



PHP Programming - Sessions and Cookies

Dr. Jenila Livingston L.M.

SCOPE

WHY PHP – Sessions ?

allows you to

- store and display information about a user,
- determine which user groups a person belongs to,
- utilize permissions on your [website](#) or you just want to do something cool on your site,

PHP Sessions store the information on the web server in a location that you chose in special files. These files are connected to the user's web browser via the server and a special ID called a "Session ID".

This is nearly 99% flawless in operation and it is virtually invisible to the user.

How session works

- PHP first creates a unique identifier for that particular session which is a random string of **32 hexadecimal numbers** such as 3c7foj34c3jj973hjkop2fc937e3443.
- A cookie called **PHPSESSID** is automatically sent to the user's computer to store unique session identification string.
- A file is automatically created on the server in the designated temporary directory and bears the name of the unique identifier prefixed by sess_ ie sess_3c7foj34c3jj973hjkop2fc937e3443.

session-start()

- A PHP session is easily started by making a call to the **session_start()** function
- This function first checks if a session is already started and if none is started then it starts a new one.
- `session_start();`

Session Variables, Counter and isset

- Session variables are stored in associative array called **\$_SESSION[]**. These variables can be accessed during lifetime of a session.
- **counter** that is incremented each time the page is visited during the session.
- **isset()** function to check if session variable is already set or not.

Session - Example

```
<?php
session_start();
if( isset( $_SESSION['counter'] ) ) {
    $_SESSION['counter'] += 1;
}
else {
    $_SESSION['counter'] = 1;
}
$msg = "You have visited this page ". $_SESSION['counter'];
$msg .= "in this session.";
echo ( $msg );
?>
```

Create session variables

```
<?php
```

```
// start a session
```

```
session_start();
```

```
// initialize session variables
```

```
$_SESSION['logged_in_user_id'] = '1';
```

```
$_SESSION['logged_in_user_name'] = 'Leni';
```

```
// access session variables
```

```
echo $_SESSION['logged_in_user_id'];
```

```
echo $_SESSION['logged_in_user_name'];
```

```
// unset a session variable
```

```
unset($_SESSION['logged_in_user_id']);
```

```
?>
```

Fetch from database

Step1: Store user name, userid in a database table

```
$_SESSION[" username "] = "Leni";  
$_SESSION[" userid"] = 1
```

Step2: Fetch from table

```
$row = mysql_fetch_array($login_record);  
$_SESSION["userid"] = $row["id"];
```


Session variables:

Set Favourite color and animal

- ```
<?php
// Start the session
session_start();
?>
<!DOCTYPE html>
<html><body>
<?php
// Set session variables
$_SESSION["favcolor"] = "green";
$_SESSION["favanimal"] = "cat";
echo "Session variables are set.";
?></body> </html>
```

# Display Favourite color and animal

- ```
<?php
session_start();
?>
<!DOCTYPE html>
<html><body>
<?php
// Echo session variables that were set on previous page
echo "Favorite color is " . $_SESSION["favcolor"]
. "<br>";
echo "Favorite animal is " . $_SESSION["favanimal"] . ".";
?>
</body></html>
```

Unset and Destroy a Session

- A PHP session can be destroyed by **session_destroy()** function.
- to destroy a single session variable then you can use **unset()** function to unset a session variable.
- `unset($_SESSION['counter']);`

Destroy PHP - Sessions

```
<?php
// start the session
session_start();
header("Cache-control: private"); //IE 6 Fix
$_SESSION = array();
session_destroy();
echo "<strong>Step 5 - Destroy This Session </strong><br />";
if($_SESSION['name']){
    echo "The session is still active";
} else {
    echo "Ok, the session is no longer active! <br />";
    echo "<a href=\"page1.php\"><< Go Back Step 1</a>";
}
?>
```

COOKIES

- Cookies are **text files** stored on the client computer and they are kept of use tracking purpose.

COOKIES – returning users

- There are three steps involved in identifying returning users.
- Server script sends a set of cookies to the browser. For example name, age, or identification number etc.
- **Browser stores this information on local machine** for future use.
- When next time browser sends any request to web server then it sends those cookies information to the server and server uses that information to identify the user.

Use of cookies

- **To store username/password** information so that the user doesn't have to log in every time they visit the website ("**remember me**" sign ins).
- To keep track of a user's progress during a specified process.
- To remember a user's theme.

SETCOOKIE

- PHP provided **setcookie()** function to set a cookie.
- This function requires upto six arguments and should be called before `<html>` tag
- **setcookie(name, value, expire, path, domain, security);**
- **Name** – This sets the name of the cookie and is stored in an environment variable called **HTTP_COOKIE_VARS** or **\$_COOKIE**. This variable is used while accessing cookies.

Setting a Cookie

- **Value** – This sets the value of the named variable and is the content that you actually want to store.
- **Expiry** – This specifies a future time in seconds since 00:00:00 GMT on 1st Jan 1970. After this time cookie will become inaccessible. If this parameter is not set then cookie will automatically expire when the Web Browser is closed.
- **Path** – This specifies the directories for which the cookie is valid. **A single forward slash character** permits the cookie to be valid for all directories.

Setting a Cookie

- **Domain** – This can be used to specify the domain name. All cookies are only valid for the host and domain which created them.
Eg: `www.example.com` **or** `example.com`
- **Security** – This can be **set to 1** to specify that the cookie should only be sent by secure transmission using HTTPS otherwise set to 0 which mean cookie can be sent by regular HTTP.

Example Cookie

```
<?php  
setcookie("name", "jenila", time()+3600, "/", "", 0);  
?>  
  
<html> <head>  
  
<title>Setting Cookies with PHP</title> </head>  
  
<body> <?php echo "Set Cookies"?>  
  
</body>  
  
</html>
```

Getting a Cookie

```
<html> <head> <title>Accessing Cookies with  
    PHP</title> </head>
```

```
<body>
```

```
<?php
```

```
echo $_COOKIE["name"]. "<br />";
```

```
/* is equivalent to */
```

```
echo $HTTP_COOKIE_VARS["name"]. "<br />";
```

```
?>
```

```
</body> </html>
```

Check whether cookie is set

```
<html> <head> <title>Accessing Cookies with  
    PHP </title> </head> <body>  
<?php  
if( isset($_COOKIE["name"]))  
echo "Welcome " . $_COOKIE["name"] . "<br  
    />";  
else  
echo "Sorry... Not recognized" . "<br />";  
?> </body> </html>
```

Delete a cookie – set the time as expired time for that cookie

```
<?php
```

```
setcookie( "name", "", time()- 60, "/", "", 0);
```

```
?>
```

```
<html>
```

```
<head> <title>Deleting Cookies with PHP</title>
```

```
</head>
```

```
<body> <?php echo "Deleted Cookies" ?>
```

```
</body>
```

```
</html>
```

Thank you!