UNIT 2 PHP ARRAYS AND OBJECTS, STRINGS

1. EXPLAIN ABOUT ARRAYS?

- ★ ARRAY is a collection of elements with similar data type. You can store number of values in a single variable by using array.
- ★ You can easily retrieve the array elements by using their index. Array index is always starts with '0'.
- ★ In php, arrays are classified into 3 types. They are
  - ✓ Indexed array
  - ✓ Associative array
  - ✓ Multidimensional array

Indexed or Numeric Arrays:

An array with a numeric index where values are stored linearly. PHP index is represented by number which starts from 0. We can store number, string and object in the PHP array. All PHP array elements are assigned to an index number by default.

There are two ways to define indexed array: 1st way:

```
$season=array("summer", "winter", "spring", "autumn");
```

2ndway:

```
$season[0]="summer";
$season[1]="winter";
$season[2]="spring";
$season[3]="autumn";
```

Eg:

```
<?php
$season=array("summer", "winter", "spring", "autumn");
Echo "Season are:$season[0],$season[1],$season[2]and $season[3]";
?>
```

Associative Arrays:

- ★ An array with a string index where instead of linear storage, each value can be assigned a specific key.
- ★ These types of arrays are similar to the indexed arrays but instead of linear storage, every value can be assigned with a user-defined key of string type.
- ★ We can associate name with each array elements in PHP using => symbol.

There are two ways to define associative array:

1st way:

```
$salary=array("Sonoo"=>"350000","John"=>"450000","Kartik"=>"200000");
```

2nd way:

```
$salary["Sonoo"]="350000";
$salary["John"]="450000";
$salary["Kartik"]="200000";
```

Example

File: arrayassociative1.php

```
<?php
$salary=array("Sonoo"=>"350000","John"=>"450000","Kartik"=>"200000");

echo "Sonoo salary: ".$salary["Sonoo"]."<br/>";

echo "John salary: ".$salary["John"]."<br/>";

echo "Kartik salary: ".$salary["Kartik"]."<br/>";

?>
```

PHP Multidimensional Array

PHP multidimensional array is also known as array of arrays. It allows you to store tabular data in an array. PHP multidimensional array can be represented in the form of matrix which is represented by row * column.

```
$emp=array(array(1,"sonoo",400000),array(2,"john",500000),array(3,"rahul",300000));
```

Example

Id	Name	Salary
1	sonoo	400000
2	john	500000
3	rahul	300000

```
<?php
$emp =array
  (
  array(1,"sonoo",400000),
  array(2,"john",500000),
  array(3,"rahul",300000)
  );
  for ($row = 0; $row < 3; $row++) {
  for ($col = 0; $col < 3; $col++) {
  echo $emp[$row][$col]." ";
  }
  echo"<br/>";
}
?>
```

2.WRITE ABOUT PHP CLASSES AND OBJECTS?

- ★ A class is a template for objects, and an object is an instance of class. A class is a collection of data members and data methods.
- ★ A class is defined by using the class keyword, followed by the name of the class and a pair of curly braces ({}). All its properties and methods go inside the braces:
  Syntax:

```
Class className{
//code;
}
```

Eg:

```
<? Php
Class book {
Public $price;
Public $title;
}
```

Below we declare a class named Fruit consisting of two properties ($name and $color) and two methods set _name () and get_ name () for setting and getting the $name property:

Eg:

```
 <?php
 Class Book{
   public $price; //Properties
    public  $title;
Function set_ price($price){  //Methods
     $this->price=$price;
   }
   function get_ price( ) {
    return $this>price; } } ?>
```

When the individual objects are created, they inherit all the properties and behaviors from the class, but each object will have different values for the properties.

Define Objects

Classes are nothing without objects! We can create multiple objects from a class. Each object has all the properties and methods defined in the class, but they will have different property values.
Objects of a class are created using the new keyword.

Example:

```
 <?php
 Class Book{
   public $name;
   public $price;
//Methods
   Function set_ name($name){
     $this->name=$name;
   }
   function get_ name() {
    return $this->name;
   }
   Function set_ price($price){
     $this->price=$price;
   }
   }
$computer CS =new Book ( );
     CS.$PRICE( ); ?>
```

```
$computer->set_ name('cs');
$computer->set_price('750');
Echo "Name:
".$computer>get_name();
echo "<br>";
echo"price:".$computer->get_price();
```

3. **WRITE ABOUT PHP ARRAY FUNCTIONS?**

Array functions are used to perform specific task. Php provides several functions. They are

COUNT ( ): To count the no of elements in array.

Syntax &Eg:

```
$colors=array ('red', 'green', 'blue');
```

Echo count ($colors); output:3

SORT ( ): To be arranged the data in ascending order. It is used to apply for numbers.

Syntax: $var2 =sort ($var1);

Eg: $numbers=array (‘10’, ‘5’ ‘25’);

Echo sort($numbers); output: 5 10 25

RSORT ( ): To be arranged the data in descending order. It is used to apply for numbers.

