

## 1) PHP strtolower() function

The strtolower() function returns string in lowercase letter.

### Syntax

1. string strtolower ( string \$string )

### Example

1. <?php
2. \$str="My name is KHAN";
3. \$str=strtolower(\$str);
4. echo \$str;
5. ?>

Output:

```
my name is khan
```

## 2) PHP strtoupper() function

The strtoupper() function returns string in uppercase letter.

### Syntax

1. string strtoupper ( string \$string )

### Example

1. <?php
2. \$str="My name is KHAN";
3. \$str=strtoupper(\$str);
4. echo \$str;
5. ?>

Output:

```
MY NAME IS KHAN
```

## 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. string ucfirst ( string \$str )

### Example

1. <?php
2. \$str="my name is KHAN";
3. \$str=ucfirst(\$str);
4. echo \$str;
5. ?>

Output:

```
My name is KHAN
```

## 4) PHP lcfirst() function

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

### Syntax

1. string lcfirst ( string \$str )

### Example

1. <?php
2. \$str="MY name IS KHAN";
3. \$str=lcfirst(\$str);
4. echo \$str;
5. ?>

Output:

```
mY name IS KHAN
```

## 5) PHP ucwords() function

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

### Syntax

1. string ucwords ( string \$str )

### Example

1. <?php
2. \$str="my name is Sonoo jaiswal";
3. \$str=ucwords(\$str);
4. echo \$str;

5. ?>

Output:

```
My Name Is Sonoo Jaiswal
```

## 6) PHP strrev() function

The strrev() function returns reversed string.

### Syntax

1. string strrev ( string \$string )

### Example

```
1. <?php
2. $str="my name is Sonoo jaiswal";
3. $str=strrev($str);
4. echo $str;
5. ?>
```

Output:

```
lawsiaj oonoS si eman ym
```

## 7) PHP strlen() function

The strlen() function returns length of the string.

### Syntax

1. int strlen ( string \$string )

### Example

```
1. <?php
2. $str="my name is Sonoo jaiswal";
3. $str=strlen($str);
4. echo $str;
5. ?>
```

Output:

```
24
```

# PHP Math

PHP provides many predefined math constants and functions that can be used to perform mathematical operations.

## PHP Math: abs() function

The abs() function returns absolute value of given number. It returns an integer value but if you pass floating point value, it returns a float value.

### Syntax

1. number abs ( mixed \$number )

#### Example

1. <?php
2. echo (abs(-7)."<br/>"); // 7 (integer)
3. echo (abs(7)."<br/>"); //7 (integer)
4. echo (abs(-7.2)."<br/>"); //7.2 (float/double)
5. ?>

Output:

```
7
7
7.2
```

## PHP Math: ceil() function

The ceil() function rounds fractions up.

### Syntax

1. float ceil ( float \$value )

#### Example

1. <?php
2. echo (ceil(3.3)."<br/>");// 4
3. echo (ceil(7.333)."<br/>");// 8
4. echo (ceil(-4.8)."<br/>");// -4
5. ?>

Output:

```
4
8
-4
```

## PHP Math: floor() function

The floor() function rounds fractions down.

### Syntax

1. float floor ( float \$value )

#### Example

1. <?php
2. echo (floor(3.3)."<br/>");// 3
3. echo (floor(7.333)."<br/>");// 7
4. echo (floor(-4.8)."<br/>");// -5
5. ?>

Output:

```
3
7
-5
```

## PHP Math: sqrt() function

The sqrt() function returns square root of given argument.

### Syntax

1. float sqrt ( float \$arg )

#### Example

1. <?php
2. echo (sqrt(16)."<br/>");// 4
3. echo (sqrt(25)."<br/>");// 5
4. echo (sqrt(7)."<br/>");// 2.6457513110646
5. ?>

Output:

```
4
5
2.6457513110646
```

## PHP Math: decbin() function

The decbin() function converts decimal number into binary. It returns binary number as a string.

