Hidaya Institute Of Science & Technology

بسم الله الرحمن الرحيم

Session In Php

By: IQRA BALOCH.

Contents

- What is Session?
- Why, Where Session is Used?
- How Sessions Works?
- Session As File.
- Session Expiry

What Is Session?

- Generally session means limited period of time.

Or

- A meeting or period devoted to a particular activity.

- To get understood about sessions in php we need to go back and review our concepts regarding http, html, client and server.
- It was clearly observed that web is stateless.
- There is no continuous connection between client and server.

- For-example :
- The server does not know that the current page was requested in past by the user or not.
- A browser request a page from server – connection opens between both, server get the request and sends back the response – connection closed.

What is Session (Continued)

- Session:
- Is one that makes the web satefull. Simply maintaining the state.
- Allows to track information of user.
- A session is basically a way of storing variables and making them available across multiple pages on your web site.
- Session is unlike other variables in the sense that we are not passing them individually to each new page, but instead retrieving them from the session we open at beginning of each page.

What is Session (Continued)

- In web applications, a session is the sequence of interactions between the server and a user.
- A **session** refers to a period of activity when a PHP script stores state information on a Web server.
- All session data is stored on the server (usually in a database, to which the client does not have direct access) linked to that identifier.
- A PHP session provides an easy, but effective, way of passing variables between web pages in a PHP application. The session starts when a PHP site is accessed and ends when the user closes their web browser.

Why, Where Session is used (Need)?

- Previously we store the information in a hidden form field. Even then, it persists only if the form is submitted.
- To make data accessible across the various pages of an entire website.
- Suppose a user is buying book online, there should be a mechanism to keep track of books added by user to basket.
- Very common example be the login system, where there is need of state maintenance to check whether user is logged in or not- is the user valid.

How Session Work

- Session Is a file stored on Server.
- To find which file belongs to which user a cookie is created.
- A special session cookie is set on their browser and then we look that cookie to find the place where a file resides on our server and then we can look in that file to find information in session.
- The information is not there for them to see or to edit they can only change id which reference to file.
- They can change id number is very long and complicated to help to ensure we do not end up with hijacking of other users session information.

How Session Works (Continued)

- To use session we use to create a file and set cookie on user machine. Then we need to find the cookie and the corresponding file on the machine this be done with session_start().
- The session_start() function starts a new session or continues an existing one
- The session_start() function generates a unique session ID to identify the session
- A **session ID** is a random alphanumeric string that looks something like:

  7f39d7dd020773f115d753c71290e11f
- The session_start() function creates a text file on the Web server that is the same name as the session ID, preceded by **sess_**

How Session Works (Continued)

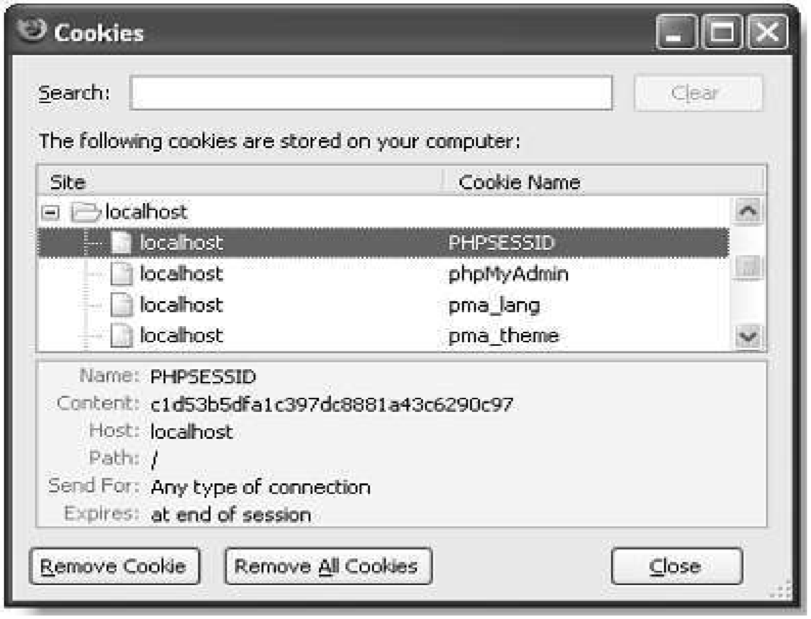
- <?php
- session_start();
- ?>
- This command should be called only once in each page, and it must be called before the PHP script generates any output, so the ideal position is immediately after the opening PHP tag.
- If any output is generated before the call to session_start(), the command fails and the session won't be activated for that page.