Syntax: rsort ($var1);

Eg: $numbers=array (‘10’, ‘5’ ‘25’);

Echo rsort ($numbers); output: 25 10 5

ASORT ( ): To be arranged the data in ascending order. It is used to apply for text.

Syntax: asort ($var1);

Eg: $colors=array (‘red’, ‘green’, ‘blue’);

echo asort ($colors); output: blue green red

ARSORT ( ): To be arranged the data in descending order. It is used to apply for text.

Syntax: arsort ($var1);

Eg: $colors=array (‘red’, ‘green’, ‘blue’);

Echo arsort ($colors); output: red green blue

ARRAY_POP ( ): To delete the last element of array.

Syntax:
```
<?php
$a=array("red","green","blue");
array_pop($a);
print_r($a);
?>
```
output: red green

ARRAY_PUSH ( ): To add the element in existing array.

Syntax:
```
<?php
$a=array("red","green");
array_push($a,"blue","yellow");
print_r($a);
?>
```
output: red green blue yellow

SEARCH ( ): To search an element in existing array.

Syntax:
```
<?php
$a=array("a"=>"red","b"=>"green","c"=>"blue");
echo array_search("red",$a);
?>
```
output: a

SUM ( ): To calculate addition two or more numbers.

Syntax:
```
<?php
$a=array(5,15,25);
echo array_sum($a);
?>
```
output: 45

UNIQUE ( ): The array_unique() function removes duplicate values from an array. If two or more array values are the same, the first appearance will be kept and the other will be removed.

Syntax:
```
<?php
$a=array("a"=>"red","b"=>"green","c"=>"red");
print_r(array_unique($a));
?>
```
output: red greeen

Combine ( ): The array_combine() function creates an array by using the elements from one "keys" array and one "values" array.

Syntax:
```
<?php
$fname=array("Peter","Ben","Joe");
$age=array("35","37","43");
$c=array_combine($fname,$age);
```

```
print_r($c);
?>                                  output: peter:35 ben 37 joe 43
```

DIFFERENCE ( ): The array_ diff () function compares **the values** of two (or more) arrays, and returns the differences. This function compares the values of two (or more) arrays, and return an array that contains the entries from *array1* that are not present in *array2* or *array3*, etc.

```
<?php
$a1=array("a"=>"red","b"=>"green","c"=>"blue","d"=>"yellow");
$a2=array("e"=>"red","f"=>"green","g"=>"blue");
$result=array_diff($a1,$a2);
print_r($result);
?>                              output: yellow
```

4.EXPLAIN ABOUT STRING FUNCTIONS?

In general practice, using the right string function will save you a lot of time as they are pre-defined in PHP libraries and all you have to do is call them to use them.

strlen($str)

This function returns the length of the string or the number of characters in the string including whitespaces.

```
<?php
$str="Welcome to Study tonight";
Echo "Length of the string is:" strlen ($str);
?>
```

```
Length of the string is: 23
```

str_word_count($str):

This function returns the number of words in the string.

```
<?php
$str="Welcome to Study tonight";
Echo "Number of words in the string are:".str_word_count($str);
?>
```

```
Number of words in the string are: 3
```

- **strrev($str)**

This function is used to reverse a string.

Syntax &eg:

```
<?php
$str = "Welcome to Study tonight";
echo "Reverse: ". strrev($str);
?>
```

```
Reverse: thginot ydutS ot emocleW
```

- **strpos($str,$text)**

This function is used to find the position of any text/word in a given string. Just like an array, string also assign index value to the characters stored in it, starting from zero

syntax & eg:

```
<?php
$str="Welcome to Study tonight";
echo"Positionof'Studytonight'instring:".strpos($str,'Studytonight');
?>
```

Position of 'Study tonight 'in string: 11

- **str_replace($replacethis,$replacewith,$str)**

Thisfunctionisusedtoreplaceapartofthestringwithsometext.Whileusingthis function, the first argument is the part of string that you want to replace, second argument is the new text you want to include, and the last argument is the string variable itself.

```
<?php
$str = str_replace("Studytonight", "Studytonight.com", "Welcome to Studytonight");
echo $str;
?>
```

WelcometoStudytonight.com

- **strtoupper($str)**

To convert every letter/character of every word of the string to uppercase, one canuse strtoupper() method.

```
<?php
$str = "welcome to studytonight"; echo
strtoupper($str);
?>
```

WELCOME TO STUDY TONIGHT

- **strtolower($str)**

This function is used to convert every letter/character of a string to lowercase.

```
<?php

$str="WELCOMETOSTUDYTONIGHT";

echostrtolower($str);

?>
```

Welcome to study tonight

- **str_repeat($str,$counter)**

This function is used to repeat a string a given number of times. The first argument is the string and the second argument is the number of times the string should be repeated.

```
<?php
$str = "Study to night";

echo str_repeat($str,4);

?>
```

Study to night Study to night Study to night Study to night

- **strcmp($str1,$str2)**

This function is used to compare two strings. The comparison is done alphabetically. If the first string is greater than second string, the result will be greater than 0, if the first string is equal to the second string, the result will be equal to 0 and if the second string is greater than the first string, then the result will be less than 0

