Session Management in PHP

Session 21

For Aptech Center Use Only

Version 1.0 © Aptech Limited.

Objectives

- *Define a session*
- *Explain the procedure to work with a session*
- *Describe the methods to start a session*
- *Explain the method to register a session*
- *Describe the method to end a session*
- *Explain the use of the php.ini file*

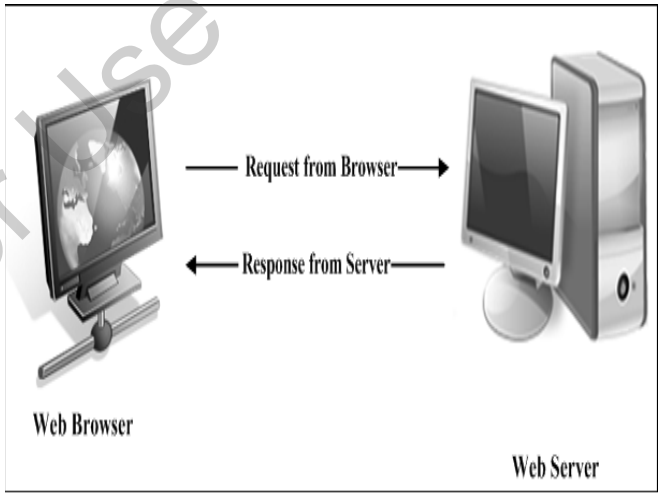
For Aptech Center Use Only

- Sessions are similar to cookies and enable the functionality of storing temporary user information

- The difference between cookies and sessions is as follows:
  - Pertinent cookies store information on the local computer
  - Session enables PHP to store information on the Web server

For Aptech Center Use Only

- Web browsers and Web servers have a stateless interaction and do not maintain track of user sessions
- HTTP protocol
  - Enables Web browsers to communicate with Web servers
  - Has no methods or functions to maintain the status of a particular user



For Aptech Center Use Only

- Web sites that cannot depend on HTTP or Web servers for complex user interaction need session tracking
- Refers to the total time the user accesses information on a particular Web site before exiting the Web site
- Manages data for a particular user in a specific session
- Enable distinguishing user specific information for the entire duration of the session

For Aptech Center Use Only

Importance of a Session

- Consider an example, in a particular Web site, the user has to first register and then log on to access any information
- For such authentication procedures, the state of the user has to be maintained across the Web site
- Web sites traditionally use `GET` and `POST` methods to pass user information from one script to another
- When these methods are used, PHP assigns user information variables in the following format:

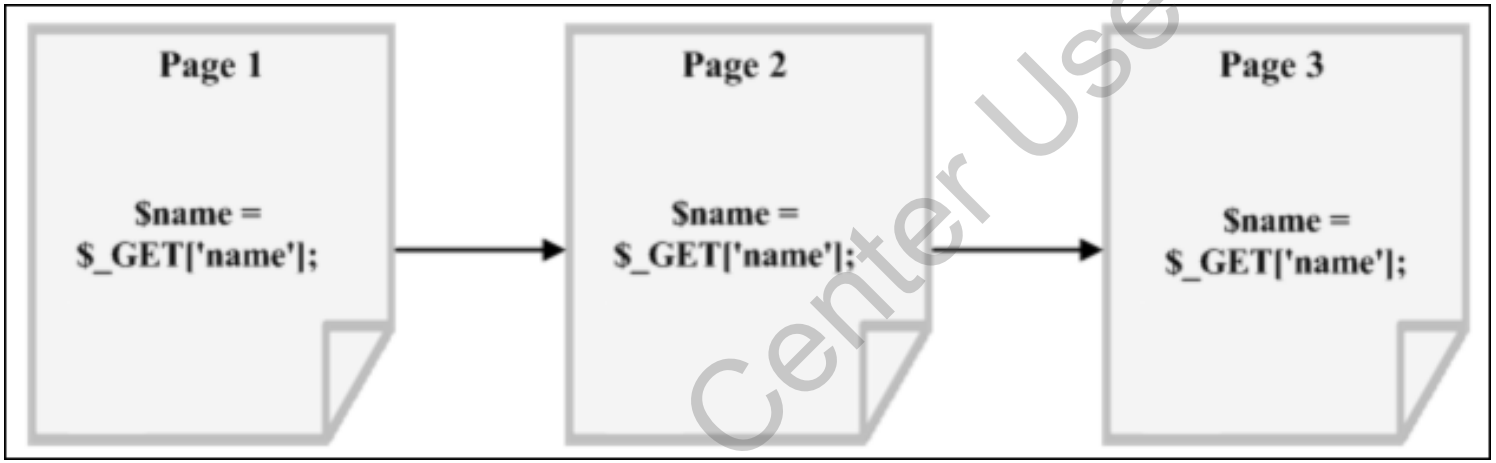
```
$name = $_GET['name'];
```

- In the code, the variable `$name` stores the value that the script retrieves from the HTML form
- This process of transferring user information is time consuming and not essential for large Web sites

