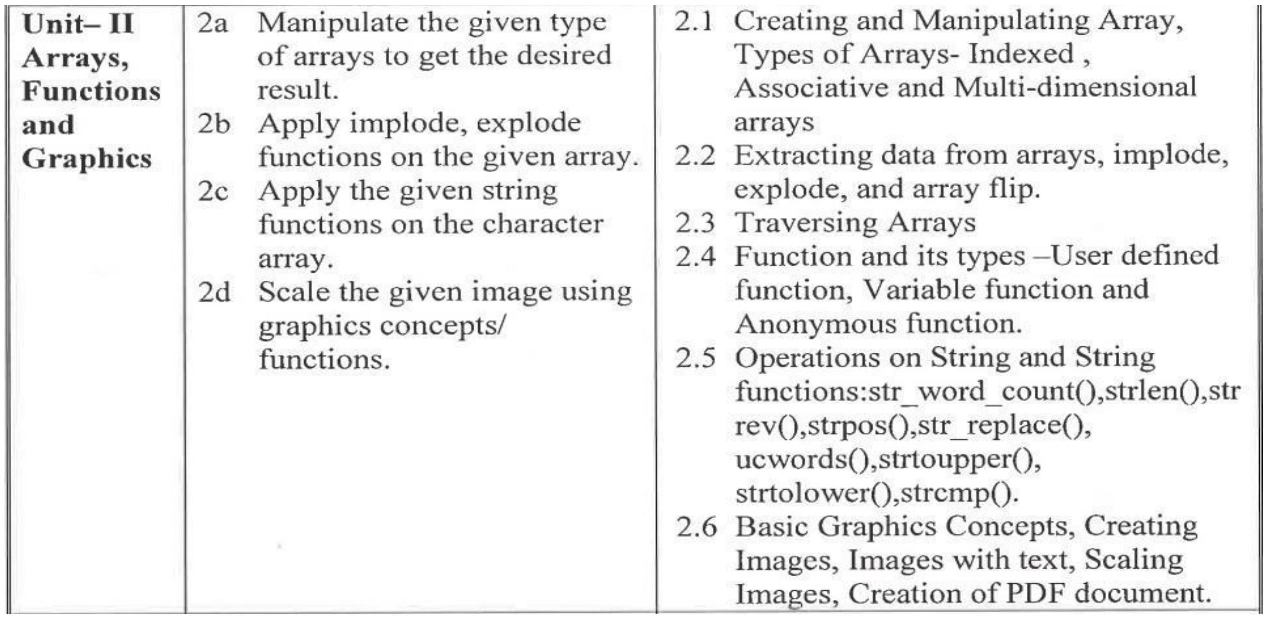
Chapter 02 16 marks



Unit– II Arrays, Functions and Graphics	2a Manipulate the given type of arrays to get the desired result. 2b Apply implode, explode functions on the given array. 2c Apply the given string functions on the character array. 2d Scale the given image using graphics concepts/ functions.	2.1 Creating and Manipulating Array, Types of Arrays- Indexed , Associative and Multi-dimensional arrays 2.2 Extracting data from arrays, implode, explode, and array flip. 2.3 Traversing Arrays 2.4 Function and its types –User defined function, Variable function and Anonymous function. 2.5 Operations on String and String functions:str_word_count(),strlen(),strrev(),strpos(),str_replace(), ucwords(),strtoupper(), strtolower(),strcmp(). 2.6 Basic Graphics Concepts, Creating Images, Images with text, Scaling Images, Creation of PDF document.

2.1 Creating and manipulation of Array

An array is a data structure that stores one or more similar type of values in a single value. For example if you want to store 100 numbers then instead of defining 100 variables its easy to define an array of 100 length.

There are three different kind of arrays and each array value is accessed using an ID c which is called array index.

Numeric array − An array with a numeric index. Values are stored and accessed in linear fashion.

Associative array − An array with strings as index. This stores element values in association with key values rather than in a strict linear index order.

Multidimensional array − An array containing one or more arrays and values are accessed using multiple indices

NOTE − Built-in array functions is given in function reference PHP Array Functions

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.
1. Numeric Array or Index Array.

Here we have used the array() function to create an array. This function is explained in functionreference.

```
<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
```

2.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.

To store the salaries of employees in an array, a numerically indexed array would not be the best choice. Instead, we could use the employees names as the keys in our associative array, and thevalue would be their respective salary.

NOTE − Don't keep an associative array inside double quotes while printing otherwise it would not return any value.

