PRACTICAL FILE OF PHP PROGRAMS

LIST OF PHP PROGRAMS

List of PHP Programs

Contents

Program to display hello world in php

```
<html>
<body>

<h1>My first PHP page</h1>

<?php
echo "Hello World!";
?>

</body>
</html>
```

program to declare a variable in php

```
<?php
$txt = "Hello world!";
$x = 5;
$y = 10.5;
?>
```

program to declare a local variable in php

```
<?php
    function local_var()
    {
        $num = 45;  //local variable
        echo "Local variable declared inside the function is: ". $num;
    }
    local_var();
?>
```

program to declare a local variable in php

```php
<?php
    function mytest()
    {
        $lang = "PHP";
        echo "Web development language: " .$lang;
    }
    mytest();
    //using $lang (local variable) outside the function will generate an error
    echo $lang;
?>
```

Output:

```
Web development language: PHP
Notice: Undefined variable: lang in
D:\xampp\htdocs\program\p3.php
```

program to declare a Global variable in php

```php
<?php
    $name = "GHGKHALSA COLLEGE";       //Global Variable
    function global_var()
    {
        global $name;
        echo "Variable inside the function: ". $name;
        echo "</br>";
    }
    global_var();
    echo "Variable outside the function: ". $name;
?>
```

Output:

```
Variable inside the function: GHGKHALSA COLLEGE
Variable outside the function: GHGKHALSA COLLEGE
```

program to access a variable in php

```
1. <?php
2.     $name = "Prof Singh";        //global variable
3.     function global_var()
4.     {
5.         echo "Variable inside the function: ". $name;
6.         echo "</br>";
7.     }
8.     global_var();
9. ?>
```

Output:-

Notice: Undefined variable: name in D:\xampp\htdocs\program\p3.php on line 5
Variable inside the function:

program to declare a static variable in php

```
1. <?php
2.     function static_var()
3.     {
4.         static $num1 = 3;          //static variable
5.         $num2 = 6;                        //Non-static variable
6.         //increment in non-static variable
7.         $num1++;
8.         //increment in static variable
9.         $num2++;
10.        echo "Static: " .$num1 ."</br>";
11.        echo "Non-static: " .$num2 ."</br>";
12.    }
13.
14.//first function call
15.    static_var();
16.
17.    //second function call
```

18. static_var();
19.?>

Output:

```
Static: 4
Non-static: 7
Static: 5
Non-static: 7
```

program to declare a constant in php

```
<?php
   define("MESSAGE", "Welcome to Website");
   echo MESSAGE;
?>
```

Create a constant with a case-insensitive name:

```
<?php
  define("MESSAGE", "Welcome to Website",true);
  echo message;
?>
```

program to declare a constant in php

```
<?php
define("MESSAGE", "Welcome to GHG");

function myTest()

{
 echo MESSAGE;
}
```

```
myTest();
?>
```

program to declare and use of contsant in php

```
1. <?php
2. const MESSAGE="Welcome to GHG KHALSA COLLEGE";
3. echo MESSAGE;
4. ?>
5. <?php
6.     define("MSG", "GHG KHALSA COLLEGE");
7.     echo MSG, "</br>";
8.     echo constant("MSG");
9.     //both are similar
10. ?>
```

program to store integer values in php

```
1. <?php
2.     $dec1 = 34;
3.     $oct1 = 0243;
4.     $hexa1 = 0x45;
5.     echo "Decimal number: " .$dec1. "</br>";
6.     echo "Octal number: " .$oct1. "</br>";
7.     echo "HexaDecimal number: " .$hexa1. "</br>";
8. ?>
```

Output:

```
Decimal number: 34
Octal number: 163
HexaDecimal number: 69
```

program to store float values in php

```php
<?php
    $n1 = 19.34;
    $n2 = 54.472;
    $sum = $n1 + $n2;
    echo "Addition of floating numbers: " .$sum;
?>
```

Output:

```
Addition of floating numbers: 73.812
```

program to store string values in php

```php
<?php
    $company = "GHG LMS";
    //both single and double quote statements will treat different
    echo "Hello $company";
    echo "</br>";
    echo 'Hello $company';
?>
```

Output:

```
Hello GHG LMS
Hello $company
```

program to declare an array in php

```php
<?php
    $bikes = array ("Royal Enfield", "Yamaha", "KTM");
    var_dump($bikes);   //the var_dump() function returns the datatype and values
    echo "</br>";
    echo "Array Element1: $bikes[0] </br>";
    echo "Array Element2: $bikes[1] </br>";
    echo "Array Element3: $bikes[2] </br>";
?>
```

Output:

```
array(3) { [0]=> string(13) "Royal Enfield" [1]=> string(6) "Yamaha" [2]=>
string(3) "KTM" }
Array Element1: Royal Enfield
Array Element2: Yamaha
Array Element3: KTM
```

program to use class and object in php

```
<?php
    class bike {
        function model() {
            $model_name = "Royal Enfield";
            echo "Bike Model: " .$model_name;
           }
    }
    $obj = new bike();
    $obj -> model();
?>
```

