Dr. SNS RAJALAKSHMI COLLEGE OF ARTS AND SCIENCE
(Autonomous)
Coimbatore-641049.
Accredited by NAAC (Cycle-III) with 'A+' Grade
(Recognised by UGC, Approved by AICTE, New Delhi and
Affiliated to Bharathiar University,Coimbatore)
ESTD : 1999

FUNCTIONS & ARRAYS

Course Name: PHP Programming

Course code: 21PCA105

UNIT: II

Prepared By : Dr.A.DEVI

* Expressions :-

↳ Expressions are the most important building blocks of PHP. In PHP, almost anything you write is an expression.

↳ The simplest way to define an expression is "anything that has a value".

↳ In other words, an expression is madeup of with variables and operators, that evalutes to a single value.

Ex:-

```
$a = 15;
$b = 20;
$c = $a + $b;
```

---

* Arrays :-

↳ An array is a special variable, which can hold more than one value at a time.

(or)

↳ An array can hold many values under single name, and we can access the values by referring to an index number.

→ In php, the array() function is used to create an array.

```
array();
```

→ There are 3 types of arrays
  ↳ Indexed Array
  ↳ Associative Array
  ↳ Multidimensional Array.

⇒ Indexed Array :-

→ In php, index is represented by number. which is starts from 0.

→ In this array, all elements are assigned to an index number by default.

→ There are two ways to define indexed array:

1st_Way :

```
$marks = array(60, 72, 66);
```

2nd_Way :

```
$marks[0] = 60;
$marks[1] = 72;
$marks[2] = 66;
```

Example :-

"IArray.php"

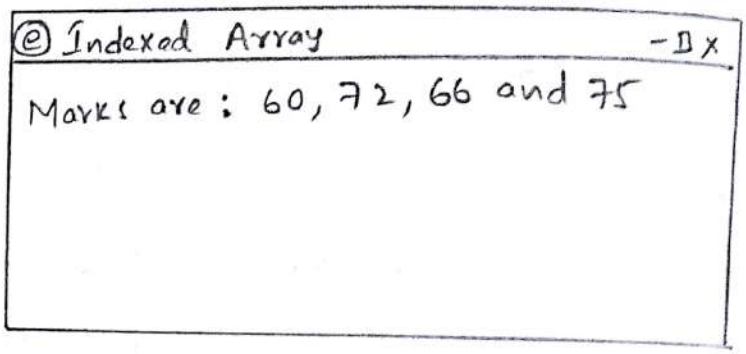
```
<html>
  <head>
    <title> Indexed Array </title>
  </head>
```

```
<body>
 <?php
    $marks = array(60, 72, 66, 75);
     echo "Marks are : $marks[0], $marks[1], $marks[
                                and $marks[3]";
 ?>
</body>
</html>
```

(or)

```
$marks[0] = 60;
$marks[1] = 72;
$marks[2] = 66;
$marks[3] = 75;
```

O/P:-



→ Associative Array :-

→ The associative arrays are very similar to Indexed arrays in term of funtionality but they are different in terms of their index.

→ Associative array will have their index as string so that we can establish a strong association between key and value.

→ In php, we can associate name with each array element using '=>' symbol.

→ There are two ways to define associative array.

1st way :

```
$marks = array("Madhu" => 60, "Kiran" => 72,
                "Giri" => 66, "Kalam" => 75);
```

2nd way :

```
$marks["Madhu"] = 60;
$marks["Kiran"] = 72;
$marks["Giri"] = 66;
$marks["Kalam"] = 75;
```