### Syntax

1. string decbin ( int \$number )

#### Example

1. <?php

2. `echo (decbin(2)."<br/>");// 10`
3. `echo (decbin(10)."<br/>");// 1010`
4. `echo (decbin(22)."<br/>");// 10110`
5. `?>`

Output:

```
10
1010
10110
```

## PHP Math: dechex() function

The `dechex()` function converts decimal number into hexadecimal. It returns hexadecimal representation of given number as a string.

### Syntax

1. `string dechex ( int $number )`

#### Example

1. `<?php`
2. `echo (dechex(2)."<br/>");// 2`
3. `echo (dechex(10)."<br/>");// a`
4. `echo (dechex(22)."<br/>");// 16`
5. `?>`

Output:

```
2
a
16
```

## PHP Math: decoct() function

The `decoct()` function converts decimal number into octal. It returns octal representation of given number as a string.

### Syntax

1. `string decoct ( int $number )`

#### Example

1. `<?php`
2. `echo (decoct(2)."<br/>");// 2`
3. `echo (decoct(10)."<br/>");// 12`
4. `echo (decoct(22)."<br/>");// 26`
5. `?>`

Output:

```
2
12
26
```

## PHP Math: base\_convert() function

The base\_convert() function allows you to convert any base number to any base number. For example, you can convert hexadecimal number to binary, hexadecimal to octal, binary to octal, octal to hexadecimal, binary to decimal etc.

### Syntax

1. string base\_convert ( string \$number , int \$frombase , int \$tobase )

### Example

1. <?php
2. \$n1=10;
3. echo (base\_convert(\$n1,10,2)."<br/>");// 1010
4. ?>

Output:

```
1010
```

## PHP Math: bindec() function

The bindec() function converts binary number into decimal.

### Syntax

1. number bindec ( string \$binary\_string )

### Example

1. <?php
2. echo (bindec(10)."<br/>");// 2
3. echo (bindec(1010)."<br/>");// 10
4. echo (bindec(1011)."<br/>");// 11
5. ?>

Output:

```
2
10
11
```

## PHP Form Handling

We can create and use forms in PHP. To get form data, we need to use PHP superglobals \$\_GET and \$\_POST.

The form request may be get or post. To retrieve data from get request, we need to use \$\_GET, for post request \$\_POST.

## PHP Get Form

Get request is the default form request. The data passed through get request is visible on the URL browser so it is not secured. You can send limited amount of data through get request.

Let's see a simple example to receive data from get request in PHP.

*File: form1.html*

1. <form action="welcome.php" method="get">
2. Name: <input type="text" name="name"/>
3. <input type="submit" value="visit"/>
4. </form>

*File: welcome.php*

1. <?php
2. \$name=\$\_GET["name");//receiving name field value in \$name variable
3. echo "Welcome, \$name";
4. ?>

## PHP Include File

PHP allows you to include file so that a page content can be reused many times. There are two ways to include file in PHP.

1. include
2. require

### Advantage

**Code Reusability:** By the help of include and require construct, we can reuse HTML code or PHP script in many PHP scripts.

---

## PHP include example

PHP include is used to include file on the basis of given path. You may use relative or absolute path of the file. Let's see a simple PHP include example.

*File: menu.html*

1. <a href="http://www.javatpoint.com">Home</a> |
2. <a href="http://www.javatpoint.com/php-tutorial">PHP</a> |



3. `<a href="http://www.javatpoint.com/java-tutorial">Java</a> |`
4. `<a href="http://www.javatpoint.com/html-tutorial">HTML</a>`  
*File: include1.php*

1. `<?php include("menu.html"); ?>`
2. `<h1>This is Main Page</h1>`

Output:

[Home](#) | [PHP](#) | [Java](#) | [HTML](#)

## This is Main Page

### PHP require example

PHP require is similar to include. Let's see a simple PHP require example.