Output:

```
Bike Model: Royal Enfield
```

Program to use Arithmetic operator

```
<?php

// variable 1

$x = 29;

// variable 2

$y = 4;

// some arithmetic operations on

// these two variables

echo ($x + $y), "\n";

echo($x - $y), "\n";

echo($x * $y), "\n";

echo($x / $y), "\n";

echo($x % $y), "\n";
```

?>

Output:

```
33
25
116
7.25
1
```

Program to use Logical Operator

```
<?php

$x = 50;
$y = 30;

if ($x == 50 and $y == 30)
    echo "and Success \n";

if ($x == 50 or $y == 20)
    echo "or Success \n";

if ($x == 50 xor $y == 20)
    echo "xor Success \n";

if ($x == 50 && $y == 30)
    echo "&& Success \n";

if ($x == 50 || $y == 20)
    echo "|| Success \n";

if (!$z)
    echo "! Success \n";

?>
```

Output:

```
and Success
or Success
xor Success
&& Success
|| Success
! Success
```

Program to use of Comparison Operator

```
<?php

$a = 80;
$b = 50;
$c = "80";

// Here var_dump function has been used to
// display structured information. We will learn
// about this function in complete details in further
// articles.
var_dump($a == $c) + "\n";
var_dump($a != $b) + "\n";
var_dump($a <> $b) + "\n";
var_dump($a === $c) + "\n";
var_dump($a !== $c) + "\n";
var_dump($a < $b) + "\n";
var_dump($a > $b) + "\n";
var_dump($a <= $b) + "\n";
var_dump($a >= $b);

?>
```

Output:

```
bool(true)
bool(true)
bool(true)
bool(false)
bool(true)
bool(false)
bool(true)
bool(false)
bool(true)
```

Program to use of Conditional or Ternary Operators

```
<?php

$x = -12;

echo ($x > 0) ? 'The number is positive' : 'The number is negative';

?>
```

Output:

```
The number is negative
```

Program to use of Assignment operator

```
<?php

// simple assign operator
$y = 75;
echo $y, "\n";

// add then assign operator
$y = 100;
$y += 200;
echo $y, "\n";

// subtract then assign operator
$y = 70;
$y -= 10;
echo $y, "\n";

// multiply then assign operator
$y = 30;
$y *= 20;
echo $y, "\n";

// Divide then assign(quotient) operator
$y = 100;
$y /= 5;
echo $y, "\n";

// Divide then assign(remainder) operator
$y = 50;
$y %= 5;
echo $y;

?>
```

Output

```
75
300
60
600
20
0
```

Program to use of Array operator

```
<?php

$x = array("k" => "Car", "l" => "Bike");
$y = array("a" => "Train", "b" => "Plane");
```

```
var_dump($x + $y);
var_dump($x == $y) + "\n";
var_dump($x != $y) + "\n";
var_dump($x <> $y) + "\n";
var_dump($x === $y) + "\n";
var_dump($x !== $y) + "\n";

?>
```

Output:

```
array(4) {
  ["k"]=>
  string(3) "Car"
  ["l"]=>
  string(4) "Bike"
  ["a"]=>
  string(5) "Train"
  ["b"]=>
  string(5) "Plane"
}
bool(false)
bool(true)
bool(true)
bool(false)
bool(true)
```

Program to use increment/Decrement operator

```
<?php

$x = 2;
echo ++$x, " First increments then prints \n";
echo $x, "\n";

$x = 2;
echo $x++, " First prints then increments \n";
echo $x, "\n";

$x = 2;
echo --$x, " First decrements then prints \n";
echo $x, "\n";

$x = 2;
echo $x--, " First prints then decrements \n";
echo $x;

?>
```

Output:

```
3 First increments then prints
3
2 First prints then increments
3
1 First decrements then prints
1
2 First prints then decrements
1
```

Program to use of String Concatenation Operators

```
<?php

$x = "GHG";
$y = "LMS";
$z = "SYSTEM!!!";
echo $x . $y . $z, "\n";

$x .= $y . $z;
echo $x;

?>
```

Output:

```
GHGLMSSYSTEM!!!
GHGLMSSYSTEM!!!
```

Program to use of echo and print statement

- **Display Text**

The following example shows how to output text with the `echo` command (notice that the text can contain HTML markup):

Example

```
<?php
echo "<h2>web page development  using php!</h2>";
echo "Hello world!<br>";
echo "I'm about to learn PHP!<br>";
echo "This ", "string ", "was ", "made ", "with multiple parameters.";
?>
```

- **Display Variables**

The following example shows how to output text and variables with the `echo` statement:

Example

```
<?php
$txt1 = "Learn PHP";
$txt2 = "ghglms.org";
$x = 5;
$y = 4;

echo "<h2>" . $txt1 . "</h2>";
echo "Study PHP at " . $txt2 . "<br>";
echo $x + $y;
?>
```