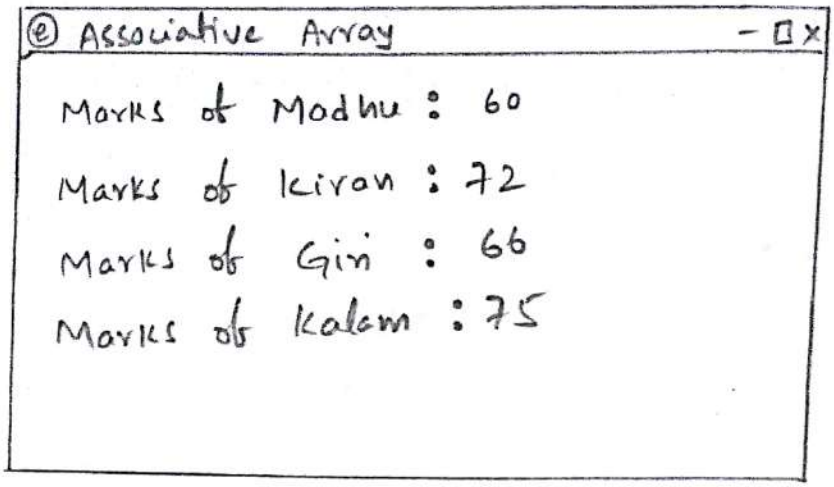
Example :-

"AArray.php"

```
<html>
 <head>
  <title> Associative Array </title>
 </head>
 <body>
  <?php
   $marks = array("Madhu" => 60, "Kiran" => 72,
                  "Giri" => 66, "Kalam" => 75);

   echo "Marks of Madhu :" . $marks["Madhu"] . "<br/>";
   echo "Marks of Kiran :" . $marks["Kiran"] . "<br/>";
   echo "Marks of Giri :" . $marks["Giri"] . "<br/>";
   echo "Marks of Kalam :" . $marks["Kalam"] . "<br/>";
  ?>
 </body>
</html>
```

o/p:-



⇒ Multidimensional Array :-

→ In php, Multidimensional array is also known as array of arrays. It allows you to store tabular data in an array.

→ Multidimensional array can be represented in the form of Matrix which represented by rows and columns.

Defn:

```
$students = array (
            array (501, "Hari", 72),
            array (502, "Madhu", 62),
            array (503, "Naveen", 82)
            );
```

→ A Multidimensional array is an array containing one or more arrays. php understands multidimensional arrays that are two, three, four, five (or) more levels deep.

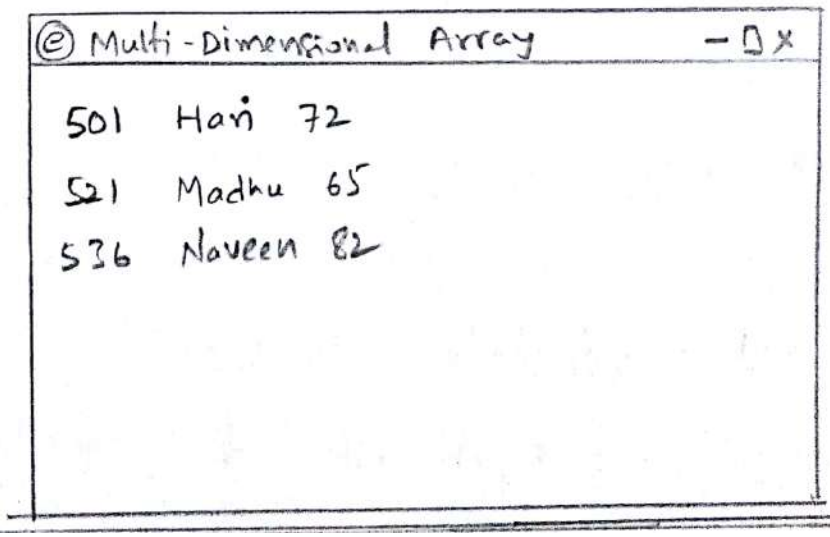
Example :-

"MDArray.php"

```
<html>
  <head>
    <title> Multi-Dimensional Array </title>
  </head>
  <body>
  <?php
    $students = array (
                  array (501, "Hari", 72),
                  array (521, "Madhu", 65);
                  array (536, "Naveen", 82));

    for ($i=0; $i<3; $i++)
    {
    for ($j=0; $j<3; $j++)
    {
      echo $students [$i] [$j]." ";
    }
     echo "<br/>";
    }
  ?>
  </body>
</html>
```

O/P:-



* Strings :-

↳ A string is a sequence of characters i.e used to store and manipulate text.