*File: menu.html*

1. `<a href="http://www.javatpoint.com">Home</a> |`
2. `<a href="http://www.javatpoint.com/php-tutorial">PHP</a> |`
3. `<a href="http://www.javatpoint.com/java-tutorial">Java</a> |`
4. `<a href="http://www.javatpoint.com/html-tutorial">HTML</a>`  
*File: require1.php*

1. `<?php require("menu.html"); ?>`
2. `<h1>This is Main Page</h1>`

Output:

[Home](#) | [PHP](#) | [Java](#) | [HTML](#)

## This is Main Page

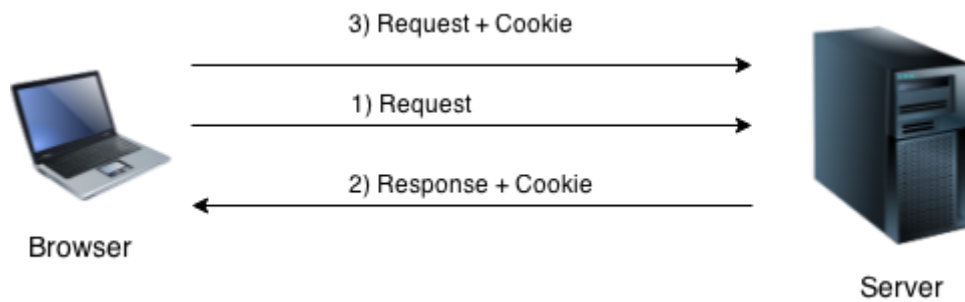
### PHP include vs PHP require

If file is missing or inclusion fails, **include** allows the *script to continue* but **require** *halts the script* producing a fatal E\_COMPILE\_ERROR level error.

## PHP Cookie

PHP cookie is a small piece of information which is stored at client browser. It is used to recognize the user.

Cookie is created at server side and saved to client browser. Each time when client sends request to the server, cookie is embedded with request. Such way, cookie can be received at the server side.



In short, cookie can be created, sent and received at server end.

**Note: PHP Cookie must be used before <html> tag.**

## PHP setcookie() function

PHP setcookie() function is used to set cookie with HTTP response. Once cookie is set, you can access it by \$\_COOKIE superglobal variable.

### Syntax

1. bool setcookie ( string \$name [, string \$value [, int \$expire = 0 [, string \$path
2. [, string \$domain [, bool \$secure = false [, bool \$httponly = false ]]]]] )

### Example

1. setcookie("CookieName", "CookieValue");/\* defining name and value only\*/
2. setcookie("CookieName", "CookieValue", time()+1\*60\*60);//using expiry in 1 hour(1\*60\*60 seconds or 3600 seconds)
3. setcookie("CookieName", "CookieValue", time()+1\*60\*60, "/mypath/", "mydomain.com", 1);

## PHP \$\_COOKIE

PHP \$\_COOKIE superglobal variable is used to get cookie.

### Example

1. \$value=\$\_COOKIE["CookieName"];//returns cookie value

## PHP Cookie Example

File: cookie1.php

1. <?php
2. setcookie("user", "Sonoo");
3. ?>

```

4. <html>
5. <body>
6. <?php
7. if(!isset($_COOKIE["user"])) {
8.     echo "Sorry, cookie is not found!";
9. } else {
10.    echo "<br/>Cookie Value: " . $_COOKIE["user"];
11.}
12. ?>
13. </body>
14. </html>

```

Output:

Sorry, cookie is not found!

Firstly cookie is not set. But, if you *refresh* the page, you will see cookie is set now.

Output:

Cookie Value: Sonoo

## PHP Delete Cookie

If you set the expiration date in past, cookie will be deleted.