- **Display Text**

The following example shows how to output text with the `print` command (notice that the text can contain HTML markup):

Example

```
<?php
print "<h2> web page development  using php!</h2>";
print "Hello world!<br>";
print "I'm about to learn PHP!";
?>
```

- **Display Variables**

The following example shows how to output text and variables with the `print` statement:

Example

```
<?php
$txt1 = "Learn PHP";
$txt2 = "ghglms.org";
$x = 5;
$y = 4;

print "<h2>" . $txt1 . "</h2>";
```

```
print "Study PHP at " . $txt2 . "<br>";
print $x + $y;
?>
```

Program of if statement

```
<?php

$marks = 60;

if ($marks > 40)

{

echo "You are passed";

}

?>
```

Program of if else statement

```
<?php
$marks = 60;


if ($marks > 40)
{
echo "You are passed";
}
else
{
echo “You are fail”;
}
?>
```

Program of elseif statement

```
<?php
$day='Wednesday';
if ($day=='Monday')
{
       echo "Week just started! long to go<BR>";
```

```
}
elseif ($day=='Tuesday')
{
      echo "Just overcame Monday Blues";
}
elseif ($day=='Wednesday')
{
       echo "Half week past";
}
elseif ($day=='Thursday')
{
       echo "Just keep patience for another day";
}
elseif ($day=='Friday')
{
       echo "Thank God its Friday";
}
elseif ($day=='Saturday')
{
       echo "Where's the party toninght>";
}
else
{
       echo "Can't we have two sundays each week";
}
?>
```

Program of switch case statement

```
<?php
$day='Friday';
switch($day)
{
case 'Monday':
                echo "Week just started! long to go<BR>";
                break;
case 'Tueday':
                echo "Just overcame Monday Blues";
                break;
case 'Wednesday':
                echo "Half week past";
                break;
case 'Thursday':
                echo "Just keep patience for another day";
                break;
case 'Friday':
                echo "Thank God its Friday";
                break;
case 'Saturday':
                echo "Where's the party tonight>";
                break;
default:
```

```
                echo "Can't we have two Sundays each week";
                break;
}
?>
```

Program of for loop

```
<?php

// code to illustrate for loop

for ($num = 1; $num <= 10; $num += 2)

{

echo "$num \n";

}

?>

Output:

1
3
5
7
9
```

Program of while loop

```
<?php

// PHP code to illustrate while loops

$num = 2;

while ($num < 12)

{

$num += 2;

echo $num, "\n";
```

```
}

?>
```

Output:-

```
4
6
8
10
12
```

Program of do while loop

```
<?php

// PHP code to illustrate do...while loops

$num = 2;

do

{

$num += 2;

echo $num, "\n";

}

while ($num < 12);

?>
```

Output:

```
4
6
8
10
12
```

Program of for each loop

```
<?php

$arr = array (10, 20, 30, 40, 50, 60);

foreach ($arr as $val)

{

echo "$val \n";

}

?>
```

Output:

```
10
20
30
40
50
60
```

Program of continue statement

```
<?php
         $array = array( 1, 2, 3, 4, 5);

         foreach( $array as $value )
{
            if( $value == 3 )continue;
            echo "Value is $value <br />";
         }
      ?>
```

This will produce the following result −

```
Value is 1
Value is 2
Value is 4
Value is 5
```

Program of index array

```
<html>
<body>
```

```
<?php
/* First method to create array. */
$numbers = array( 1, 2, 3, 4, 5);

foreach( $numbers as $value )
      {
echo "Value is $value <br />";
}

/* Second method to create array. */
$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>
```

This will produce the following result −

```
Value is 1
Value is 2
Value is 3
Value is 4
Value is 5
Value is one
Value is two
Value is three
Value is four
Value is five
```

Program of Associative array

```
<html>
<body>

<?php
/* First method to associate create array. */
$salaries = array("akash" => 2000, "amit" => 1000, "rahul" => 500);

echo "Salary of akash is ". $salaries['akash'] . "<br />";
echo "Salary of amit is ".  $salaries['amit']. "<br />";
echo "Salary of rahul is ".  $salaries['rahul']. "<br />";

/* Second method to create array. */
```

```
$salaries['akash'] = "high";
$salaries['amit'] = "medium";
$salaries['rahul'] = "low";

echo "Salary of akash is ". $salaries['akash'] . "<br />";
echo "Salary of amit is ".  $salaries['amit']. "<br />";
echo "Salary of rahul is ".  $salaries['rahul']. "<br />";
?>

</body>
</html>
```

This will produce the following result −

```
Salary of akash is 2000
Salary of amit is 1000
Salary of rahul is 500
Salary of akash is high
Salary of amit is medium
Salary of rahul is low
```