How Session Works (Continued)

- When a session is initiated, the server stores information in session variables that can be accessed by other pages as long as the session remains active (normally until the browser is closed).
- Because the identifier is unique to each visitor, the information stored in session variables cannot be seen by anyone else.

Session As File.

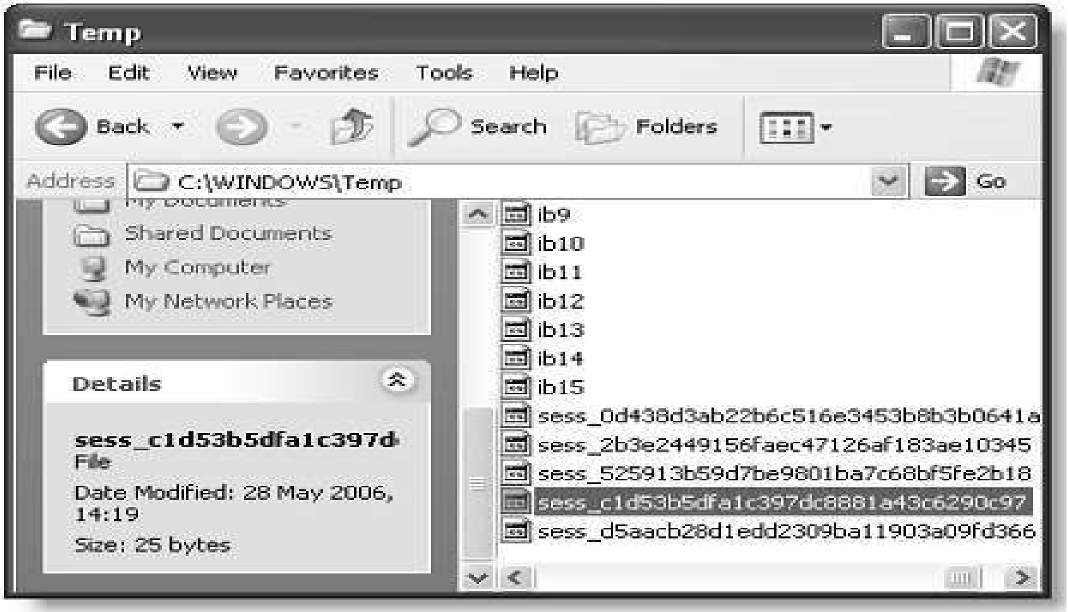
- The cookie stored in the browser is called PHPSESSID, unique identifier.
- A matching file, which contains the same unique identifier as part of its filename, is created on the web server.
- PHP sessions store a unique identifier as a cookie in the browser (left) and on the server(right).



The following cookies are stored on your computer:

Site	Cookie Name
localhost	
localhost	PHPSESSID
localhost	phpMyAdmin
localhost	pma_lang
localhost	pma_theme

Name: PHPSESSID
Content: c1d53b5dfa1c397dc8881a43c6290c97
Host: localhost
Path: /
Send For: Any type of connection
Expires: at end of session



