

## PHP Recursive Function

PHP also supports recursive function call like C/C++. In such case, we call current function within function. It is also known as recursion.

It is recommended to avoid recursive function call over 200 recursion level because it may smash the stack and may cause the termination of script.

### Example 1: Printing number

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

Output:

1 2 3 4 5

---

## What is Apache?

Apache is free and open-source software of web server that is used by approx 40% of websites all over the world. Apache HTTP Server is its official name. It is developed and maintained by the Apache Software Foundation. Apache permits the owners of the websites for serving content over the web. It is the reason why it is known as a "web server." One of the most reliable and old versions of the Apache web server was published in 1995.

If someone wishes to visit any website, they fill-out the name of the domain in their browser address bar. The web server will bring the requested files by performing as the virtual delivery person.

### Web Server Meaning

Mail servers, database servers, web servers, and file servers use different types of server software. All these applications may access a lot of files saved on the physical server and apply them for many objectives.

The aim of the web servers is to deliver websites over the internet. It behaves as a middleman among the client machines and servers to achieve that aim. It can pull the content through the server over every user request. Also, it delivers this request to the web.

One of the most critical tasks of any web server is to provide services to various different users of the web at the same time. Web servers execute files specified in a different type of programming languages like Java, Python, PHP, and many others. Web servers turn these files into static HTML files. It provides services to these files within the web server browser. Web browser can be defined as a tool liable for decent client-server communication.

---

## 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) );
```

### PHP Multidimensional Array Example

Let's see a simple example of PHP multidimensional array to display following tabular data. In this example, we are displaying 3 rows and 3 columns.

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/>";
```

```
}
```

```
?>
```

### Output:

```
sonoo 400000
```

```
john 500000
```

```
rahul 300000
```

## Sorting Arrays

In the previous chapter you've learnt the essentials of PHP arrays i.e. what arrays are, how to create them, how to view their structure, how to access their elements etc. You can do even more things with arrays like sorting the elements in any order you like.

PHP comes with a number of built-in functions designed specifically for sorting array elements in different ways like alphabetically or numerically in ascending or descending order. Here we'll explore some of these functions most commonly used for sorting arrays.

- `sort()` and `rsort()` — For sorting indexed arrays
- `asort()` and `arsort()` — For sorting associative arrays by value
- `ksort()` and `krsort()` — For sorting associative arrays by key

### Sorting Indexed Arrays in Ascending Order

The `sort()` function is used for sorting the elements of the indexed array in ascending order (alphabetically for letters and numerically for numbers).

### Sorting Indexed Arrays in Descending Order

The `rsort()` function is used for sorting the elements of the indexed array in descending order (alphabetically for letters and numerically for numbers).

### Sorting Associative Arrays in Ascending Order By Value

The `asort()` function sorts the elements of an associative array in ascending order according to the value. It works just like `sort()`, but it preserves the association between keys and its values while sorting.

### Sorting Associative Arrays in Descending Order By Value

The `arsort()` function sorts the elements of an associative array in descending order according to the value. It works just like `rsort()`, but it preserves the association between keys and its values while sorting.

---

## How to change date format in PHP?

To convert the date-time format PHP provides **`strtotime()`** and **`date()`** function. We change the date format from one format to another. **For example** - we have stored date in MM-DD-YYYY format in a variable, and we want to change it to DD-MM-YYYY format.

We can achieve this conversion by using `strtotime()` and `date()` function. These are the built-in functions of PHP. The `strtotime()` first converts the date into the seconds, and then `date()` function is used to reconstruct the date in any format. Below some examples are given to convert the date format.

Change YYYY-MM-DD to DD-MM-YYYY

In the below example, we have date 2019-09-15 in YYYY-MM-DD format, and we will convert this to 15-09-2019 in DD-MM-YYYY format.

```
<?php
    $orgDate = "2019-09-15";
    $newDate = date("d-m-Y", strtotime($orgDate));
    echo "New date format is: ".$newDate." (MM-DD-YYYY)";
?>
```

### Output

New date format is: 15-09-2019 (DD-MM-YYYY)

---

## Merging arrays

The `array_merge()` function merges one or more arrays into one array. It returns an array in which the elements of all arrays passed in parameters are merged.