For Aptech Center Use Only

Importance of a Session

- The code to be used across all the Web pages of the Web site is as follows:

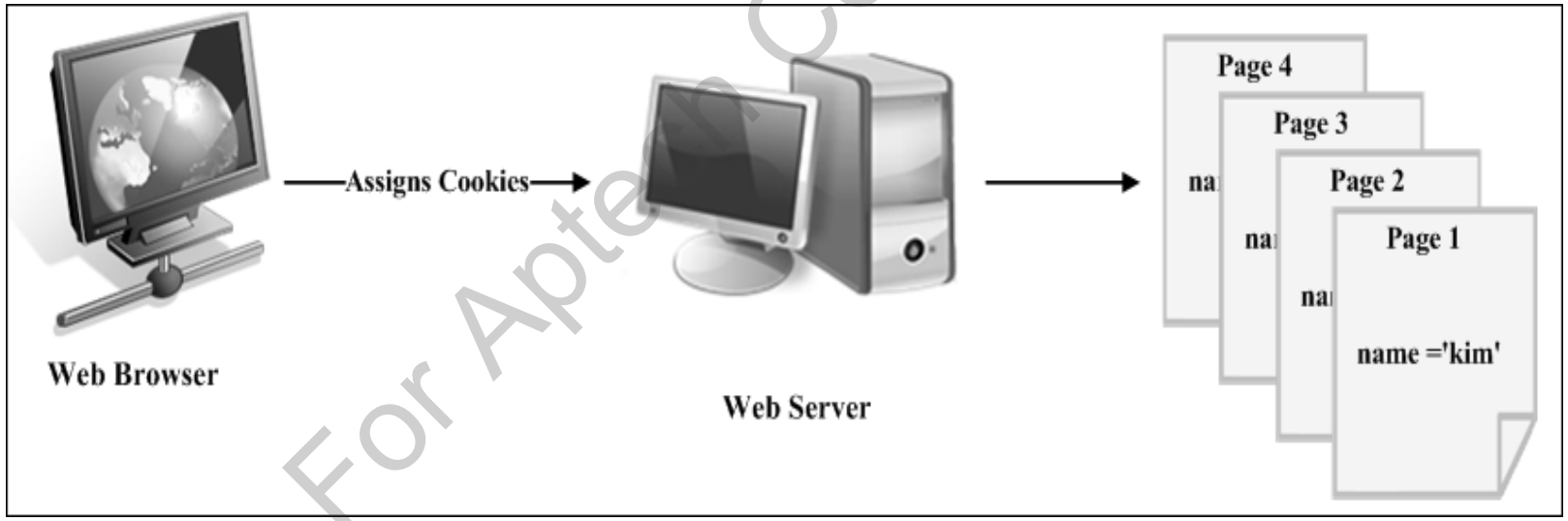


For Aptech Center Use Only

Importance of a Session

- Consider a scenario, where it is required to store and retrieve information for 20 or more users across 10 different pages
- Due to this disadvantage of using the `GET` and `POST` methods, Web developers prefer using cookies

Figure displays the assignment of cookies by the Web server to the browser



For Aptech Center Use Only

Importance of a Session

- Cookies enable to store data into a variable and access it across all the pages of the Web site
- Cookies are prone to security risks because the user information is saved at the client-end
- The risks involved are greater when users access Web sites from a public computer or a shared computer

For Aptech Center Use Only

- **Size of the cookie:**
  - The amount of information stored in the cookie determines the size of the cookie
  - The size of the cookie determines the size of the Web page and increase in file size of the Web page results in poor performance
- **Cookies disabled:**
  - Web sites store cookies on the hard disk of the client as a result the performance of computers reduces
  - To improve the performance of such computers, users disable cookies making the assignment of cookies pointless

For Aptech Center Use Only

Importance of a Session

- Sessions play an important role in such situations
- Sessions eliminate deletion and assignment of new cookies to the same user
- The size of a cookie does not affect the performance of a Web site
- Both the Web server and Web browser benefit because statistical information in the server database is accurate

For Aptech Center Use Only

Table explains the difference between cookies and sessions

Cookies	Sessions
Stores user information on the client system (Web Browser)	Stores user information on the Web server
Available even after the user exits the Web browser	Destroyed when the user exits the Web browser
Users can disable cookies	Users cannot disable sessions
Have size limits	Do not have size limits