↳ There are 2 ways to specify string in PHP.

→ Single quotes Ex:- $str = 'Hello world';

→ Double quotes Ex:- $str = "Hello world";

→ Where in single quoted string, we can store multi-line text, special characters and escape sequences.

→ Where in Double quoted string, we can't able use special characters directly.

Example:- $str = ' php stands for "Hypertext preprocessor" ';

echo $str; o/p: php stands for "Hypertext pre processor"

$str = " php stands for "Hypertext preprocessor" ";

echo $str; o/p: parse error, syntax error.

⇒ String functions in php :-

↳ PHP provides various string functions to access and manipulate strings.

↳ A list of important string functions are

1. strtolower() :-

→ It returns string in lowercase letter.

Syntax :- strtolower (string $str)

Example :-
```
<?php
$str = "My name is MADHU";
$slr = strtoLower ($str);
echo $str;
?>
```
O/P :- my name is madhu

2. strtoupper() :-

→ It returns string in upper case letter.

Syntax :- strtoupper (string $str)

Example :-
```
<?php
$slr = "madhu";
$str = strtoupper ($str);
echo $str;
?>
```
O/P :- MADHU

3. ucwords() :-

→ It returns string converting first character of each word into uppercase

Syntax :- ucwords (string $str)

Example :-
```
<?php
$str = "my name is madhu";
$slr = ucwords ($str);
echo $str;
```
O/P :- My Name Is Madhu

4. strlen() :-

→ It returns length of the string.

Syntax:- strlen(string $str)

Example:-
```
<?php
$str = "Madhu T";
$len = strlen($str);
echo $len;
?>
```
o/p:- 7

5. strrev() :-

→ It returns reversed string.

Syntax:- strrev(string $str)

Example:-
```
<?php
$str = "Madhu T";
$str = strrev($str);
echo $str;
?>
```
o/p:- T uhdam

6. str_word_count() :-

→ It counts the number of words in a string.

Syntax:- str_word_count(string $str)

Example:-
```
<?php
$str = "Hello World!";
$wc = str_word_count($str);
echo $wc
?>
```
o/p:- 2

7. strpos() :-

→ It 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.

Syntax :- strpos($str, $text)

Example :-
```
<?php
echo strpos("Hello world!", "world");
?>
```
o/p: 6.

8. str_replace() :-

→ It replaces some characters with some other characters in a string.

Syntax :- str_replace($oText, $nText, $str)

Example :-
```
<?php
echo str_replace("world", "Madhu", "Hello world!");
?>
```
o/p :- Hello Madhu!

9. substr() :-

→ It returns a sub part of a string.

Syntax :- substr($string, $start, $length)

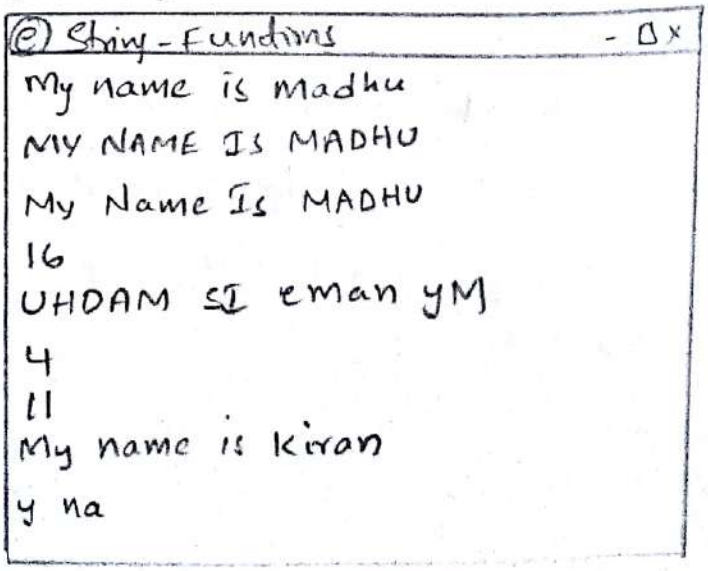
Example :-
```
<?php
echo substr("Hello world", 6);
echo substr("Hello world", 1, 4);
?>
```
o/p :- world
ello