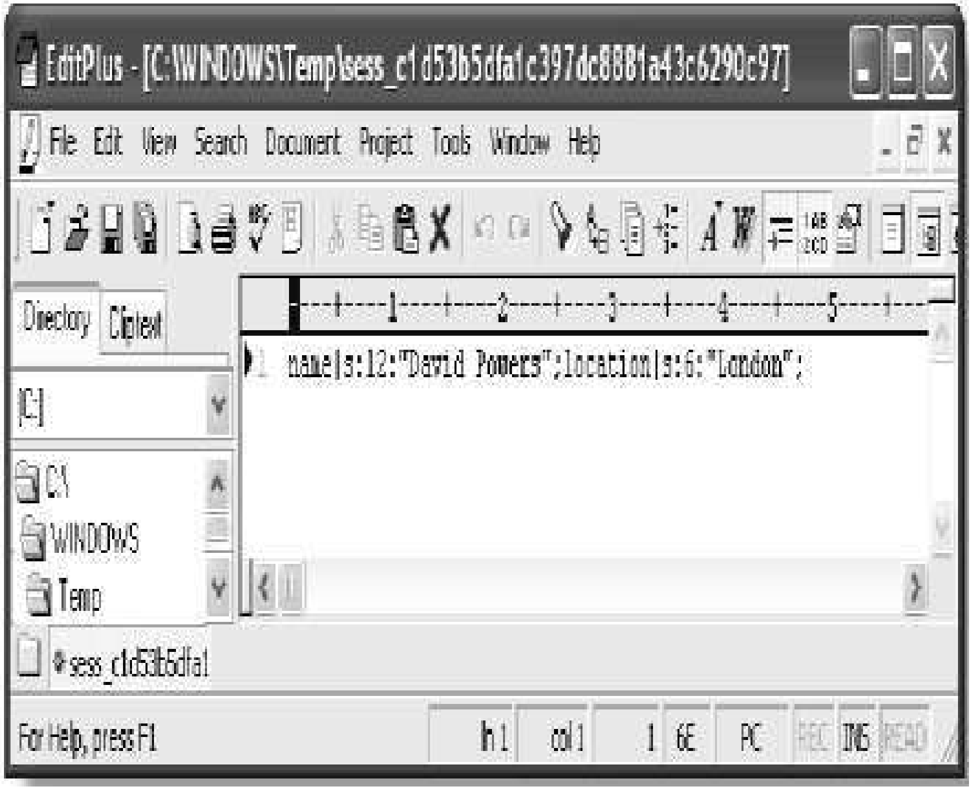
Address C:\WINDOWS\Temp

ib9
ib10
ib11
ib12
ib13
ib14
ib15
sess_0d438d3ab22b6c516e3453b8b3b0641a
sess_2b3e2449156faec47126af183ae10345
sess_525913b59d7be9801ba7c68bf5fe2b18
sess_c1d53b5dfa1c397dc8881a43c6290c97
sess_d5aacb28d1edd2309ba11903a09fd366

sess_c1d53b5dfa1c397d
File
Date Modified: 28 May 2006, 14:19
Size: 25 bytes

Session As File(Continued).

- There is no danger of private information being exposed through someone examining the contents of a cookie on a shared computer.
- The session variables and their values are stored on the web server.
- Figure Shows contents of a simple session file.
- As you can see, it's in plain text, and the content isn't difficult to decipher.
- The session shown in the figure has two variables: name and location.
- The variable names are followed by a vertical pipe, then the letter "s", a colon, a number, another colon, and the variable's value in quotes. The "s" stands for string, and the number indicates how many characters the string contains.



EditPlus - [C:\WINDOWS\Temp\sess_c1d53b5dfa1c397dc8881a43c6290c97]
File Edit View Search Document Project Tools Window Help
Directory Cliptext
[C:]
C:\
WINDOWS
Temp
sess_c1d53b5dfa1
For Help, press F1
ln 1 col 1 1 6E PC REC INS READ

```
name|s:12:"David Powers";location|s:6:"London";
```

Storing & Retrieving Session Data

- Php provides us the $_SESSION[] – Environment variable.
- We can use $_SESSION[“varibaleName”] to store and retrieve data as we do in our normal practice.
- Example:

```
<?php
    session_start();
    $_SESSION[“varibaleName”]=“THIS IS DATA TO BE STORED”;
    echo  $_SESSION[“varibaleName”];
?>
```