Program of Multidimensional Array

```
<?php

$emp = array

(

array(1,"akash",400000),

array(2,"amit",500000),

array(3,"rahul",300000)

);

for ($row = 0; $row < 3; $row++)

{

for ($col = 0; $col < 3; $col++)

{

echo $emp[$row][$col]."";
```

```
}

echo "<br/>";

}

?>
```

Output:

```
1 akash 400000
2 amit 500000
3 rahul 300000
```

Program of 2-D array

```
<html>
<body>

<?php
$marks = array("akash" => array (
"physics" => 35,
"maths" => 30,
"chemistry" => 39
),

"amit" => array (
"physics" => 30,
"maths" => 32,
"chemistry" => 29
),

"rahul" => array (
"physics" => 31,
"maths" => 22,
"chemistry" => 39
)
);

/* Accessing multi-dimensional array values */
echo "Marks for akash in physics : " ;
echo $marks['akash']['physics'] . "<br />";

echo "Marks for amit in maths : ";
echo $marks['amit']['maths'] . "<br />";

echo "Marks for rahul in chemistry : " ;
echo $marks['rahul']['chemistry'] . "<br />";
?>

</body>
```

```
</html>
```

This will produce the following result −

```
Marks for akash in physics : 35
Marks for amit in maths : 32
Marks for rahul in chemistry : 39
```

Program of array count() function

```
<html>
<body>
<?php
$language = array("PHP", "JAVA", "DBMS");
echo count($language);
?>
</body>
</html>
```

Output:
3

Program of print_r() function

```
<?php
$language = array("PHP", "JAVA", "DBMS");
print_r($language);
?>
```

Array ([0] =>"PHP" [1] =>"JAVA" [2] =>"DBMS")

Program of sort() function

```
1. <?php
2. $season=array("summer","winter","spring","autumn");
```

```
3. sort($season);
4. foreach( $season as $s )
5. {
6.   echo "$s<br />";
7. }
8. ?>
```

Output:

```
autumn
spring
summer
winter
```

program of array_search() function

```
1. <?php
2. $season=array("summer","winter","spring","autumn");
3. $key=array_search("spring",$season);
4. echo $key;
5. ?>
```

Output:

```
2
```

Program of array_values($arr) function

```
<?php

$cars = array(
"Suzuki" =>"Baleno",
"Skoda" =>"Fabia",
"Hyundai" =>"i20",
"Tata" =>"Tigor"
);

$cars_value = array_values($cars);

print_r($cars_value);

print_r($cars);
?>
```

Output:
Array ([0] => Baleno [1] => Fabia [2] => i20 [3] => Tigor)
Array ([Suzuki] => Baleno [Skoda] => Fabia [Hyundai] => i20 [Tata] => Tigor)

Program of `array_keys($arr)`

```
<?php

$cars = array(
"Suzuki" =>"Baleno",
"Skoda" =>"Fabia",
"Hyundai" =>"i20",
"Tata" =>"Tigor"
);

$cars_value = array_values($cars);
$cars_key=array_keys($cars);

print_r($cars_value);
echo "<br>";
print_r($cars_key);
echo "<br>";
print_r($cars);
?>
```

Output:-
Array ([0] => Baleno [1] => Fabia [2] => i20 [3] => Tigor)
Array ([0] => Suzuki [1] => Skoda [2] => Hyundai [3] => Tata)
Array ([Suzuki] => Baleno [Skoda] => Fabia [Hyundai] => i20 [Tata] => Tigor)

Program of array_slice($arr, $offset, $length)

```
<?php

$colors = array("red", "black", "blue", "green", "white", "yellow");
```

```
print_r(array_slice($colors, 2, 3));

?>
Array (
[0] => blue
[1] => green
[2] => white
)
```

Program of array_splice()

```
<!DOCTYPE html>
<html>
<body>

<?php
$a1=array("a"=>"red","b"=>"green","c"=>"blue","d"=>"yellow");
$a2=array("a"=>"purple","b"=>"orange");
print_r(array_splice($a1,0,2,$a2));
echo "<br>";
print_r($a1);
echo "<br>";
print_r($a2);
?>

</body>
</html>
```

Output:

```
Array ( [a] => red [b] => green )
Array ( [0] => purple [1] => orange [c] => blue [d] => yellow )
Array ( [a] => purple [b] => orange )
```

How to Add Elements to an Array with array_splice

```
<!DOCTYPE html>
```

```
<html>
<body>
<?php
$ar = array('PHP', 'JAVA', 'DBMS', 'VB');
// array_splice arguments: array to modify, offset (where to insert),
// number of elements to remove, array of elements to add
array_splice($ar, 2, 0, ['C', 'PYTHON'] );
print_r($ar);
/* print_r output (as seen in page source view):

?>

</body>
</html>
```

OUTPUT

```
Array ( [0] => PHP [1] => JAVA [2] => C [3] => PYTHON [4] => DBMS [5] => VB )
```

How to Add Elements to an Array with array_splice

```
<!DOCTYPE html>
<html>
<body>
<?php
$ar = array('PHP', 'JAVA', 'DBMS', 'VB');
// array_splice arguments: array to modify, offset (where to insert),
// number of elements to remove, array of elements to add
array_splice($ar, -1, 0, ['C', 'PYTHON'] );
print_r($ar);
/* print_r output (as seen in page source view):

?>

</body>
</html>
```