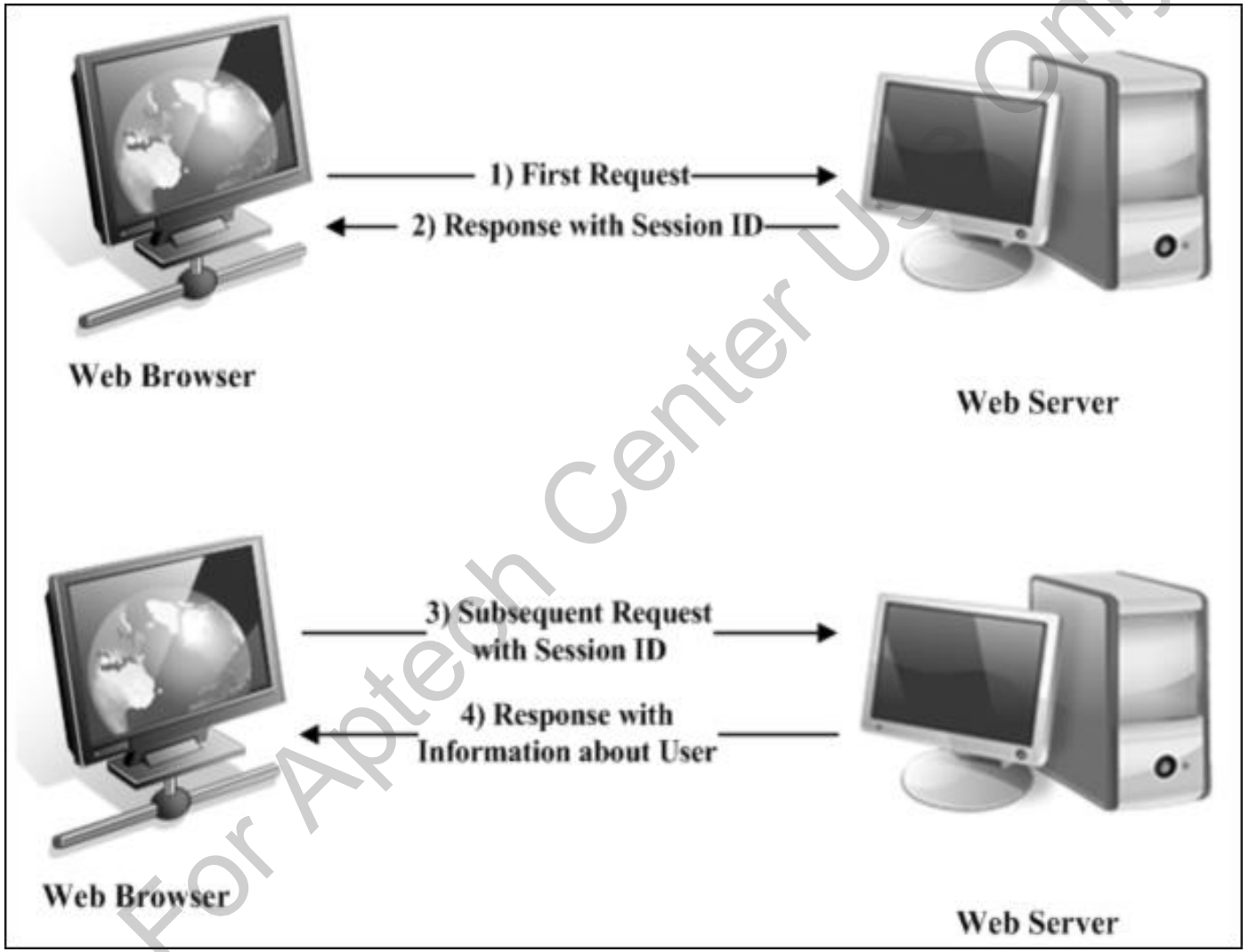
For Aptech Center Use Only

- Session commences when a user accesses a session-enabled Web site
- Web server assigns a unique session ID to each user when the user starts a session
- The scripts store and access user information through the session ID depending on the following two situations:
  - **Cookies enabled:**
    - Web server allots a session ID to the Web browser through a cookie, using the `setcookie()` function
    - Cookies enable transfer of user information between the browser and the server
    - PHP stores session IDs in cookies

For Aptech Center Use Only

- **Cookies disabled:**
  - Web server allots a session ID to the browser using the Uniform Resource Locator (URL)
  - The URL transfers user information from the browser to the server
  - PHP stores the session variables in a file and names the file based on the session ID
- While using a session, PHP stores all the user information in a file on the Web server
- The file includes a session ID that is related the user's session variable
- Each session ID identifies a different user and relates to a file that belongs to that user
- PHP destroys the session file once the user exits the Web site

For Aptech Center Use Only



For Aptech Center Use Only

- PHP works with sessions in the following sequence:
  - User accesses a session-enabled Web site which checks the user identity
  - If the user is a new visitor, the Web site allocates a unique session ID to the user. The Web site saves a cookie containing the session ID on the Web browser
  - The Web browser records the cookie that holds the session ID. The browser uses the same cookie to retrieve the session ID and record all the session-related information
  - The session file is destroyed from the Web server, when the user exists from the Web site

For Aptech Center Use Only

Life cycle of a Session

- There are three stages in the life cycle of a session based on the communication between the Web browser and the Web server
- They are as follows:
  - Starting the session
  - Registering the session variable
  - Ending the session