**Note** – In case of same keys of two or more array elements, the last one overrides the other.

Syntax

```
array_merge(arr1, arr2, arr3, ...)
```

### Parameters

- **arr1** – Initial array to merge
- **arr2** – Another array
- **arr3** – Another array

### Return

The `array_merge()` function returns an array in which the elements of all arrays passed in parameters are merged.

The following is an example that merges two array with a key repeated in the second array. In this case the last one overrides the other.

### Example

```
<?php
$arr1 = array("p"=>"red","q"=>"green");
$arr2 = array("p"=>"blue","r"=>"yellow");
print_r(array_merge($arr1,$arr2));
?>
```

### Output

```
Array
( [p] => blue
  [q] => green
  [r] => yellow )
```

---

## Overloading

Method Overloading is a concept of Object Oriented Programming which helps in building the composite application in an easy way. Function overloading or method overloading is a feature that permits making creating several methods with a similar name that works differently from one another in the type of the input parameters it accepts as arguments.

The above concept is fine for other programming languages and it is called static polymorphic i.e method overloading.

### Example

Let's understand through an example.

```
<?php
class machine {
    function doTask($var1){
        return $var1; }
    function doTask($var1,$var2){
        return $var1 * $var1 ; } }
$task1 = new machine();
$task1->doTask(5,10);
?>
```

### Output:

Error

---

## Classes & Objects

### Class:

Like C++ and Java, PHP also supports object oriented programming

1. Classes are the blueprints of objects. One of the big differences between functions and classes is that a class contains both data (variables) and functions that form a package called an: 'object'.
2. Class is a programmer-defined data type, which includes local methods and local variables.
3. Class is a collection of objects. Object has properties and behavior.

**Syntax:** We define our own class by starting with the keyword '**class**' followed by the name you want to give your new class.

```
<?php
    class person {
    }?>
```

### Objects:

An **Object** is an individual instance of the data structure defined by a class. We define a class once and then make many objects that belong to it. Objects are also known as instances.

### Creating an Object:

Following is an example of how to create object using **new** operator.

```
class Books {
    // Members of class Books
}
// Creating three objects of Books
$physics = new Books;
$maths = new Books;
$chemistry = new Books;
```

---

## Basic Structure of PHP

The server executes the PHP script, which will send the HTML output to the browser. Normally HTML and PHP tags can be present. A popular open source general programming language, PHP or Hypertext Preprocessor may be built into an HTML. The .php extension is used to store PHP files. In PHP tags, PHP scripts and standard HTML may be written in any page.

For beginners or pros alike, PHP is a pleasant language to use. We start working with PHP in this PHP Tutorial Series on the first floor. Even more experienced developers who need refreshment are very great to start from the start. Frequently you will realize that sophisticated difficulties are often a fairly fundamental mistake. That's what makes the fundamentals so vital. They are the foundation for building on your whole understanding of programming.

### Examples:-

```
<!DOCTYPE html>
<html>
    <body>
        <h1>ScmGalaxy</h1>

        <?php
            echo 'Hello ScmGalaxy';
        ?>
    </body>
</html>
```

## Overloading

Method Overloading is a concept of Object Oriented Programming which helps in building the composite application in an easy way. Function overloading or method overloading is a feature that permits making creating several methods with a similar name that works differently from one another in the type of the input parameters it accepts as arguments.

The above concept is fine for other programming languages and it is called static polymorphic i.e method overloading.

### Example

Let's understand through an example.

```
<?php
class machine {
    function doTask($var1){
        return $var1;
    }
    function doTask($var1,$var2){
        return $var1 * $var1 ; } }
$task1 = new machine();
$task1->doTask(5,10);
?>
```

Output:

Error

---

## Inheritance

It is a concept of accessing the features of one class from another class. If we inherit the class features into another class, we can access both class properties. We can extend the features of a class by using 'extends' keyword.

