Cookies and Sessions are used to store information. Cookies are only stored on the client-side machine, while sessions get stored on the server.

**Session**

A session creates a file in a temporary directory on the server where registered session variables and their values are stored. This data will be available to all pages on the site during that visit.

A session ends when the user closes the browser or after leaving the site, the server will terminate the session after a predetermined period of time, commonly 30 minutes duration.

**Cookies:**

**Why to use Cookie ?:** http is a stateless protocol which doesn’t keep any information(i.e. regarding user identification means the user who request to the server). Cookies allow us to track the state of the application using small files stored on the user’s computer.

Steps: 1) A user requests for a page (e.g. [www.facebook.com](http://www.facebook.com)) through web browser (i.e through http) to the server.

2) The server responds the request by sending the page (if available on the server) through the web browser and sets the cookie on the user’s computer (i.e. in the web browser).

3) When next time browser sends any request (e.g. [www.facebook.com](http://www.facebook.com)) to web server then it sends those cookies information to the server and server uses that information to identify the user**.**

A cookie is often used to identify a user. Cookies are text files stored on the client computer and they are kept of use tracking purpose. A cookie is a small file with the maximum size of 4KB that the web server stores on the client computer. Server script sends a set of cookies to the browser. The browser stores this information on a local machine for future use.

**Create Cookies With PHP**

A cookie is created with the **setcookie()** function. The php setcookie() function must be executed before the HTML opening tag.

### Syntax

### setcookie(name, value, expire, path, domain, secure, httponly);

Only the name parameter is required. All other parameters are optional.

**“setcookie”** is the PHP function used to create the cookie.

**“name”** is the name of the cookie that the server will use when retrieving its value from the $\_COOKIE($\_COOKIE is a PHP built in super global variable) array variable. It’s mandatory.

**“value”** is the value of the cookie and its mandatory.

**“expire”** or (expiry\_time) is optional; it can be used to set the expiry time for the cookie such as 1 hour. The time is set using the PHP time() functions plus or minus a number of seconds greater than 0 i.e. time() + 3600 for 1 hour.

**“path”** or (cookie\_path) is optional; it can be used to set the cookie path on the server. The forward slash “/” means that the cookie will be made available on the entire domain. Sub directories limit the cookie access to the subdomain.

**“domain”** is optional, it can be used to define the cookie access hierarchy.

**“secure”** is optional, the default is false. It is used to determine whether the cookie is sent via https if it is set to true or http if it is set to false.

**“httponly”** is optional. If it is set to true, then only client side scripting languages i.e.[JavaScript](https://www.guru99.com/interactive-javascript-tutorials.html)cannot access them.

**Create and retrieve cookie:**

Q. Create a cookie named "abc" with the value "bca". The cookie will expire after 30 days (86400 \* 30). The "/" means that the cookie is available in entire website (otherwise, the directory path may be set).

We can retrieve the value of the cookie "user" (using **$\_COOKIE** which is a PHP built in super global variable ). To check whether coockie variable is set or not, we can use the isset() function to find out if the cookie is set.

The number of values that the **$\_COOKIE** array can contain depends on the memory size set in php.ini.

The default value is 1GB.

### <?php $cookie\_name = "abc"; $cookie\_value = "bca"; setcookie($cookie\_name, $cookie\_value, time() + (86400 \* 30), "/"); // 86400 = 1 day ?> <html> <body> <?php if(!isset($\_COOKIE[$cookie\_name]))

### {   echo "Cookie named '" . $cookie\_name . "' is not set!"; }

### else

### {   echo "Cookie '" . $cookie\_name . "' is set!<br>";   echo "Value is: " . $\_COOKIE[$cookie\_name]; } ?> </body> </html>

**Check if Cookies are Enabled:**

To check whether cookies are enabled or not, first, try to create a cookie with the **setcookie()** function, then count the **$\_COOKIE** array variable.

<?php

$cookie\_name = "abc";

$cookie\_value = "bca";

setcookie($cookie\_name, $cookie\_value, time() + (86400 \* 30), "/");

?>

<html>

<body>

<?php

if(count($\_COOKIE) > 0)

{

echo "Cookies are enabled.";

}

else

{

echo "Cookies are disabled.";

}

?>

</body>

</html>

**Delete a Cookie**

To delete a cookie, use the **setcookie()**function with an expiration date in the past. If you want to destroy a cookie before its expiry time, then you set the expiry time to a time that has already passed.

### <?php // set the expiration date to one hour ago

$cookie\_name = "abc";

$cookie\_value = "bca";

setcookie($cookie\_name, $cookie\_value, time() -3600);

### ?> <html> <body> <?php echo "Cookie 'user' is deleted."; ?> </body> </html>