Using Session Data/Variables

- Having established a session, you can now create, store and retrieve information pertaining to that session.
- Information to be stored for a session is kept in session variables.
- This is where you both store and retrieve session data.
- These variables are all stored in a single array $_SESSION.
-
```
<?php
session_start(); // this starts the session
 $_SESSION['color']='red'; // this sets variables in the session
$_SESSION['size']='small';
 $_SESSION['shape']='round';
 ?>
```

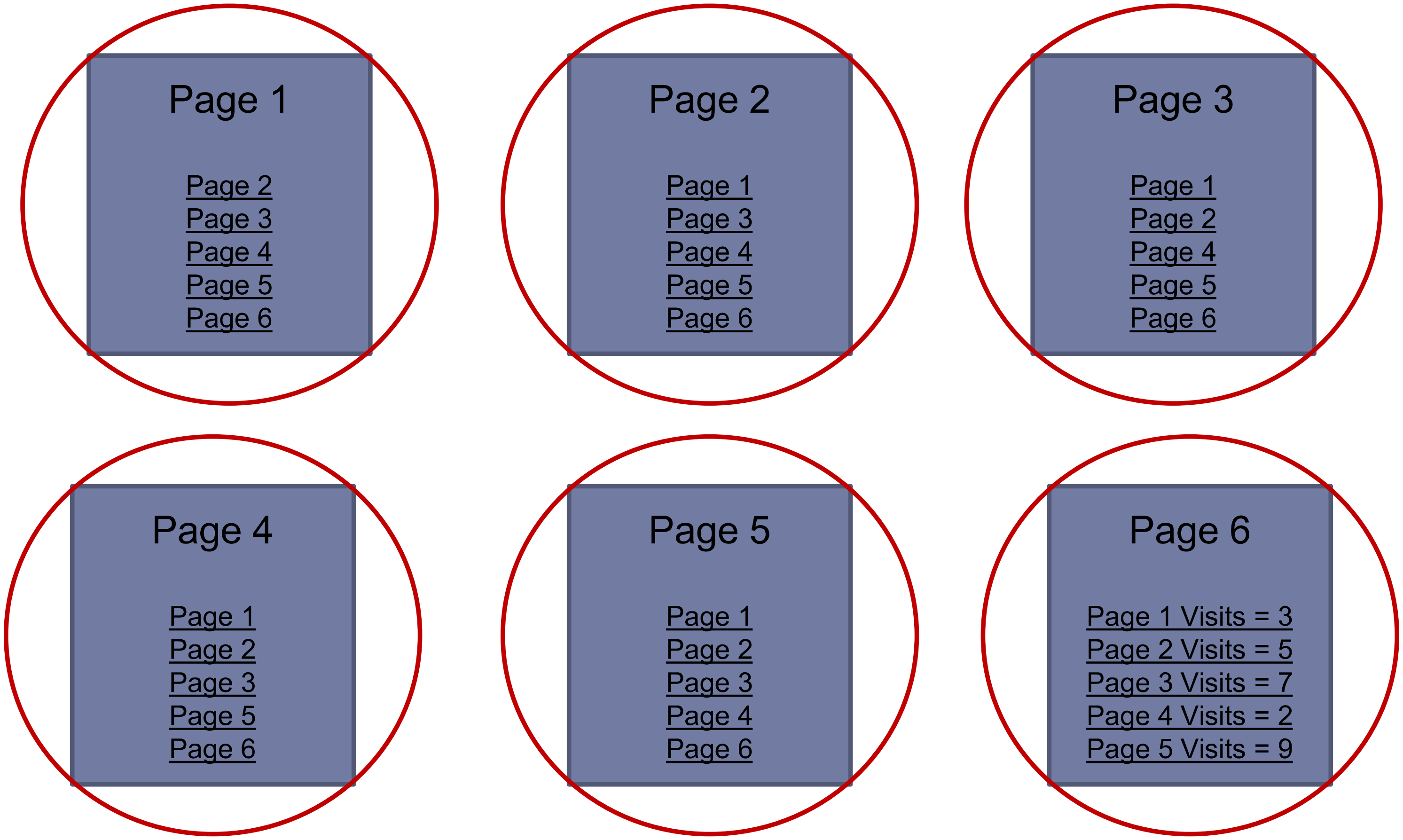
Retrieving Session Data

- The PHP session's $_SESSION variables can be used in exactly the same way as other PHP variables.
- To retrieve a session variable you simply need to echo that variable.
- If you need to access that variable on any other page you need to start session first.

-
  ```
  <?php
  session_start(); // this starts the session
   echo $_SESSION['color']; // this is session variable.
  echo $_SESSION['size'];
   echo $_SESSION['shape'];
   ?>
  ```