For Aptech Center Use Only

Starting the Session

- A session starts when a user logs on to the Web site
- The `session_start()` function enables to start a session
- The process of starting a session is also called as initializing a session
- The session file is created in the `/tmp` directory
- PHP assigns a name to this file based on the unique session identifier value generated by the PHP engine
- The session identifier is also known as the session ID
- The session ID is a hexadecimal string of 32 digits

For Aptech Center Use Only

Starting the Session

- The file naming convention for the session file is as follows:

  `sess_<32_digit_hexadecimal_value>`

- The session file name is always preceded by `sess_` and is followed by a random 32 digit hexadecimal value
- The Web server passes the session ID as a response to the browser
- The response sets up a session cookie in the browser with the name `PHPSESSID` and the value of the identifier
- The `session_start()` function must be specified on the top of every Web page or before the start of the actual coding
- If the session is valid and existing, it activates the frozen variables of the session
- If the session is invalid or non existing, it creates a session ID for the new session

For Aptech Center Use Only

Starting the Session

- The `session_start()` function is as follows:

Syntax

```
session_start();
```

- To start a session, perform the following steps:

1. Open a new file in **gedit** text editor
2. Enter the following code:

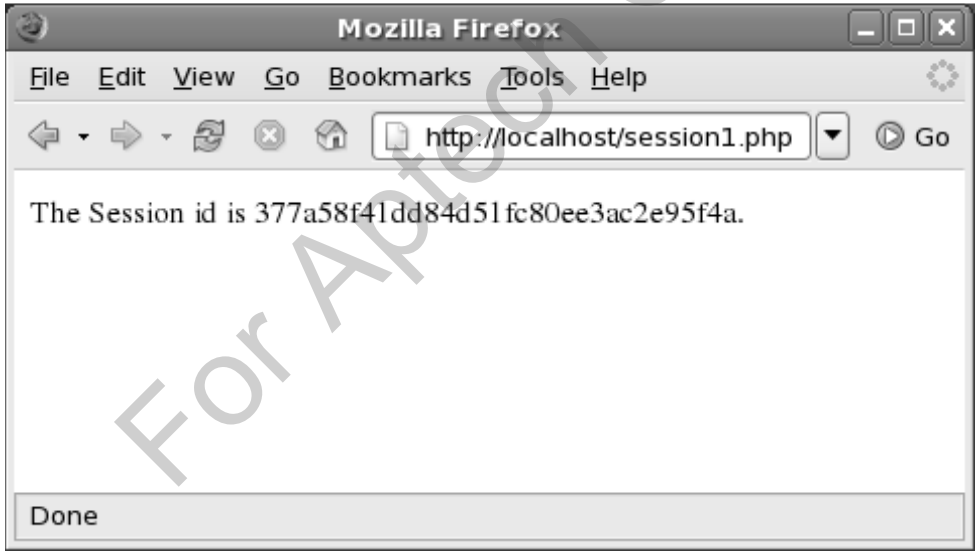
Snippet

```
<?php
// initializing a session
session_start();
/* The session_id() function displays the session id
that PHP allots to
a user. */
echo "The Session id is " .session_id(). ".<br>";
?>
```

For Aptech Center Use Only

3. Save the script as session1.php in the `/usr/local/apache2/htdocs/` directory.
4. Open the Mozilla Firefox Web browser.
5. Enter `http://localhost/session1.php` in the Address bar and press Enter.

The following output is displayed:



For Aptech Center Use Only

Registering the Session Variable

- Variables in a session file contain user specific information
- Session library enables creation, serialization, and storage of session data
- There are three methods to set a session variable, which are as follows:
  - `$_SESSION[]` - recommended for PHP 4.1.0
  - `$HTTP_SESSION_VARS[]` - recommended for PHP 4.0.6 or less
  - `session_register()` - not recommended as it has been deprecated
- Session variables can be of any data type such as integer, string, Boolean, or object
- PHP stores the session variables in a session file by serializing the values
- PHP automatically handles the process of serializing the session variables

For Aptech Center Use Only

Registering the Session Variable

- Steps to register the value of a session variable are as follows:

1. Open a new file in **gedit** text editor
2. Enter the following code:

Snippet

```
<?php
session_start();
$_SESSION['myname']= "Jessica";
?>
<HTML>
<HEAD> <TITLE> Session </TITLE></HEAD>
<BODY>
<A HREF = "mypage.php"> Homepage of MyPage.com </A>
</BODY>
</HTML>
```

For Aptech Center Use Only

3. Save the file as **sessionstart.php** in the `/usr/local/apache2/htdocs/`directory

- To display the value of the session variable, perform the following steps:
  1. Open a new file in **gedit** text editor.
  2. Enter the following code:

Snippet

```
<?php
session_start();
$myname = $_SESSION['myname'];
?>
<HTML>
<HEAD> <TITLE> Homepage </TITLE></HEAD>
<BODY>
Welcome <?php echo $myname ?> to MyPage.com <br>
</BODY>
</HTML>
```