Example

```
<html>
  <body>

    <?php
      /* First method to associate creating an array. */
      $salaries = array("mohammad" => 2000, "qadir" => 1000, "zara" => 500);

      echo "Salary of mohammad is ". $salaries['mohammad'] . "<br />";
      echo "Salary of qadir is ". $salaries['qadir']. "<br />";
      echo "Salary of zara is ".  $salaries['zara']. "<br />";

      /* Second method is to create an array. */
      $salaries['mohammad'] = "high";
      $salaries['qadir'] = "medium";
      $salaries['zara'] = "low";

      echo "Salary of mohammad is ". $salaries['mohammad'] . "<br />";
      echo "Salary of qadir is ". $salaries['qadir']. "<br />";
      echo "Salary of zara is ".  $salaries['zara']. "<br />";
    ?>

  </body>
</html>
```

This will produce the following result −

```
Salary of mohammad is 2000
Salary of qadir is 1000
Salary of zara is 500
Salary of mohammad is high
Salary of qadir is medium
Salary of zara is low
```

3.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 threesubjects –

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-dimensional 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>
```

This will produce the following result –

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

2.2 .

1. extract() Function:-

The extract() Function is an inbuilt function in PHP. The extract() function does array to variable conversion. That is it converts array keys into variable names and array values into variable value. In other words, we can say that the extract() function imports variables from an array to the symbol table.

Syntax:

int extract($input_array, $extract_rule, $prefix)

Parameters: The extract() function accepts three parameters, out of which one is compulsory and other two are optional. All three parameters are described below:

1. $input_array: This parameter is required. This specifies the array to use.
2. $extract_rule: This parameter is optional. The extract() function checks for invalid variable names and collisions with existing variable names. This parameter specifies how invalid and colliding names will be treated.
3. $prefix: This parameter is optional. This parameter specifies the prefix. The prefix is automatically separated from the array key by an underscore character.
4. Example-1:

```
<?php
 // input array
   $state = array("AS"=>"ASSAM", "OR"=>"ORRISA", "KR"=>"KERELA");
      extract($state);
 // after using extract() function
    echo"\$AS is $AS\n\$KR is $KR\n\$OR is $OR";
      ?>
```

Output:

```
$AS is ASSAM
$KR is KERELA
$OR is ORRISA
```

2.PHP implode() and explode()

The implode() function takes an array, joins it with the given string, and returns the joined string.
The explode() function takes a string, splits it by specified string, and returns an array.
implode()
implode() function is used to join arrays. It joins an array with a given glue and returns a string.
The implode function has two arguments. implode(glue, array)
glue - the string which connects each array element
array - the array to be joined
E.g

implode() with indexed arrays

```
<?php
$array = ['Breakfast', 'Lunch', 'Dinner'];
echo implode(', ', $array);
?>
```

Output

```
Breakfast, Lunch, Dinner
```

explode()

explode() function is used to split strings. It splits a string by a delimiter, and returns an array.
The explode function has two arguments. explode(delimiter, string)
delimiter - the string which is used to split. (The boundary)
string - the string to be splitted
If the string is Apple,Banana, and the delimeter is ,, you will get an array with two elements Apple and Banana. The important thing to notice is that delimeter is not included in the splitted elements. They work excatly as boundaries.
explode() Basic Examples

```
<?php
$str = 'Apple, Banana, Cherry';
print_r( explode(',', $str) );  // There's an annoying whitespace

// No more whitespaces in element
// now the boundary (delimiter) is ', '
// string chunks around the delimiter will be elements of array
print_r( explode(', ', $str) );
```

Output

```
Array ( [0] => Apple [1] => Banana [2] => Cherry ) Array ( [0] => Apple [1] => Banana [2] =>
Cherry )
```

array_flip()

The array_flip() function flips/exchanges all keys with their associated values in an array.

Syntax
array_flip(array)
Parameter Values

Parameter	Description
array	Required. Specifies an array of key/value pairs to be flipped

```
<?php
        $input = array("a" => 1, "b" => 1, "c" => 2);
        $flipped = array_flip($input);

        print_r($flipped);
        ?>
        The above example will output:

        Array
        (
 [1] => b
[2] => c
)
```

2.3 Traversing Array

The most common task with arrays is to do something with every element—for instance, sending mail to each element of an array of addresses, updating each file in an array of filenames, or adding up each element of an array of prices. There are several ways to traverse arrays in PHP, and the one you choose will depend on your data and the task you're performing.

The foreach Construct
The most common way to loop over elements of an array is to use the foreach construct:

```
$addresses = array("spam@cyberpromo.net", "abuse@example.com");

foreach ($addresses as $value) {
  echo "Processing {$value}\n";
}
```

```
Processing spam@cyberpromo.net
Processing abuse@example.com
```

PHP executes the body of the loop (the echo statement) once for each element of $addresses in turn, with $value set to the current element. Elements are processed by their internal order.

An alternative form of foreach gives you access to the current key:

```
$person = array('name' => "Bharma", 'age' => 35, 'wife' => "Sarswati");

foreach ($person as $key => $value) {
  echo "Fred's {$key} is {$value}\n";
}
```

```
Fred's name is Bharma
Fred's age is 35
Fred's wife is Sarswati
```

In this case, the key for each element is placed in $key and the corresponding value is placed in $value.

The foreach construct does not operate on the array itself, but rather on a copy of it. You can insert or delete elements in the body of a foreach loop, safe in the knowledge that the loop won't attempt to process the deleted or inserted elements.

2.4 Function and its types

PHP functions are similar to other programming languages. A function is a piece of code which takes one more input in the form of parameter and does some processing and returns a value.You already have seen many functions like fopen() and fread() etc. They are built-infunctions but PHP gives you option to create your own functions as well.

There are two parts which should be clear to you −
Creating a PHP Function
Calling a PHP Function
In fact you hardly need to create your own PHP function because there are already more than 1000 of built-in library functions created for different area and you just need to call them according to your requirement.Please refer to PHP Function Reference for a complete set of useful functions.

Creating PHP Function

Its very easy to create your own PHP function. Suppose you want to create a PHP function which will simply write a simple message on your browser when you will call it. Following example creates a function called writeMessage() and then calls it just after creating it.

Note that while creating a function its name should start with keyword function and all the PHP code should be put inside { and } braces as shown in the following example below −

```
<html>

  <head>
    <title>Writing PHP Function</title>
  </head>

  <body>

    <?php
      /* Defining a PHP Function */
      function writeMessage() {
        echo "You are really a nice person, Have a nice time!";
      }

      /* Calling a PHP Function */
      writeMessage();
    ?>

  </body>
</html>
```

This will display following result −

```
You are really a nice person, Have a nice time!
```

PHP Functions with Parameters

PHP gives you the option to pass your parameters inside a function. You can pass as many parameters as you like. These parameters work like variables inside your function. Following example takes two integer parameters and add them together and then print them.

```
<html>

  <head>
    <title>Writing PHP Function with Parameters</title>
  </head>

  <body>

    <?php
      function addFunction($num1, $num2) {
        $sum = $num1 + $num2;
        echo "Sum of the two numbers is : $sum";
      }

      addFunction(10, 20);
    ?>

  </body>
</html>
```

This will display following result −

```
Sum of the two numbers is : 30
```

Passing Arguments by Reference

It is possible to pass arguments to functions by reference. This means that a reference to the variable is manipulated by the function rather than a copy of the variable's value.

Any changes made to an argument in these cases will change the value of the original variable. You can pass an argument by reference by adding an ampersand to the variable name in either the function call or the function definition.

Following example depicts both the cases.

```
<html>

  <head>
    <title>Passing Argument by Reference</title>
  </head>

  <body>

    <?php
      function addFive($num) {
        $num += 5;
      }

      function addSix(&$num) {
        $num += 6;
      }
```

```
        $orignum = 10;
        addFive( $orignum );

        echo "Original Value is $orignum<br />";

        addSix( $orignum );
        echo "Original Value is $orignum<br />";
      ?>

   </body>
</html>
```

This will display following result −

```
Original Value is 10
Original Value is 16
```

PHP Functions returning value

A function can return a value using the return statement in conjunction with a value or object. return stops the execution of the function and sends the value back to the calling code.

You can return more than one value from a function using the return array(1,2,3,4).

Following example takes two integer parameters and add them together and then returns their sum to the calling program. Note that return keyword is used to return a value from a function.

```
<html>
      <head>
      <title>Writing PHP Function which returns value</title>
   </head>

   <body>

      <?php
        function addFunction($num1, $num2) {
           $sum = $num1 + $num2;
           return $sum;
        }
        $return_value = addFunction(10, 20);

        echo "Returned value from the function : $return_value";
      ?>
       </body>
</html>
```

This will display following result −

```
Returned value from the function : 30
```

Setting Default Values for Function Parameters

You can set a parameter to have a default value if the function's caller doesn't pass it.Following function prints NULL in case use does not pass any value to this function.