Syntax&eg:

```
<?php
$str1="Studytonight";
$str2="Studytonight.com";
// comparing str1 and str2 echo
strcmp($str1, $str2);
// comparing str2 and str1 echo
strcmp($str2, $str1);
// comparing str1 with str1 echo
strcmp($str1, $str1);
?>
```

- **substr($str,$start, $length)**

This function is used to take out a part of the string(substring), starting from a particular position, of a particular length.
The first argument is the string itself, second argument is the starting index of the substring to be extracted and the third argument is the length of the substring to be extracted.

```
<?php
$str = "Welcome to Studytonight";

echo substr($str, 11, 15);

?>
```

- **($str,charlist)**

This function is used to remove extra whitespaces from beginning and the end of a string. The second argument charlist is optional. We can provide a list of character,just like a string, as the second argument, to trim/remove those characters from the main string.

```
<?php
$str1 = "Hello World";

echo trim($str1)."<br/>";

$str2 = "Hello Hello";

echo trim($str2,"Heo");

?>
```

```
HelloWorld

llo Hell
```

As you can see in the output, additional spaces from the beginning and end are removed and in the second case, the characters specified are removed from the beginning and the end of the string.

- **explode(separator,$str,$limit)**

This function is used to break a string, create an array of the broken parts of the string and return the array. The first argument, **separator** defines where to break the string from. It can be a space, hiphen(-) or any other character

The second argument of this function is the string itself and the third argument is the **limit**, which specifies the number of array elements to return. the third argument is optional.

```
<?php
$str = "Its a beautiful day";
print_r(explode("",$str));
?>
```

```
Array(
    [0]=>Its
    [1]=>a
    [2]=>beautiful
    [3]=>day
)
```

In the example above, we have provided **space** as separator to break the string and return an array.

If we provide the third argument **limit** as well, we can limit the number of array elements returned. For example, if we provide 2 as the third argument, then we will only get 2 elements in the array, the first two.

- **implode(separator,$arr)**

This function is used to form a string using the array elements from the array provided and join them using the **separator**.

```
<?php
$arr=array("Its","a","beautiful","day");
// <br> is used to jump to next line echo

implode("", $arr) . "<br>";

echoimplode("-",$arr)."<br>";
echoimplode("/",$arr)."<br>";
?>
```

```
Its a beautiful day

Its-a-beautiful-day
```

5.WRITE ABOUT DATE AND TIME FUNCTIONS?

★ The date & time using the date() function in PHP, we will also see the various formatting

options available with these functions& understand their implementation through the examples.

- ★ Date and time are some of the most frequently used operations in PHP while executing SQL queries or designing a website etc.
- ★ PHP serves us with predefined functions for these tasks. Some of the predefined functions in PHP for date and time are discussed below.

➢ **PHP date() Function:**
The PHP date() function converts time stamp to a more readable date and time format.

Syntax:

Date (format, timestamp)

Explanation:
The format parameter in the date() function specifies the format of returned date and time. The timestamp is an optional parameter, if it is not included then the current date and time will be used.

```
<?php
  Echo "Today's date is:";
  $today=date("d/m/Y");

   echo $today;

?>
```

```
Today'sdateis:05/12/2017
```

Formatting options available in date() function:

The format parameter of the date() function is a string that can contain multiple characters allowing to generate the dates in various formats.

Date-related formatting characters that are commonly used in the format string:

- ✓ d:Represents day of the month; two digits with leading zeros(01or31).
- ✓ D:Represents day of the week in the text as an abbreviation(MontoSun).
- ✓ m:Represents month in numbers with leading zeros(01or12).
- ✓ M:Represents month in text, abbreviated(Jan toDec).
- ✓ y:Represents year in two digits (08 or 14).
- ✓ Y:Represents year in four digits (2008 or 2014).

The parts of the date can be separated by inserting other characters, like hyphens (-), dots(.),slashes(/),or spaces to add additional visual formatting.

```
<?php
  echo"Today'sdateinvariousformats:"."\n"; echo
  date("d/m/Y") . "\n";
  echodate("d-m-Y")."\n";
  echodate("d.m.Y")."\n";
  echo date("d.M.Y/D");

 ?>
```

```
Today's date in various

formats: 05/12/2017

05-12-2017

05.12.2017

05.Dec.2017/Tue
```

The following characters can be used along with the date() function to format the time string:

- ✓ h:Represents hour in 12-hour format with leading zeros (01 to 12).
- ✓ H:Represents hour in 24-hour format with leading zeros (00 to 23).
- ✓ i:Represents minutes with leading zeros (00 to 59).
- ✓ s:Represents seconds with leading zeros (00 to 59).
- ✓ a:Represents lowercase antemeridian and postmeridian(am or pm).
- ✓ A:Represents uppercase antemeridian and postmeridian(AM or PM).

```
<?php
  echodate("h:i:s")."\n";
  echodate("M,d,Yh:i:sA")."\n"; echo
  date("h:i a");
?>
```

```
03:04:17

Dec,05,201703:04:17PM

03:04pm
```