*File: cookie1.php*

```

1. <?php
2. setcookie ("CookieName", "", time() -
   3600); // set the expiration date to one hour ago
3. ?>

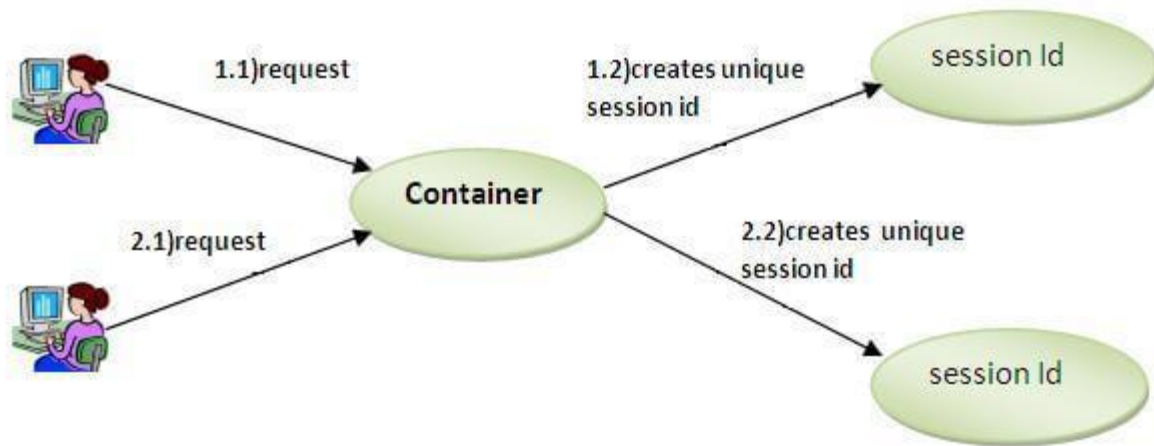
```

## PHP Session

PHP session is used to store and pass information from one page to another temporarily (until user close the website).

PHP session technique is widely used in shopping websites where we need to store and pass cart information e.g. username, product code, product name, product price etc from one page to another.

PHP session creates unique user id for each browser to recognize the user and avoid conflict between multiple browsers.



## PHP session\_start() function

PHP session\_start() function is used to start the session. It starts a new or resumes existing session. It returns existing session if session is created already. If session is not available, it creates and returns new session.

### Syntax

1. bool session\_start ( void )

### Example

1. session\_start();

## PHP \$\_SESSION

PHP \$\_SESSION is an associative array that contains all session variables. It is used to set and get session variable values.

### Example: Store information

1. \$\_SESSION["user"] = "Sachin";

### Example: Get information

1. echo \$\_SESSION["user"];

## PHP Session Example

*File: session1.php*

1. <?php
2. session\_start();

```
3. ?>
4. <html>
5. <body>
6. <?php
7. $_SESSION["user"] = "Sachin";
8. echo "Session information are set successfully.<br/>";
9. ?>
10. <a href="session2.php">Visit next page</a>
11. </body>
12. </html>
```

*File: session2.php*

```
1. <?php
2. session_start();
3. ?>
4. <html>
5. <body>
6. <?php
7. echo "User is: ".$_SESSION["user"];
8. ?>
9. </body>
10. </html>
```

## PHP Session Counter Example

*File: sessioncounter.php*

```
1. <?php
2. session_start();
3.
4. if (!isset($_SESSION['counter'])) {
5.     $_SESSION['counter'] = 1;
6. } else {
7.     $_SESSION['counter']++;
8. }
9. echo ("Page Views: ".$_SESSION['counter']);
10. ?>
```

## PHP Destroying Session

PHP `session_destroy()` function is used to destroy all session variables completely.

*File: session3.php*

```
1. <?php
2. session_start();
```

3. session\_destroy();
4. ?>

## PHP Write File

PHP fwrite() and fputs() functions are used to write data into file. To write data into file, you need to use w, r+, w+, x, x+, c or c+ mode.

### PHP Write File - fwrite()

The PHP fwrite() function is used to write content of the string into file.

#### Syntax

1. int fwrite ( resource \$handle , string \$string [, int \$length ] )

#### Example

1. <?php
2. \$fp = fopen('data.txt', 'w');//opens file in write-only mode
3. fwrite(\$fp, 'welcome ');
4. fwrite(\$fp, 'to php file write');
5. fclose(\$fp);
- 6.
7. echo "File written successfully";
8. ?>

Output: data.txt

```
welcome to php file write
```

## PHP Overwriting File

If you run the above code again, it will erase the previous data of the file and writes the new data. Let's see the code that writes only new data into data.txt file.