```
<html>

  <head>
    <title>Writing PHP Function which returns value</title>
  </head>

  <body>

    <?php
      function printMe($param = NULL) {
        print $param;
      }

      printMe("This is test");
      printMe();
    ?>

  </body>
</html>
```

This will produce following result –This is test

Dynamic Function Calls

It is possible to assign function names as strings to variables and then treat these variables exactly as you would the function name itself. Following example depicts this behaviour.

```
<html>

  <head>
    <title>Dynamic Function Calls</title>
  </head>

  <body>

    <?php
      function sayHello() {
        echo "Hello<br />";
      }

      $function_holder = "sayHello";
      $function_holder();
    ?>

  </body>
</html>
```

This will display following result –Hello

Variable Function

PHP support the concept of the variable function. This means that if a variable name has parentheses appended to it, PHP will look for a function with the same name as the whatever variable evaluate to, and will attempt to execute it. This can be used to implement callbacks, function tables, etc.

```
<!--?php
 function bar()
 {
   echo "This is the output of the bar function";
 }
 $foo='bar';
 $foo();  // This will call the bar function
?-->
```

Anonymous Function

Anonymous functions, also known as closures, allow creation of functions which have no specified name.

Closures can also be used as the values of variables; PHP automatically converts such expressions into instances of the Closure internal class. Assigning a closure to a variable uses the same syntax as any other assignment, including the trailing semicolon:

```
<!--?php
  $unnamedFunction = function($name){
    printf("Hello %s\r\n", $name);
  };

  $unnamedFunction('World');
  $unnamedFunction('PHP');
?-->
```

2.5 Operation on string and string function

1) PHP strtolower() function

The strtolower() function returns string in lowercase letter.

Syntax

string strtolower (string $string)

Example

```
<?php
$str="My name is MHAN";
$str=strtolower($str);
```

```
echo $str;
?>
```
Output:

```
my name is MHAN
```

2) PHP strtoupper() function

The strtoupper() function returns a string in uppercase letter.

Syntax

```
string strtoupper ( string $string )
```

Example

```
<?php
$str="My name is MHAN";
$str=strtoupper($str);
echo $str;
?>
```

Output:

```
MY NAME IS MHAN
```

3) PHP ucfirst() function

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

Syntax

```
string ucfirst ( string $str )
```

Example

```
<?php
$str="my name is MHAN";
$str=ucfirst($str);
echo $str;
?>
```

Output:

```
My name is MHAN
```

4) PHP lcfirst() function

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

Syntax

```
string lcfirst ( string $str )
```

Example

```
<?php
$str="MY name IS MHAN";
$str=lcfirst($str);
echo $str;
?>
```
Output:

```
mY name IS MHAN
```

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

Syntax

```
string ucwords ( string $str )
```
Example

```
<?php
$str="my name is Sonoo jaiswal";
$str=ucwords($str);
echo $str;
?>
```
Output:

```
My Name Is Sonoo Jaiswal
```

6) PHP strrev() function
The strrev() function returns a reversed string.

Syntax

```
string strrev ( string $string )
```
Example

```
<?php
$str="my name is Sonoo jaiswal";
$str=strrev($str);
echo $str;
?>
```
Output:

```
lawsiaj oonoS si eman ym
```

7) PHP strlen() function
The strlen() function returns length of the string.

Syntax

```
int strlen ( string $string )
```
Example

```
<?php
$str="my name is Sonoo jaiswal";
$str=strlen($str);
echo $str;
?>
```
Output:

```
24
```

8. str_word_count() Function

The str_word_count() is in-built function of PHP. It is used to return information about words used in a string or counts the number of words in a string.

Syntax:

```
str_word_count(string,return,char)
```

```
<?php
 $str="PHP Javatpoint";
 echo "Your string is:".$str;
 echo "<br>";
echo "By using str_word_count(): ".str_word_count($str);
?>
```
Output:

```
Your string is:PHP Javatpoint
By using str_word_count(): 2
```

9. strpos() Function

The strops() is in-built function of PHP. It is used to find the position of the first occurrence of a string inside another string or substring in a string.

Syntax:

1. int strpos (string $haystack , mixed $needle [, int $offset = 0]);

Example 1