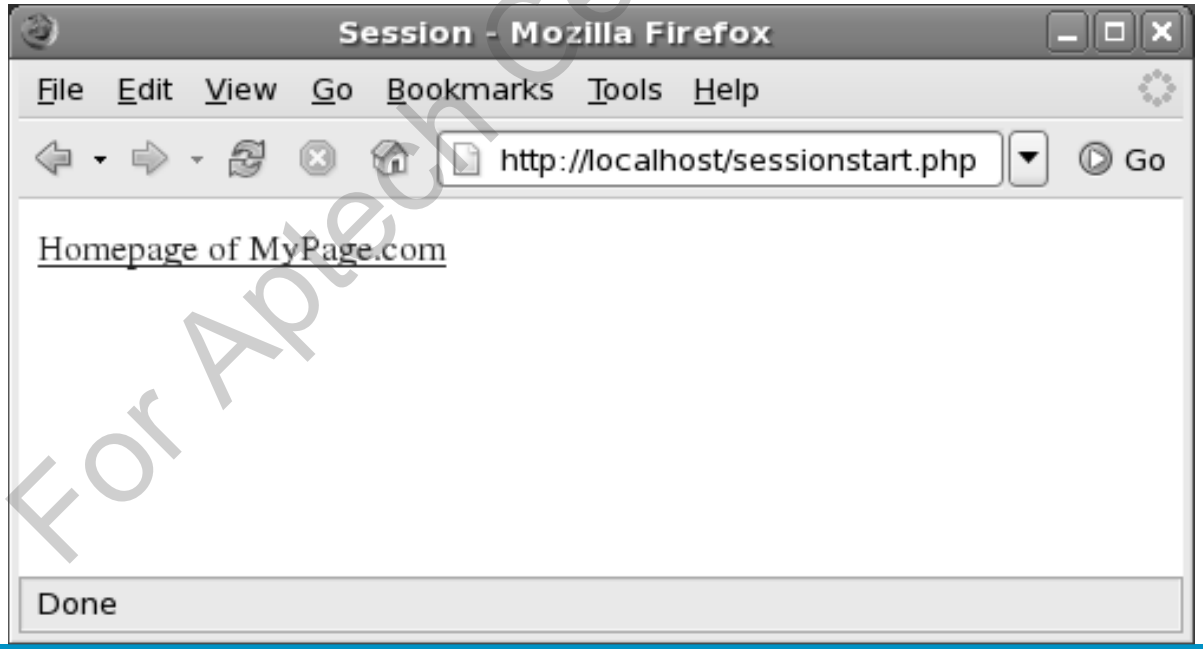
For Aptech Center Use Only

3. Save the file as `mypage.php` in the `/usr/local/apache2/htdocs/directory`

4. Open the Mozilla Firefox Web browser

5. Enter http : `//localhost/sessionstart.php` in the Address bar and press **Enter**

The following output is displayed:

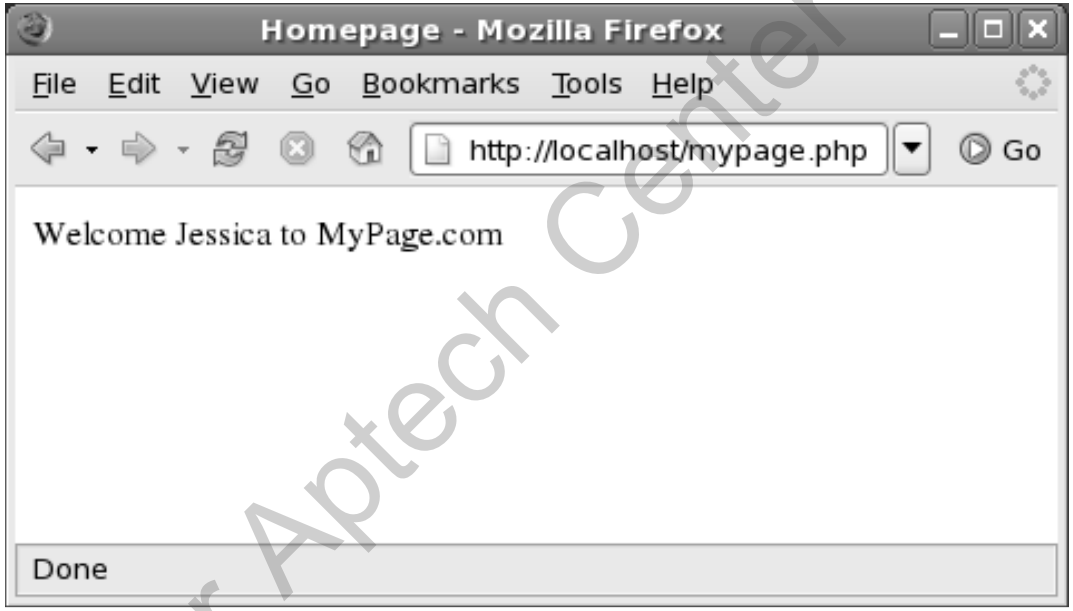


For Aptech Center Use Only

6. Click Homepage of `MyPage.com`

The Homepage page appears with the message,

```
Welcome Jessica to MyPage.com
```