OUTPUT:

```
Array ( [0] => PHP [1] => JAVA [2] => DBMS [3] => C [4] => PYTHON [5] => VB )
```

How to Remove Elements from an Array with array_splice

```
<!DOCTYPE html>
<html>
<body>
<?php
$ar = array('PHP', 'JAVA', 'VB', 'C', 'HTML', 'PYTHON', 'ASP');
$r = array_splice($ar, 3, 2);
// view array returned by array_splice (array of removed elements)
print_r($r);
echo "<br>";
// view modified $ar
print_r($ar);
?>

</body>
</html>
```

OUTPUT:-

```
Array ( [0] => C [1] => HTML )
Array ( [0] => PHP [1] => JAVA [2] => VB [3] => PYTHON [4] => ASP )
```

If you leave out the third argument, the elements from the offset to the end of the array will be removed:

```
<!DOCTYPE html>
<html>
<body>
<?php
$ar = array('PHP', 'JAVA', 'VB', 'C', 'HTML', 'PYTHON', 'ASP');
$r = array_splice($ar, 3);
// view array returned by array_splice (array of removed elements)
print_r($r);
echo "<br>";
// view modified $ar
print_r($ar);
?>

</body>
</html>
```

OUTPUT-
Array ([0] => C [1] => HTML [2] => PYTHON [3] => ASP)
Array ([0] => PHP [1] => JAVA [2] => VB)

How to Replace Elements in an Array with array_splice

```
<!DOCTYPE html>
<html>
<body>
<?php
$ar = array('PHP', 'JAVA', 'VB', 'C', 'HTML', 'PYTHON', 'ASP');
// offset 3, remove 2, replace with single element
$r = array_splice($ar, 3,2,['C#']);
// view array returned by array_splice (array of removed elements)
print_r($r);
echo "<br>";
// view modified $ar
print_r($ar);
?>

</body>
</html>
```

OUTPUT-
Array ([0] => C [1] => HTML)
Array ([0] => PHP [1] => JAVA [2] => VB [3] => C# [4] => PYTHON [5] => ASP)

Program to declare function

```
1. <?php
2. function sayHello()
3. {
4. echo "Hello PHP Function";
5. }
6. sayHello(); //calling function
```

```
7. ?>
```

Output:

Hello PHP Function

Program of function arguments

```
1. <?php
2. function sayHello($name)
3. {
4. echo "Hello $name<br/>";
5. }
6. sayHello("BCA");
7. sayHello("BSC");
8. sayHello("MSC");
9. ?>
```

Output:

```
Hello BCA
Hello BSC
Hello MSC
```

Program of call by reference

```
1. <?php
2. function adder(&$str2)
3. {
4. $str2.='Call By Reference';
5. }
6. $str='Hello';
7. adder($str);
8. echo $str;
9. ?>
```

Output:

```
Hello Call By Reference
```

Program of function with default arguments

```
1. <?php
2. function sayHello($name="MSC")
3. {
4. echo "Hello $name<br/>";
5. }
6. sayHello("BCA");
7. sayHello();//passing no value
8. sayHello("BSC");
9. ?>
```

Output:

```
Hello BCA
Hello MSC
Hello BSC
```

Program of function with returning value

```
1. <?php
2. function cube($n)
3. {
4. return $n*$n*$n;
5. }
6. echo "Cube of 3 is:".cube(3);
7. ?>
```

Output:

```
Cube of 3 is: 27
```

Program of parameterized function

```
<!DOCTYPE html>

<html>
```

```
<head>
<title>Parameter Addition and Subtraction Example</title>
</head>
<body>
<?php
//Adding two numbers
function add($x, $y)
{
$sum = $x + $y;
echo "Sum of two numbers is = $sum <br><br>";
}
add(467, 943);
//Subtracting two numbers
function sub($x, $y) {
$diff = $x - $y;
echo "Difference between two numbers is = $diff";
}
sub(943, 467);
?>
</body>
</html>
```

Program of recursive function

```
<?php
function display($number)
 {
    if($number<=5)
{
     echo "$number <br/>";
     display($number+1);
    }
}

display(1);
?>
```

Output

```
1
2
3
4
5
```

Program to store string using single quote in php

```
<?php
$str1='Hello text
multiple line
text within single quoted string';
$str2='Using double "quote" directly inside single quoted string';
$str3='Using escape sequences \n in single quoted string';
echo "$str1 <br/> $str2 <br/> $str3";
?>
```

Output:

```
Hello text multiple line text within single quoted string
Using double "quote" directly inside single quoted string
Using escape sequences \n in single quoted string
```