Example :-

"StringFunDemo.php"

```
<html>
  <head>
    <title> String-Functions </title>
  </head>
  <body>
    <?php
      $str = "My name is MADHU";
      echo strtolower($str); echo "<br/>";
      echo strtoUpper($str); echo "<br/>";
      echo ucwords($str); echo "<br/>";
      echo strlen($str); echo "<br/>";
      echo strrev($str); echo "<br/>";
      echo str_word_count($str); echo "<br/>";
      echo strpos($str, "MADHU"); echo "<br/>";
      echo str_replace("MADHU", "Kiran", $str); echo "<br/>";
      echo substr($str, 1, 4);
    ?>
  </body>
</html>
```

O/P



* Functions:-

→ A function is a piece of code that is used to perform a particular task.

→ PHP supports both built-in and user-defined functions.

→ The main advantage of functions is that code-resusability. (Write once Invoke Multiple).

→ PHP supports thousands of built-in functions.

→ And, php allows the user to define own functions, by using "function" Keyword.

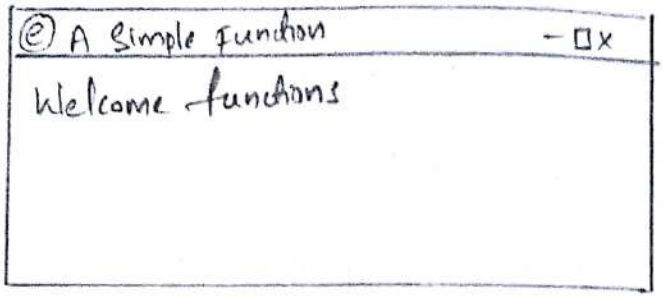
Syntax:-
```
function functionname()
{
  //code
}
```

Note:- Function name must be start with letter and underscore only.

Example:- "simplefun.php"
```
<html>
<head>
  <title> A simple function </title>
</head>
<body>
 <?php
   function msg(){ //defining function
    } echo "welcome functions";
```

```
msg();// calling function.
?>
</body>
</html>
```

o/p:-



→ parameterized functions are functions with parameters. you can pass any number of parameters inside a function.

→ we can pass the information in function through arguments which are separated by comma.

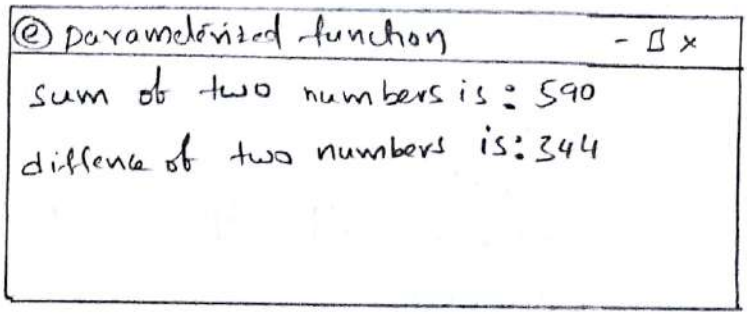
→ These passed parameters (or) arguments acts as variables inside your function.

Example:- "paramfun.php"

```
<html>
<head>
  <title> parameterized functions </title>
</head>
<body>
<?php
  function add ( $x, $y) // defining function
  {
    $sum = $x + $y;
    echo " sum of two numbers is : $sum <br/>";
  }
  add (467, 123); // calling function
```

```
function sub($x, $y) // defining function
{
  $diff = $x - $y;
 echo " diffence of two numbers is : $diff";
}
sub(467, 123); // calling function
?>
</body>
</html>
```

o/p:



→ PHP allows you to call function by value and reference.