For Aptech Center Use Only

Ending the Session

- When the user logs out of the Web site, the PHP script executes the `session_destroy()` function
- When the session file is deleted, the `$PHPSESID` cookie is not removed from the Web browser
- The syntax for the `session_destroy()` function is as follows:

Syntax

```
session_destroy();
```

For Aptech Center Use Only

Ending the Session

- PHP uses the following configuration directives when a session ends:
  - `gc_maxlifetime()`:
    - Enables PHP to determine the time to wait before ending a session and the process of cleaning up is called as garbage collection
  - `gc_probability()`:
    - Enables PHP to determine with what probability the garbage collection routine must be invoked

For Aptech Center Use Only

Ending the Session

- To destroy a session, perform the following steps:

1. Open a new file in **gedit** text editor
2. Enter the following code:

Snippet

```
<?php

session_start();

$myname = $_SESSION['myname'];

// The session_unset() function unregisters a session
variable.

session_unset();

session_destroy();

echo "Session destroyed!";
```

For Aptech Center Use Only

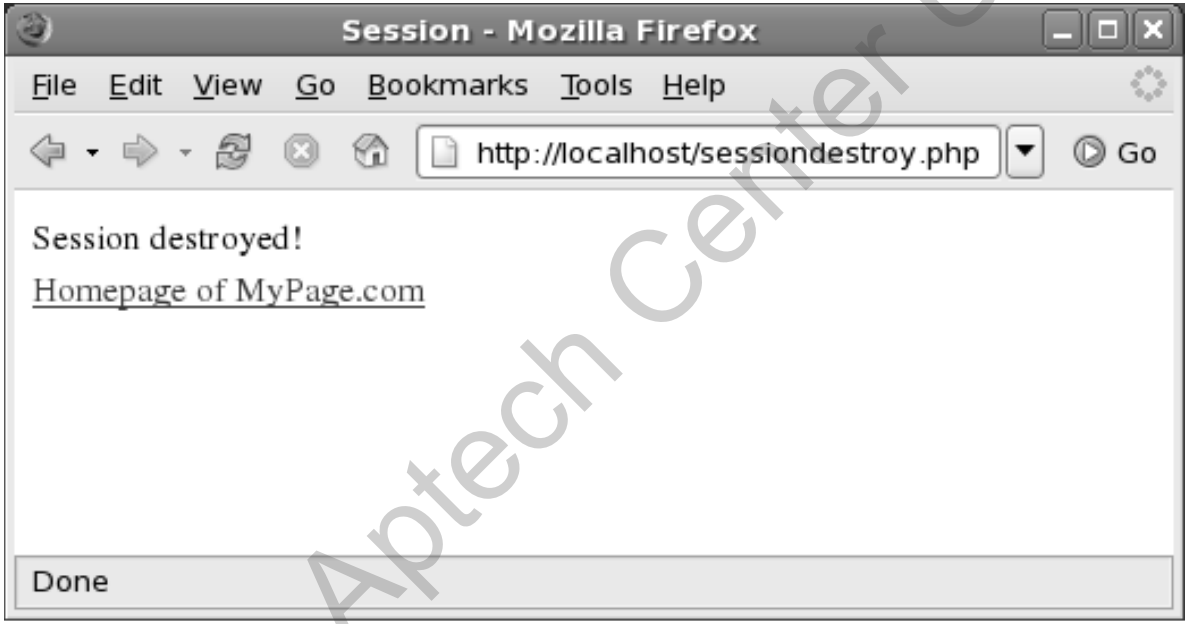
```
?>
<HTML>
<HEAD> <TITLE> Session </TITLE></HEAD>
<BODY>
<br>
<A HREF = "mypage.php"> Homepage of MyPage.com </A>
</BODY>
</HTML>
```

3. Save the file as **`sessiondestroy.php`** in the `/usr/local/apache2/htdocs/directory`
4. Open the Mozilla Firefox Web browser