Program to store string using single quote in php

```
1. <?php
2. $num1=10;
3. $str1='trying variable $num1';
4. $str2='trying backslash n and backslash t inside single quoted string \n \t';
5. $str3='Using single quote \'my quote\' and \\backslash';
6. echo "$str1 <br/> $str2 <br/> $str3";
7. ?>
```

Output:

```
trying variable $num1
trying backslash n and backslash t inside single quoted string \n \t
Using single quote 'my quote' and \backslash
```

Program to store string using double quote in php

```
1. <?php
2. $str="Hello text within double quote";
3. echo $str;
4. ?>
```

Output:

```
Hello text within double quote
```

Program to store string using double quote in php

```
1. <?php
2. $str1="Hello text
3. multiple line
4. text within double quoted string";
```

```
5. $str2="Using double \"quote\" with backslash inside double quoted string";
6. $str3="Using escape sequences \n in double quoted string";
7. echo "$str1 <br/> $str2 <br/> $str3";
8. ?>
```

Output:

```
Hello text multiple line text within double quoted string
Using double "quote" with backslash inside double quoted string
Using escape sequences in double quoted string
```

Program to store string using heredoc operator in php

```
1. <?php
2.     $str = <<<Demo
3. It is a valid example
4. Demo;    //Valid code as whitespace or tab is not valid before closing identifier
5. echo $str;
6. ?>
```

Output:

```
It is a valid example
```

PHP String Function Examples

1) PHP strtolower() function

The strtolower() function returns string in lowercase letter.

Syntax

```
1. strtolower ($string )
```

Example

```
1. <?php
2. $str="GHG LMS ";
```

```
3. $str=strtolower($str);
4. echo $str;
5. ?>
```

Output:

```
ghg lms
```

2) PHP strtoupper() function

The strtoupper() function returns string in uppercase letter.

Syntax

```
1. strtoupper($string)
```

Example

```
1. <?php
2. $str="ghg lms";
3. $str=strtoupper($str);
4. echo $str;
5. ?>
```

Output:

```
GHG LMS
```

3) PHP ucfirst() function

The ucfirst() function returns string converting first character into uppercase. It doesn't change the case of other characters.

Syntax

```
1. ucfirst ($str )
```

Example

```
1. <?php
2. $str="learning management system";
3. $str=ucfirst($str);
4. echo $str;
```

```
?>
```

Output:

```
Learning management system
```

4) PHP lcfirst() function

The lcfirst() function returns string converting first character into lowercase. It doesn't change the case of other characters.

Syntax

```
lcfirst ($str )
```

Example

```
<?php
$str="Learning management System";
$str=lcfirst($str);
echo $str;
?>
```

Output:

```
learning management System
```

5) PHP ucwords() function

The ucwords() function returns string converting first character of each word into uppercase.

Syntax

```
ucwords($str)
```

Example

```
<?php
$str="learning management system";
$str=ucwords($str);
echo $str;
?>
```

Output:

```
Learning Management System
```

6) PHP strrev() function

The strrev() function returns reversed string.

Syntax

1. strrev ($string)

Example

1. <?php
2. $str="learning management system";
3. $str=strrev($str);
4. echo $str;
5. ?>

Output:

7) PHP strlen() function

The strlen() function returns length of the string.

Syntax

1. strlen ($string)

Example

1. <?php
2. $str=”learning management system";
3. $str=strlen($str);
4. echo $str;
5. ?>

Output:

26

Creating table using MySQLi Object-oriented Procedure

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

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

// sql code to create table
$sql = "CREATE TABLE employees(
        id INT(2)  PRIMARY KEY,
        firstname VARCHAR(30) NOT NULL,
        lastname VARCHAR(30) NOT NULL,
        email VARCHAR(50)
        )";

if ($conn->query($sql) === TRUE)
{
    echo "Table employees created successfully";
} else {
    echo "Error creating table: " . $conn->error;
}

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

Output **:**

Table employees created successfully

Creating table using MySQLi Procedural procedure

```
<?php
$servername = "localhost:3306";
$username = "root";
$password = "";
$dbname = "bca_db";

// Checking connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn)
{
```

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

// sql code to create table
$sql = "CREATE TABLE employees (
        id INT(2)  PRIMARY KEY,
        firstname VARCHAR(30) NOT NULL,
        lastname VARCHAR(30) NOT NULL,
        email VARCHAR(50)
        )";

if (mysqli_query($conn, $sql))
{
    echo "Table employees created successfully";
} else {
    echo "Error creating table: " . mysqli_error($conn);
}
mysqli_close($conn);
?>
```

Output **:**

Table employees created successfully

Program to Insert Data Into MySQL Using MySQLi

```
<?php

$servername = "localhost";

$username = "root";

$password = "";

$dbname = "bca_db";


// Create connection

$conn = mysqli_connect($servername, $username, $password, $dbname);

// Check connection

if (!$conn)