Assignment 1:

Make six web pages, every page contain the links of all other pages, on the sixth page the result of all visited pages should be shown that how many times the user has visited each page.



Page 1

Page 2
Page 3
Page 4
Page 5
Page 6

Page 2

Page 1
Page 3
Page 4
Page 5
Page 6

Page 3

Page 1
Page 2
Page 4
Page 5
Page 6

Page 4

Page 1
Page 2
Page 3
Page 5
Page 6

Page 5

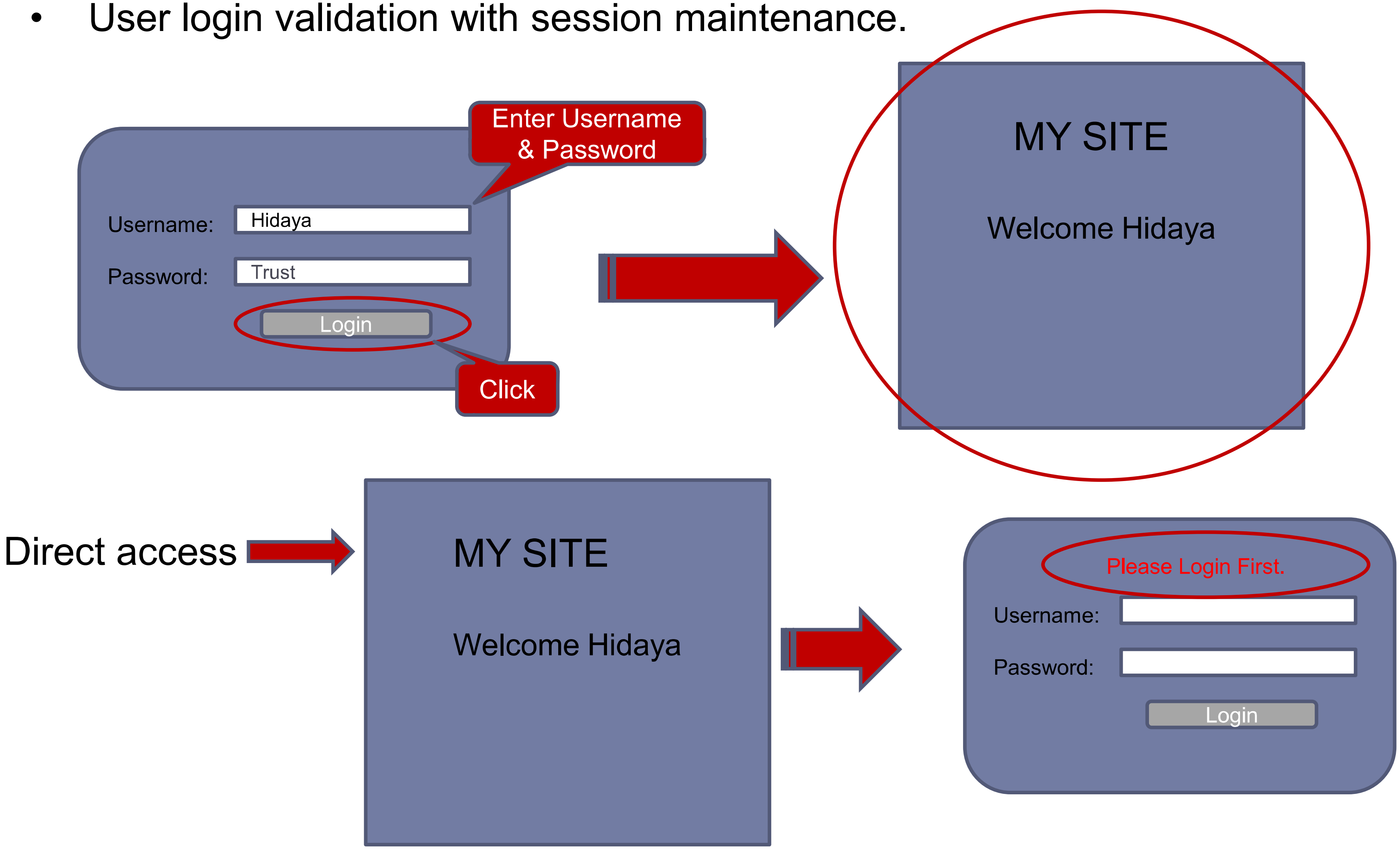
Page 1
Page 2
Page 3
Page 4
Page 6

Page 6

Page 1 Visits = 3
Page 2 Visits = 5
Page 3 Visits = 7
Page 4 Visits = 2
Page 5 Visits = 9

Assignment 2:

- User login validation with session maintenance.



Enter Username & Password

Username: Hidaya

Password: Trust

Login

Click

MY SITE

Welcome Hidaya

Direct access

MY SITE

Welcome Hidaya

Please Login First.

Username:

Password:

Login

Session Expiry:

- By default, PHP sessions expire:
  - After a certain length of inactivity (default 1440s/24 min), the PHP garbage collection processes deletes session variables.
  - To check session expiry time stored in php.ini file we use:
    - ➢ini_get(“session.gc_maxlifetime”);

Unset():

Description:

- **unset()** destroys the specified variables.

Return Values

- No value is returned.

Note :

- If *$_SESSION* is used, use unset() to unregister a session variable, i.e. *unset ($_SESSION['varname']);*.

Caution:

- Do NOT unset the whole *$_SESSION* with *unset($_SESSION)* as this will disable the registering of session variables through the *$_SESSION* superglobal.

session_destroy:

Description:

- Destroys all data registered to a session.
- **session_destroy()** destroys all of the data associated with the current session.
- It does not unset any of the global variables associated with the session, or unset the session cookie.

Return Values

- Returns **TRUE** on success or **FALSE** on failure.

session_id:

Description:

- session_id() Get and/or set the current session id.

Parameters:

- ***id***
- If *id* is specified, it will replace the current session id.