```
<?php
  $str1="Hello Php";
    $str2="Hello Php javatpoint!";
    echo "First string is: ".$str1;
    echo "<br>";
    echo "First string is: ".$str2;
    echo "<br>";
    echo "By using 'strpos()' function:".strpos("$str1,$str2","php");
    //$str1 and $str2 'Php'first letter is upper case so output is nothing.
    // It is case-sensitive
?>
```

Output:

```
First string is: Hello Php
First string is: Hello Php javatpoint!
By using 'strpos()' function:
```

2.6 Basic Graphics Function.

1. The next step to send content-Type header to the browser with appropriate content type for the kind of image being create.
header('Content-Type:image.jpeg');

2. Creating Image

1. The next step to send content-Type header to the browser with appropriate content type for the kind of image being create.
header('Content-Type:image.jpeg');

2. Creating Image
We can create an image in PHP using imagecreate() function.
Syntax $image=imagecreate(width,height)
3. imagecolorallocate() function:-The imagecolorallocate() function is an inbuilt function in PHP which is used to set the color in an image. This function returns a color which is given in RGB format.
Syntax:
imagecolorallocate ($image, $red, $green, $blue)

E.g

```
<?php
header ("Content-type: image/png");
$a = ImageCreate (200, 200);//imagecreate(x_size, y_size);
$bg = ImageColorAllocate ($a, 240, 0,0);
//imagecolorallocate(image, red, green, blue)
$txt_color = ImageColorAllocate ($a, 0, 23, 0);
ImageString ($a, 10, 10, 18, "Vesp Chembur Mumbai", $txt_color);//bool imagestring( $image,
$font, $x, $y, $string, $color )
imagepng($a);s
?>
```

Scaling Images:-
Scaling an image means making the image either smaller in size or larger in size than original.
Using PHP we can resize or rescale the image using the function Imagescale()

Syntax:
imagescale($image,$new_width,$new_height=1,$mode=IMG_BILINERA_FIXED)
Parameters:
$image is returned by one of the image creation function.
$new_width holds the width to scale the image.
$new_height holds the height to scale the image.
If the value of this parameter is negative or ommitted then aspect ration of image will preserve.$mode holds the mode. The value of this parameter is one of IMG_NEAREST_NEIGHBOUR, IMG_BILINEAR_FIXED, IMG_BICUBIC, IMG_BICUBIC_FIXED ...

Creation of PDF document

FPDF is an open source library which is used for creating a PDF document.it is open source.Link to Download latest version of FPDF class: http://www.fpdf.org/en/download.php
Features of fpdf
1) It is an open source package,hence freely available on internet
2) it provides the choice of measure unit,page format and margins for pdf page
3)it provides page header and footer management.
4) It provides automatic page breaks to the pdf document
5) it provides the support for various fonts,colors,encoding and image formate.

E.g

```
<?php
require('fpdf183/fpdf.php');
$pdf=new FPDF();
$pdf->AddPage();
$pdf->SetFont('Arial','B',16);
$pdf->Cell(60,10,'Hello PHP World!',1,1,'C');
 $pdf->Output();
?>
```

Function use

$pdf=new FPDF("L", "mm", "A4");
1. L(landscape default), P can be used instead to default to portrait pages.
2.mm:-default measurement unit, a choice of point (pt), millimeter (mm), centimeter (cm) and inch (in)
3. size of the page, the choice of A3, A4, A5, Letter & Legal is given.

$pdf->AddPage();
Simple function, just add the page, you can tell the function to create either a portrait (P) or landscape (L) by giving it as a first value (ex: $pdf->AddPage("L"), $pdf->AddPage("P")).
$pdf->SetFont('Arial','BIU',38);
1.font:- Arial,
2.BIU' simply tells that we want it to be Bold, Italic & Underlined.
3.38 mm in size (because of the default size unit).

$pdf->SetTextColor(0,0,255);

1.color :- red (r), the second is green (g) & blue (b).

$pdf->Cell(60,20,'PDF+PHP Test',0,1,C,0);
1.60 mm of width & 20 mm of height,
2.String:- ‘PDF+PHP Test’
3.Border: 0 means we do not want a border.
4. Line position:The 1 next to it means it will go to the beginning of the next line, if 0 is provided, then it will be to the right of it, if 2 is provided then it will go below. 5.Alignment:The C is just the alignment which is the center of the text inside the box, possible values are left (L), center (C), right (R).

$pdf->Output();
Output a new colorful PDF file!