* call by value:

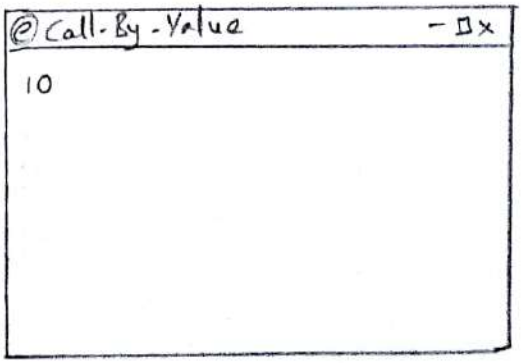
In case of call by value, actual value is not modified if it is modified inside the function.

"callbyvalueFun.php"

Example:-

```
<html>
<head> <title> Call-By-Value </title> </head>
<body>
<?php
function increment($i)
{
  $i++;
}
$i=10;
increment($i);
echo $i;
</body>
</html> ?>
```

o/p:



* call by reference :

→ In case of call by reference, actual value is modified if it is modified inside the function.

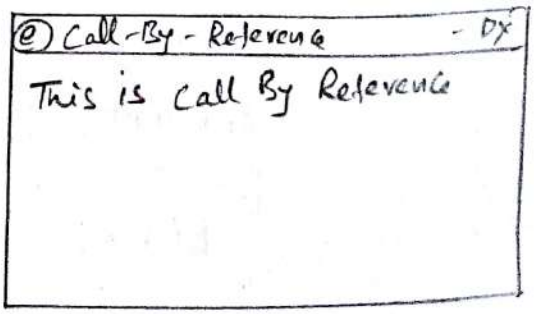
→ In such case, you need to use & (ampersand) symbol with formal arguments.

→ The & represents reference of the variable.

Example :- "callbyreferenceFun.php"

```
<html>
<head>
<title> Call - By - Reference </title>
</head>
<body>
<?php
function adder(&$str2)
{
  $str2 .= 'Call By Reference';
}
$str = 'This is ';
adder($str);
echo $str;
?>
</body>
</html>
```

o/p :-

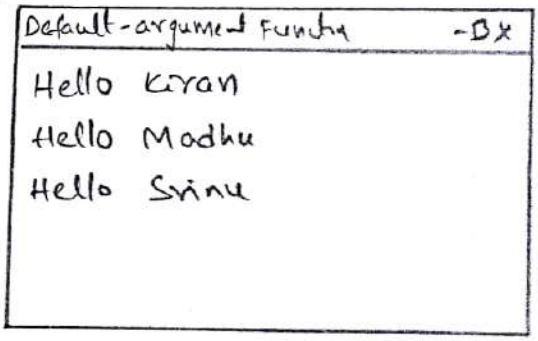


→ PHP allows you to define default argument values. In such case if you don't pass any value to the function, it will use default argument value.

Example:-

```
<html>
<head>
<title> Default-argument Function </title>
</head>
<body>
<?php
function msg ($name = "Madhu")
{
echo " Hello $name <br/>";
}
msg("Kiran");
msg();
msg("Srinu");
?>
</body>
</html>
```

O/P:



Example:-

```
<?php
function add($n1=10, $n2=10){
$n3 = $n1 + $n2;
echo " Addition is : $n3 <br/>";
}
add();
add(20);
add(20,40);
?>
```

O/P:

Addition is : 20
Addition is : 30
Addition is : 60

* Recursive Function :-

→ PHP also supports recursive function. In such case, we call current function within function. It is also known as recursion.

"RecursiveFun.php"

Example :-

```
<html>
<head>
 <title> Recursive Function </title>
</head>
<body>
 <?php
  function factorial ($n)
  {
    if ($n < 0)
      return -1;
    if ($n == 0)
      return 1;
    return ($n * factorial ($n-1));
  }
  echo factorial (5);
 ?>
</body>
</html>
```

O/P