- **session_id()** needs to be called **before** session_start() for that purpose.
- Depending on the session handler, not all characters are allowed within the session id. For example, the file session handler only allows characters in the range *a-z A-Z 0-9 , (comma) and - (minus)*!

Return Values

- **session_id()** returns the session id for the current session or the empty string (*""*) if there is no current session (no current session id exists).

session_id (Contd:)

Example :

```
<?php
   /* set the session id to hidaya */
   session_id(“hidaya");
   session_start();
   echo “My session id is session_id<br />";
   ?>
```

session_name:

Description:

- Get and/or set the current session name.

Parameters:

- ***name***
- The session name references the name of the session, which is used in cookies. It should contain only alphanumeric characters; it should be short and descriptive. If *name* is specified, the name of the current session is changed to its value.

Warning The session name can't consist of digits only, at least one letter must be present. Otherwise a new session id is generated every time.

Return Values

- Returns the name of the current session.

session_name (Contd:)

Example :

```
<?php
    /* set the session name to  MyWebsite  */
    session_name(“MyWebsite");
    session_start();
    echo “My session name is session_name<br />";
?>
```

session_save_path:

Description:

- Get and/or set the current session save path.
- **session_save_path()** returns the path of the current directory used to save session data.

Parameters:

- ***Path***
- Session data path.
- If specified, the path to which data is saved will be changed.
- **session_save_path()** needs to be called **before** session_start() for that purpose.

Return Values:

- Returns the path of the current directory used for data storage.

session_save_path:(Contd:)

Example :

```
<?php
    session_save_path("C:\users");
    session_start();
    echo “My session path is  session_save_path <br />";
?>
```