- It supports the concept of **hierarchical classification**.
- Inheritance has three types, **single, multiple and multilevel Inheritance**.
- **PHP** supports only **single inheritance**, where only one class can be **derived from single parent class**.
- We can simulate multiple inheritance by using **interfaces**.

### Example

```
<?php
class a
{ function fun1() { echo "javatpoint"; } }
class b extends a
{ function fun2()
{ echo "SSSIT"; } }
$obj= new b();
$obj->fun1();
?>
```

Output:

Javatpoint

---

## Constructors and Destructors

### Introduction

In object oriented programming terminology, constructor is a method defined inside a class is called automatically at the time of creation of object. Purpose of a constructor method is to initialize the object. In PHP, a method of special name **\_\_construct** acts as a constructor.

### Syntax

```
__construct ([ mixed $args = "" [, $... ] ] ) : void
```

### Constructor example

This example shows that constructor is automatically executed when object is declared

Example

```
<?php
class myclass{
    function __construct(){
        echo "object initialized";
    }
}
$obj=new myclass();
?>
```

### Output

This will produce following result. –

object initialized

### Destructor

Destructor is a method automatically as soon as garbage collector finds that a particular object has no more references. In PHP, destructor method is named as **\_\_destruct**. During shutdown sequence too, objects will be destroyed. Destructor method doesn't take any arguments, neither does it return any data type

Example

```
<?php
class myclass{
    function __construct(){
        echo "object is initialized";
    }
    function __destruct(){
        echo "object is destroyed";
    }
}
$obj=new myclass();
?>
```

### Output

This will show following result

object is initialized

object is destroyed

## PHP Timestamp

PHP provides several date-time functions to perform required operations with temporal data. Now, we are going to see about PHP timestamp functions. The timestamp is the value represented as seconds calculated, since UNIX Epoch, January 1, 1970, and also called as UNIX timestamp.

In PHP, it includes several functions to work with a timestamp. In this article, we are going to see how the following list of timestamp-related functionalities is obtained by using PHP date-time functions.

- getting current timestamp
- date/time to timestamp conversion

### Getting Current Timestamp in PHP

We can get the current timestamp value in three possible ways with the help of the PHP core functions described here.

#### time()

This is the simple and widely used PHP function to get the current timestamp value. It requires no arguments to be sent for returning the expected resultant UNIX timestamp. The usage of this simple function is shown in the example PHP program below.

```
<?php
$current_timestamp = time();
echo $current_timestamp;
?>
```

#### Date/Time to Timestamp Conversion

strtotime() and mktime() functions are also used to convert specified date into the form of a timestamp.

For using strtotime(), we need to pass the date with any one of PHP-supported date formats, for example, dd/mm/yyyy, mm/dd/yyyy and etc. And, for using mktime(), we need to explode the given date and send the exploded components to this function. And, we can perform the reverse, that is, converting timestamp value to date, by the use of date() function. For that, we should specify the required date format as the first parameter, and timestamp as the second one. For example,

```
<?php
$date_from_timestamp = date("d-m-Y", $current_timestamp);
echo "Formatted date from timestamp:" . $date_from_timestamp;
?>
```

---

## PHP Functions

PHP function is a piece of code that can be reused many times. It can take input as argument list and return value. There are thousands of built-in functions in PHP.

In PHP, we can define Conditional function, Function within Function and Recursive function also.

### Advantage of PHP Functions

**Code Reusability:** PHP functions are defined only once and can be invoked many times, like in other programming languages.

**Less Code:** It saves a lot of code because you don't need to write the logic many times. By the use of function, you can write the logic only once and reuse it.

**Easy to understand:** PHP functions separate the programming logic. So it is easier to understand the flow of the application because every logic is divided in the form of functions.

---