For Aptech Center Use Only

5. Enter `http://localhost/sessiondestroy.php` in the Address bar and press **Enter**

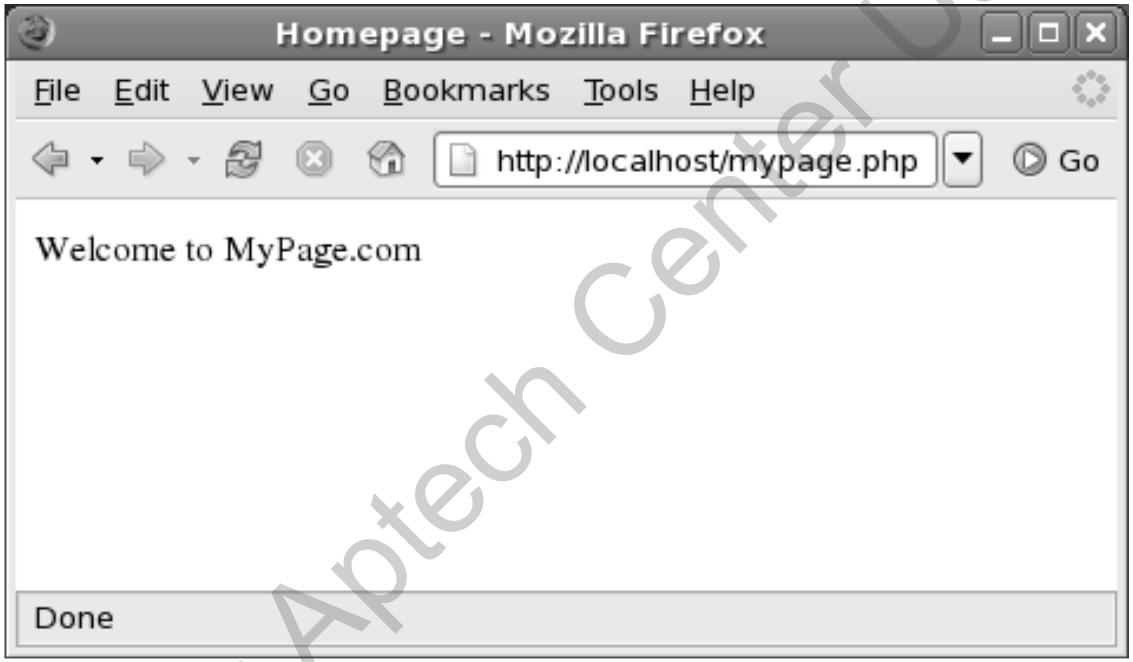
   The following output is displayed:



For Aptech Center Use Only

6. To view the execution of the `session_destroy()` function, click the Homepage of `MyPage.com` hyperlink

The following output is displayed:



The name **Jessica** does not appear in the message because the variable has been cleared.

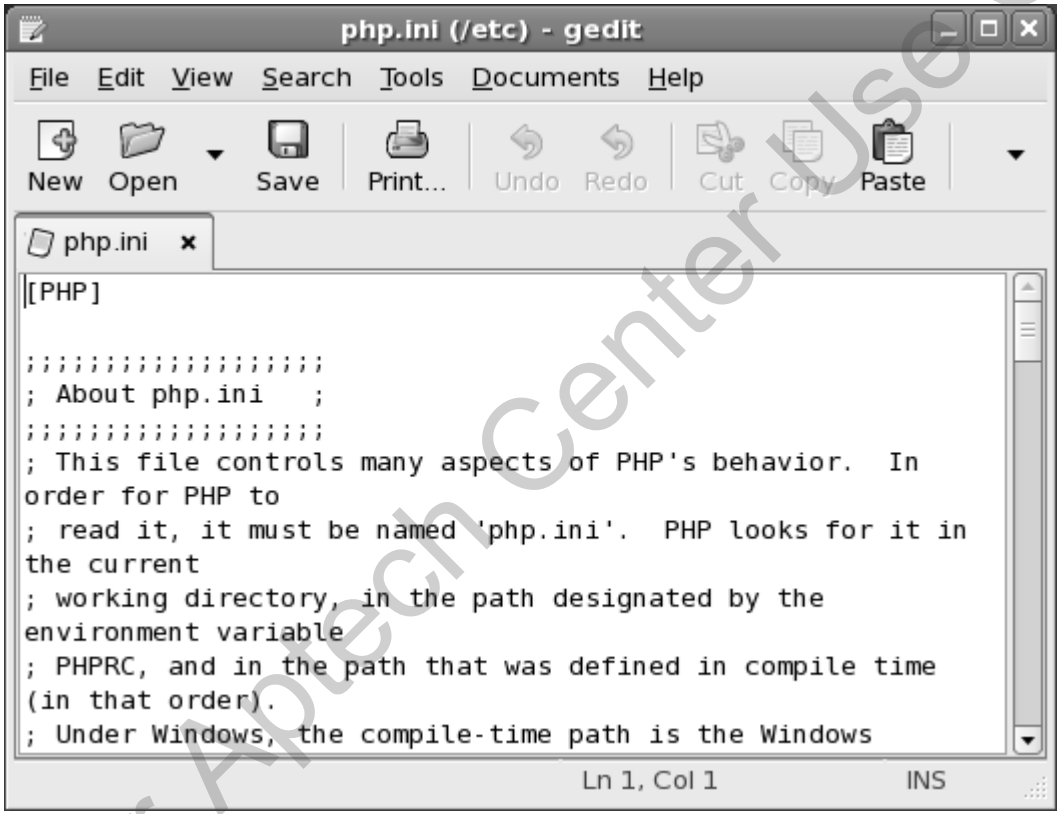
For Aptech Center Use Only

Working with the php.ini File

- A Web server can contain multiple `php.ini` files
- Create a `php.ini` file if it does not exist on the Web server
- The complete source code must be downloaded to create a new `php.ini` file
- PHP interpreter works according to the instructions included in the `php.ini` file
- The Web server searches sequentially for the `php.ini` file in the following locations:
  - Directory where the PHP script was called
  - Root of the Web directory
  - Directory containing the `default.ini` file on the Web server

For Aptech Center Use Only

- Figure displays the PHP configuration file



The **`php.ini`** file contains directive listed in the directive = value format.