 {
```

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

$sql = "INSERT INTO employees (id,firstname, lastname, email)
VALUES (101,'ajay', 'kumar', 'ghglms@example.com')";

if (mysqli_query($conn, $sql))
 {
  echo "New record created successfully";
}
Else
 {
  echo "Error: " . $sql . "<br>" . mysqli_error($conn);
}

mysqli_close($conn);
?>
```

Program to fetch data using MySQL Select command

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

// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn)
```

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

$sql = "SELECT id, firstname, lastname FROM employee";
$result = mysqli_query($conn, $sql);

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

mysqli_close($conn);
?>
```

Delete Data From a MySQL Table Using MySQLi

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

// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn)

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

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

```
if (mysqli_query($conn, $sql))

{
  echo "Record deleted successfully";
}

else

 {
  echo "Error deleting record: " . mysqli_error($conn);
}

mysqli_close($conn);
?>
```

Update Data In a MySQL Table Using MySQLi

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

// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn)

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

$sql = "UPDATE employee SET lastname='singh' WHERE id=1";

if (mysqli_query($conn, $sql))

 {
  echo "Record updated successfully";
}
```

```
 else

{
  echo "Error updating record: " . mysqli_error($conn);
}

mysqli_close($conn);
?>
```

Single-line text input controls

This control is used for items that require only one line of user input, such as search boxes or names. They are created using HTML <input> tag.

Example

Here is a basic example of a single-line text input used to take first name and last name −

```
<!DOCTYPE html>
<html>

   <head>
      <title>Text Input Control</title>
   </head>

   <body>
      <form >
         First name: <input type = "text" name = "first_name" />
         <br>
         Last name: <input type = "text" name = "last_name" />
      </form>
   </body>

</html>
```

Password input controls

This is also a single-line text input but it masks the character as soon as a user enters it. They are also created using HTML <input>tag but type attribute is set to **password**.

Example

Here is a basic example of a single-line password input used to take user password −

```
<!DOCTYPE html>
```

```
<html>

   <head>
      <title>Password Input Control</title>
   </head>

   <body>
      <form >
         User ID : <input type = "text" name = "user_id" />
         <br>
         Password: <input type = "password" name = "password" />
      </form>
   </body>

</html>
```

Multiple-Line Text Input Controls

This is used when the user is required to give details that may be longer than a single sentence. Multi-line input controls are created using HTML <textarea> tag.

Example

Here is a basic example of a multi-line text input used to take item description −

```
<!DOCTYPE html>
<html>

   <head>
      <title>Multiple-Line Input Control</title>
   </head>

   <body>
      <form>
         Description : <br />
         <textarea rows = "5" cols = "50" name = "description">
            Enter description here...
         </textarea>
      </form>
   </body>
```

</html>

Checkbox Control

Checkboxes are used when more than one option is required to be selected. They are also created using HTML <input> tag but type attribute is set to **checkbox.**.

Example

Here is an example HTML code for a form with two checkboxes −

```
<!DOCTYPE html>
<html>

   <head>
      <title>Checkbox Control</title>
   </head>

   <body>
      <form>
         <input type = "checkbox" name = "maths" value = "on"> Maths
         <input type = "checkbox" name = "physics" value = "on"> Physics
      </form>
   </body>

</html>
```

Radio Button Control

Radio buttons are used when out of many options, just one option is required to be selected. They are also created using HTML <input> tag but type attribute is set to **radio**.

Example

Here is example HTML code for a form with two radio buttons −

```
<!DOCTYPE html>
<html>

   <head>
      <title>Radio Box Control</title>
   </head>

   <body>
      <form>
         <input type = "radio" name = "subject" value = "maths"> Maths
         <input type = "radio" name = "subject" value = "physics"> Physics
      </form>
   </body>

</html>
```

Select Box Control

A select box, also called drop down box which provides option to list down various options in the form of drop down list, from where a user can select one or more options.

Example

Here is example HTML code for a form with one drop down box

```
<!DOCTYPE html>
<html>

   <head>
      <title>Select Box Control</title>
   </head>

   <body>
      <form>
         <select name = "dropdown">
            <option value = "Maths" selected>Maths</option>
            <option value = "Physics">Physics</option>
         </select>
      </form>
   </body>

</html>
```

File Upload Box

If you want to allow a user to upload a file to your web site, you will need to use a file upload box, also known as a file select box. This is also created using the <input> element but type attribute is set to **file**.

Example

Here is example HTML code for a form with one file upload box −

```
<!DOCTYPE html>
<html>

   <head>
      <title>File Upload Box</title>
   </head>

   <body>
      <form>
         <input type = "file" name = "fileupload" accept = "image/*" />
      </form>
   </body>
```

```
</html>
```

Button Controls

There are various ways in HTML to create clickable buttons. You can also create a clickable button using <input>tag by setting its type attribute to **button**.

Example

Here is example HTML code for a form with three types of buttons −

