myTest();
echo $b;
?>
```

The script above will output 15.

4. What are the elements required to build PHP form explain with example?

Ans. GET and POST

When a form is submitted using the GET method, its values are encoded directly in the query string portion of the URL

When a form is submitted using the POST method, its values will not be displayed the query string portion of the URL

An interactive web site requires user input, which is generally gathered through forms. As in the paper - based world, the user fills in a form and submits its content for processing. In a web application, the processing isn't performed by a sentient being; rather, it is performed by a PHP script. Thus, the script requires some sort of coded intelligence.

When you fill in a paper form, you generally use a means to deliver its content (the postal service is one example) to a known address (such as a mail - order bookstore). The same logic applies to online forms. The data from an HTML form is sent to a specific location and processed.

The form element is rather simple in HTML. It states where and how it will send the contents of the elements it contains once submitted. It is after that point that PHP comes into play. PHP uses a set of simple yet powerful expressions that, when combined, provide you with the means to do virtually anything you want. The PHP script receives the data from the form and uses it to perform an action such as updating the contents of a database, sending an e - mail, testing the data format, and so on.

Handling forms is a multipart process. First a form is created, into which a user can enter the required details. This data is then sent to the web server, where it is interpreted, often with some error checking. If the PHP code identifies one or more fields that require reentering, the form may be redisplayed with an error message. When the code is satisfied with the accuracy of the input, it takes some action that usually involves the database, such as entering details about a purchase

To build a form, you must have at least the following elements:

- An opening <form> and closing </form> tag

- A submission type specifying either a GET or POST method
- One or more input fields
- The destination URL to which the form data is to be submitted

```
<html>
  <head>
    <title>Form Test</title>
  </head>
  <body>
    <form method="post" action="formtest.php">
      What is your name?
      <input type="text" name="name" />
      <input type="submit" />
    </form>
  </body>
</html>
```

5. What is session? How the session is created explain with example.

Ans. When you work with an application, you open it, do some changes, and then you close it. This is much like a Session. The computer knows who you are. It knows when you start the application and when you end. But on the internet there is one problem: the web server does not know who you are or what you do, because the HTTP address doesn't maintain state.

Session variables solve this problem by storing user information to be used across multiple pages (e.g. username, favoritecolor, etc). By default, session variables last until the user closes the browser.

So; Session variables hold information about one single user, and are available to all pages in one application.

Start a PHP Session

A session is started with the session_start() function.

Session variables are set with the PHP global variable: \$_SESSION.

Now, let's create a new page called "demo_session1.php". In this page, we start a new PHP session and set some session variables:

Example

```
<?php
// Start the session
session_start();
?>
<!DOCTYPE html>
<html>
<body>
<?php
// Set session variables
$_SESSION["favcolor"] = "green";
$_SESSION["favanimal"] = "cat";
echo "Session variables are set.";
?>
</body>
</html>
```

6. Explain mail() function with its syntax. Give example.

Ans. Mail() function

The mail() function is used to mail information to a given recipient.

Syntax:

`booleanmail(string recipient, string subject, string message [, string addl_headers])`

The recipient is the person to whom the mail has to be sent, the subject is, of course, the subject of the email. The message is the textual body of the email, and the optional input parameter `addl_headers` is used to supply any additional header information (such as HTML formatting) that is sent along with the email.

The codespec for a `emailinfo.php` is

```
<?php
// Change -$recipient to be the recipient of the form information
$recipient= "admin@ivanbayross.com";
// Email subject
$subject= "User Comments from LPOST['name']";
// Extra email headers
$headers= "From: LPOST['email'] ";
// Create a message
$message= "Messagefrom LPOST['name'] \n\n Most PurchasedProduct:
$POST['products'] \n\n Comments: \n\n LPOST['comments']";
// Send the email or produce an error
mail($recipient, $subject, $message, $headers) or die("Could not send email!");
// Send the user an appropriate message
print "Thank you LPOST['name'] for taking a moment to send us your comments!";
?>
```

Output:

Thank you Sharanam Shah for taking a moment to send us your comments!

Long Answer Questions (Understanding)

7. Write a simple php form with at least 4 input fields to display values obtained from inputfields.

Ans. `<html>`

`<head>`

`<title> Form </title>`

`</head>`

`<body>`

`<form action="b.php" method="post">`

`<table>`

`<tr>`

`<td> Name </td>`

`<td> <input type="text" name="name"/ > </td>`

`</tr> <tr>`

`<td colspan="2" style="text-align: center;" >`

`<input type="submit" name="submit" value="Submit" /> </td>`

`</tr><tr>`

`<input type="checkbox" name="check_list[]" value="C/C++"><label>C/C++</label>
`

`<input type="checkbox" name="check_list[]" value="Java"><label>Java</label>
`

`<input type="checkbox" name="check_list[]" value="PHP"><label>PHP</label>
`

`</tr>`

`<tr>`

```
Vegetables <select name="veg" size="1">
<option value="Peas">Peas</option>
<option value="Beans">Beans</option>
<option value="Carrots">Carrots</option>
<option value="Cabbage">Cabbage</option>
<option value="Broccoli">Broccoli</option>
</select>
</tr>
</table>
</form>
</body>
</html>
```

8. Why do we use the get method, explain the variable used to fetch the values passed by getmethod, give the drawback of the same?

Ans. PHP \$_GET Variable

In PHP, the **\$_GET** variable is used to collect values from HTML forms using method *get*. Information sent from an HTML form with the GET method is displayed in the browser's address bar, and it has a limit on the amount of information to send.

Example:

```
<html>

<body>

<form action="registration.php" method="get">

Name: <input type="text" name="name">

Email: <input type="text" name="email">

<input type="submit">

</form>

</body>

</html>
```

- The variable names and values will be visible in URL if HTML forms submitted by the GETmethod.
- The GET method is restricted to send up to *2048 characters* only.
- When you submit sensitive information like passwords then should not use this method.

- GET method can't be used, to send binary data like images and Word documents.
 - GET method data can be accessed using PHP QUERY_STRING environment variable.
- PHP \$_GET associative array is used to access all the sent information by GET method

9. Write a note on POST method and \$_POST variable.

Ans. In PHP, the `$_POST` variable is used to collect values from HTML forms using method `post`. Information sent from a form with the POST method is invisible and has no limits on the amount of information to send.

Note : However, there is an 8 MB max size for the POST method, by default (can be changed by setting the `post_max_size` in the `php.ini` file).

Example

```
<html>
<body>
<form action="registration.php" method="post">
Name: <input type="text" name="name">
Email: <input type="text" name="email">
<input type="submit">
</form>
</body>
</html>
```



The POST method does not have any restriction on data size to be sent.

The POST method can be used to send ASCII as well as binary data.

The data sent by POST method goes through HTTP header, so security depends on HTTP protocol. By using Secure HTTP, you can make sure that your information is secure.

PHP \$_POST associative array is used to access all the sent information by POST method. Variables are not visible in the URL so users can't bookmark your page.

10. Why do we need include and require statements? Differentiate include and require statements in PHP. With php script.

Ans. The include and require statements are identical, except upon failure:

- ☐ require will produce a fatal error (E_COMPILE_ERROR) and stop the script
- ☐ include will only produce a warning (E_WARNING) and the script will continue

So, if you want the execution to go on and show users the output, even if the include file is missing, use the include statement. Otherwise, in case of Framework, CMS, or a complex PHP application coding, always use the require statement to include a key file to the flow of execution.

This will help avoid compromising your application's security and integrity, just in-case one key file is accidentally missing.

Including files saves a lot of work. This means that you can create a standard header, footer, or

menu file for all your web pages. Then, when the header needs to be updated, you can only update the header include file.

Example 1

Assume we have a standard footer file called "footer.php", that looks like this:

```
<?php
echo "<p>Copyright &copy; 1999-" . date("Y") . " W3Schools.com</p>";
?>
```

To include the footer file in a page, use the include statement:

```
<!DOCTYPE html>
<html>
<body>
<h1>Welcome to my home page!</h1>
<p>Some text.</p>
<p>Some more text.</p>
<?php include 'footer.php';?>
</body>
</html>
```

The output will be

Welcome to my home page!

11. Give the syntax of cookie, explain the parameters in it.

Ans. A cookie is often used to identify a user. A cookie is a small file that the server embeds on the user's computer. Each time the same computer requests a page with a browser, it will send the cookie too. With PHP, you can both create and retrieve cookie values.

Setting Cookies With PHP

A cookie is created with the setcookie() function.

Syntax

```
<?php
setcookie(cookie_name, cookie_value, [expiry_time], [cookie_path], [domain], [secure],
[httponly]);
?>
```

HERE,

- ☐ Php "setcookie" is the PHP function used to create the cookie.
- ☐ "cookie_name" is the name of the cookie that the server will use when retrieving its value from the \$_COOKIE array variable. It's mandatory.
- ☐ "cookie_value" is the value of the cookie and its mandatory
- ☐ "[expiry_time]" is optional; it can be used to set the expiry time for the cookie such as 1 hour. The time is set using the PHP time() functions plus or minus a number of seconds greater than 0 i.e. time() + 3600 for 1 hour.
- ☐ "[cookie_path]" is optional; it can be used to set the cookie path on the server. The forward slash "/" means that the cookie will be made available on the entire domain. Sub directories limit the cookie access to the subdomain.
- ☐ "[domain]" is optional, it can be used to define the cookie access hierarchy i.e. www.cookie domain.com means entire domain while www.sub.cookie domain.com limits the cookie access to www.sub.cookie domain.com and its sub domains.

12. How to create a upload script in PHP, explain the different variables in it.

Ans.

There is one global PHP variable called \$_FILES. This variable is an associate double dimension array and keeps all the information related to uploaded file. Five things are stored in the \$_FILES array when a file is uploaded, So if the value assigned to the input's name attribute in uploading form was file, then PHP would create following variables –

- \$_FILES['file']['tmp_name'] – the uploaded file in the temporary directory on the webserver.
- \$_FILES['file']['name'] – the actual name of the uploaded file.
- \$_FILES['file']['size'] – the size in bytes of the uploaded file.
- \$_FILES['file']['type'] – the MIME type of the uploaded file.
- \$_FILES['file']['error'] – Error code resulting from file upload

Upload1.html

```
<html>
<body>
<form action="upload.php" enctype="multipart/form-data" method="post">
Your File Name <input type="file" name="file"/><br/>
<input type="submit" value="Upload" name="upload"/>
</form>
</body>
```

upload.php

```
<?php
if ($_POST['upload'])
{
if (
move_uploaded_file($_FILES["file"]["tmp_name"],
"/var/www/" . $_FILES["file"]["name"]))
{
echo "uploaded successfully!";
echo "<br>";
echo "Upload: " . $_FILES["file"]["name"] . "<br>";
echo "Type: " . $_FILES["file"]["type"] . "<br>";
echo "Size: " . ($_FILES["file"]["size"] / 1024) . " kB<br>";
echo "Stored in: " . $_FILES["file"]["tmp_name"];
} else {
print "Upload failed!";
}
}
?>
```