You can use a semicolon to add a comment to the file.

For Aptech Center Use Only

Table lists the categories in the `php.ini` file

Categories	Options
Language Options	Enable PHP scripting language engine under Apache
	Allow ASP style tags
	Enforce year 2000 compliance
Safe Mode	Perform a UID compare check when opening files
	Allow executables under specific directories to be executed via exec family
	Allow user set environment variables that begin with PHP prefix
Font Colors	Indicate the colors that PHP uses for highlighting syntax
Misc	Indicate whether or not PHP discloses the fact that it is installed on the server
Resource Limits	Indicate the maximum time for script execution
	Indicate the maximum time for parsing request data
	Indicate the maximum amount of memory a script requires
Data Handling	Control list of separators used in PHP generated URLs to separate arguments
	Describe the order in which PHP registers Get, Post, Cookie, Environment and built-in variables
Path and Directories	Specifies the name of the directory under which PHP opens the script
	Specifies the name of the directory under which the loadable extensions exist

For Aptech Center Use Only

Categories	Options
Error handling and logging	Report all errors and warnings
	Report fatal compile time errors
	Report fatal run-time errors
	Report non-fatal error
	Report fatal errors that occur during initial startup of PHP
	Report user generated errors, warnings, and messages
Magic Quotes	Set magic quotes for incoming Get, Post, Cookie data
	Use Sybase style magic quotes
	Automatically adds file before or after any PHP document
File Uploads	Indicate whether or not to allow HTTP file uploads
	Indicate temporary directory for HTTP uploaded files
	Indicate the maximum allowed size for upload files
Session	Store and retrieve data
	Indicate whether or not cookies should be used
	Initializes session on request startup
	Serializes data

For Aptech Center Use Only

- Session category of the `php.ini` file include options, which are as follows:
  - `session.save_handler` - specifies how PHP stores and retrieves session variables
  - Either of the following values can be used for this option:
    - `files`: indicates the use of the session files
    - `mm`: stores and retrieves data from a shared memory
    - `user`: stores and retrieves variables with custom defined handlers

For Aptech Center Use Only

- `session.save_path` - specifies the name of the directory where the session files will be stored
- `session.use_cookies` - indicates whether PHP must send a session ID to the Web browser through a cookie. The value to enable a cookie to store a session ID is 1
- `session.use_only_cookies` - indicates whether the modules can use only cookies for storing session IDs. By default, this option is disabled
- `session.cookie_lifetime` - specifies the lifetime of the cookie and the value is specified in seconds

For Aptech Center Use Only

- `session.name` - manages the cookie name and form attributes such as `GET` and `POST` that holds the session ID. By default, the value of this option is `PHPSESSID`
- `session.auto_start` - enables sessions to automatically initialize if the session ID is not found in the browser request
- `session.cookie_secure` - specifies whether the cookies must be sent over secured connections. By default, the cookies are not sent through secured connections

For Aptech Center Use Only

- Other settings can also be modified in the `php.ini` file, such as:
  - `register_globals` - controls the functioning of server, forms, and environment variables

    If this option is disabled, variables can be retrieved using the `GET` or `POST` methods as follows:

```
$_POST['$variable_name'];
$_GET['$variable_name'];
```

    If `register_globals` is enabled, the variable can be directly accessed using the variable name as follows:

```
$storeValue = $variable_name;
```

    `$variable_name` is the name of the variable that contains the session data of another Web page

    The variable `$storeValue` stores the information included in the variable `variable_name`

For Aptech Center Use Only

- `upload_tmp_dir` - sets the location of the temporary file that is uploaded with the HTML form
- `display_errors and display_startup_errors` - enables PHP to display errors on the Web browser
- `log_errors and error_log` - enables PHP to display error logs

For Aptech Center Use Only

Summary

- Cookies provide the functionality of storing temporary user information on the local computer
- Sessions enable PHP to store user information on the Web server
- HTTP is considered as a stateless protocol that enables Web browsers to communicate with the Web servers
- Session refers to the total time a user accesses information on a particular Web site before exiting the Web site
- A PHP script can access the session ID when the Web user enables or disables cookies
- The three stages in the life cycle of a session are: starting a session, registering a session variable, and ending a session

For Aptech Center Use Only