1. <?php
2. \$fp = fopen('data.txt', 'w');//opens file in write-only mode
3. fwrite(\$fp, 'hello');
4. fclose(\$fp);
- 5.
6. echo "File written successfully";
7. ?>

Output: data.txt

```
hello
```

## PHP Append to File

You can append data into file by using a or a+ mode in fopen() function. Let's see a simple example that appends data into data.txt file.

Let's see the data of file first.

data.txt

```
welcome to php file write
```

## PHP Append to File - fwrite()

The PHP fwrite() function is used to write and append data into file.

### Example

1. <?php
2. \$fp = fopen('data.txt', 'a');//opens file in append mode
3. fwrite(\$fp, ' this is additional text ');
4. fwrite(\$fp, 'appending data');
5. fclose(\$fp);
- 6.
7. echo "File appended successfully";
8. ?>

Output: data.txt

```
welcome to php file write this is additional text appending data
```

## PHP Delete File

In PHP, we can delete any file using unlink() function. The unlink() function accepts one argument only: file name. It is similar to UNIX C unlink() function.

PHP unlink() generates E\_WARNING level error if file is not deleted. It returns TRUE if file is deleted successfully otherwise FALSE.

### Syntax

1. bool unlink ( string \$filename [, resource \$context ] )

**\$filename** represents the name of the file to be deleted.

## PHP Delete File Example

```
1. <?php
2. $status=unlink('data.txt');
3. if($status){
4. echo "File deleted successfully";
5. }else{
6. echo "Sorry!";
7. }
8. ?>
```

Output

```
File deleted successfully
```

## PHP Read File

PHP provides various functions to read data from file. There are different functions that allow you to read all file data, read data line by line and read data character by character.

The available PHP file read functions are given below.

- fread()
- fgets()
- fgetc()

### PHP Read File - fread()

The PHP fread() function is used to read data of the file. It requires two arguments: file resource and file size.

### Syntax

```
1. string fread (resource $handle , int $length )
```

**\$handle** represents file pointer that is created by fopen() function.

**\$length** represents length of byte to be read.

### Example

```
1. <?php
2. $filename = "c:\\file1.txt";
3. $fp = fopen($filename, "r");//open file in read mode
4.
5. $contents = fread($fp, filesize($filename));//read file
```



- 6.
7. `echo "<pre>$contents</pre>";`*//printing data of file*
8. `fclose($fp);`*//close file*
9. `?>`

Output

```
this is first line
this is another line
this is third line
```

## PHP Read File - fgets()

The PHP fgets() function is used to read single line from the file.

### Syntax

1. `string fgets ( resource $handle [, int $length ] )`

### Example

1. `<?php`
2. `$fp = fopen("c:\\file1.txt", "r");`*//open file in read mode*
3. `echo fgets($fp);`
4. `fclose($fp);`
5. `?>`

Output

```
this is first line
```

## PHP Read File - fgetc()

The PHP fgetc() function is used to read single character from the file. To get all data using fgetc() function, use !feof() function inside the while loop.

### Syntax

1. `string fgetc ( resource $handle )`

### Example

1. `<?php`
2. `$fp = fopen("c:\\file1.txt", "r");`*//open file in read mode*
3. `while(!feof($fp)) {`
4.  `echo fgetc($fp);`

```
5. }  
6. fclose($fp);  
7. ?>
```

Output

```
this is first line this is another line this is third line
```

## PHP File Upload Example

*File: uploadform.html*

```
1. <form action="upload.php" method="post" enctype="multipart/form-data">  
2.   Select File:  
3.   <input type="file" name="fname"/>  
4.   <input type="submit" value="Upload Image" name="submit"/>  
5. </form>
```

*File: upload.php*

```
1. <?php  
2. $path = "e:/";  
3. $path = $path.basename( $_FILES['fname']['name']);  
4.  
5. if(move_uploaded_file($_FILES['fname']['tmp_name'], $path)) {  
6.     echo "<img src=\".$path.\" height=200 width=300 />";  
7.     echo "File uploaded successfully!";  
8. } else{  
9.     echo "Sorry, file not uploaded, please try again!";  
10. }  
11. ?>
```