```
<!DOCTYPE html>
<html>

   <head>
      <title>File Upload Box</title>
   </head>

   <body>
      <form>
         <input type = "submit" name = "submit" value = "Submit" />
         <input type = "reset" name = "reset"  value = "Reset" />
         <input type = "button" name = "ok" value = "OK" />
         <input type = "image" name = "imagebutton" src =
"/html/images/logo.png" />
      </form>
   </body>

</html>
```

Program to Start a PHP Session

```
<?php
// Start the session
session_start();
?>
<!DOCTYPE html>
<html>
<body>

<?php
// Set session variables
$_SESSION["title"] = "ghglms";
$_SESSION["author"] = "GHG KHALSA COLLEGE";
echo "Session variables are set.";
?>
```

```
</body>
</html>
```

Program to display session variable

```
<?php
session_start();
?>
<!DOCTYPE html>
<html>
<body>

<?php
// Echo session variables that were set on previous page
echo "Title is is " . $_SESSION["title"] . ".<br>";
echo "Author  is " . $_SESSION["author"] . ".";
?>

</body>
</html>
```

Program to Destroy a PHP Session

```
<?php

// Starting session

session_start();

 // Removing session data

if(isset($_SESSION["title"]))

{

    unset($_SESSION["title"]);

}

?>
```

Program of encoding of session variable

```
<?php
session_start();
?>
<html>
<head>
	<title>
		session
	</title>
</head>
<body>
	<?php
		$_SESSION ["title"]="php";
		$_SESSION ["college"]="ghg khalsa college";
		$encod=session_encode();
		echo "encoded data is<br>";
		echo $encod;
	?>
	</body>
</html>
```

Program of decoding of session variable

```
<?php
session_start();
```

```
?>
<html>
<head>
	<title>
		session
	</title>
</head>
<body>
	<?php
		$_SESSION ["title"]="php";
		$_SESSION ["college"]="ghg khalsa college";
		$encod=session_encode();
		echo "encoded data is<br>";
		echo $encod;
		echo "<br>Decoded data is<br>";
		session_decode($encod);
		echo "<pre>";
		print_r($_SESSION);
		echo "</pre>";
	?>
	</body>
</html>
```

Program to create a cookies

```
<?php
$cookie_name = "user";
$cookie_value = "ghglms";
```

```
setcookie($cookie_name, $cookie_value, time() + (86400 * 30), "/"); // 86400 = 1 day
?>
<html>
<body>

<?php
Echo “cookie is set”;
?>

</body>
</html>
```

Program to Retrieve a Cookie

```
<html>

   <head>
      <title>Accessing Cookies with PHP</title>
   </head>

   <body>

      <?php
         echo $_COOKIE["user"]. "<br />";

         /* is equivalent to */
         echo $HTTP_COOKIE_VARS["user"]. "<br />";

      ?>

   </body>
</html>
```

Program to Setting Time in a Cookie

```
<?php
$value = "Hello world!";

// cookie will expire when the browser close
setcookie("myCookie", $value);

// cookie will expire in 1 hour
setcookie("myCookie", $value, time() + 3600);

// cookie will expire in 1 hour, and will only be available
// within the php directory + all sub-directories of php
setcookie("myCookie", $value, time() + 3600, "/php/");
```

```
// cookie will expire in 30days

setcookie("myCookie", $value, time() + 30*24*60*60);


?>
<html>
<body>

<?php

Echo “the cookie is set for time variatioin”;

?>

</body>
</html>
```

Program to Deleting Cookie with PHP

```
<?php
// set the expiration date to one hour ago
setcookie("user", "", time() - 3600);
?>
<html>
<body>

<?php
echo "Cookie 'user' is deleted.";
?>

</body>
</html>
```

Program of Reading a file

```
<html>

  <head>

    <title>Reading a file using PHP</title>

  </head>
```

```
  <body>
        <?php
      $filename = "tmp.txt";
      $file = fopen( $filename, "r" );
       if( $file == false ) {
        echo ( "Error in opening file" );
        exit();
      }
      $filesize = filesize( $filename );
      $filetext = fread( $file, $filesize );
      fclose( $file );
      echo ( "File size : $filesize bytes" );
      echo ( "<pre>$filetext</pre>" );
    ?>
  </body>
</html>
```

Program of writing a file

```
<?php
  $filename = "new.txt";
  $file = fopen( $filename, "w+" );
    if( $file == false )
  {
    echo ( "Error in opening new file" );
    exit();
```

```
   }
   fwrite( $file, "This is  a simple test\n" );
   fclose( $file );
?>
<html>
      <head>
      <title>Writing a file using PHP</title>
   </head>
      <body>

      <?php
         $filename = "new.txt";
         $file = fopen( $filename, "r" );
           if( $file == false ) {
            echo ( "Error in opening file" );
            exit();
         }
         $filesize = filesize( $filename );
         $filetext = fread( $file, $filesize );
          fclose( $file );
         echo ( "File size : $filesize bytes" );
         echo ( "<pre>$filetext</pre>" );
         echo("file name: $filename");
      ?>
      </body>
```

</html>