

# Architecting Web Applications using PHP

## Session 8

### Cookies and Sessions Management in PHP

# Session Overview

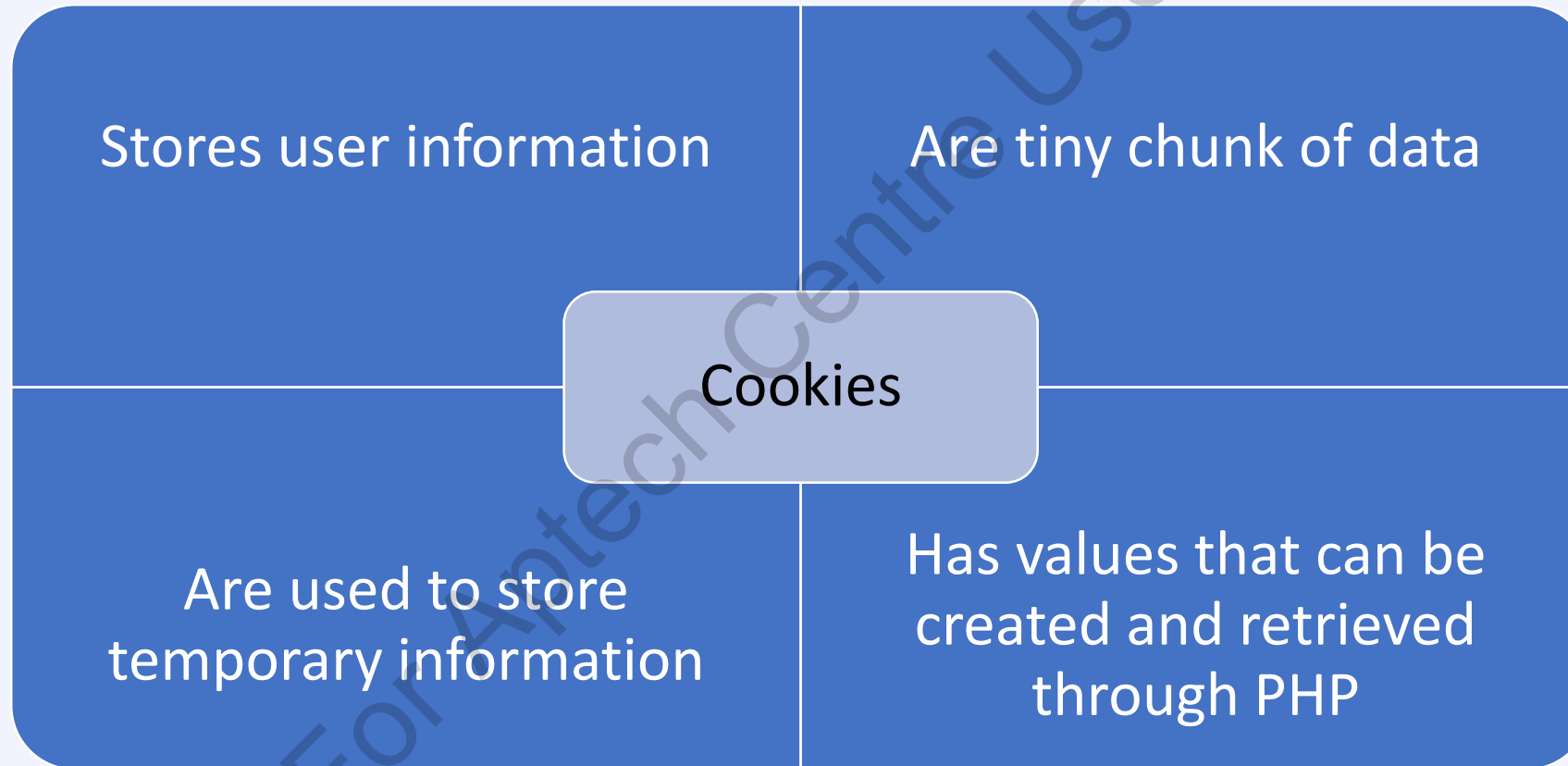
---

In this session, you will be able to:

- Define PHP cookies
- Explain the functions of cookies
- Explain setting of cookies with PHP through several functions
- Identify how to work with cookies
- Elaborate on PHP session management

For Aptech Centre Use Only

# Introduction to Cookies



# Types of Cookies [1-2]

Persistent	Non-persistent
Exist in the Web browser for a period specified at the time of its creation	Deleted from the Web browser as soon as the user exist the browser

For Aptech Centre Use Only

# Types of Cookies [2-2]

The Web server script sends across a set of cookies to the Web browser

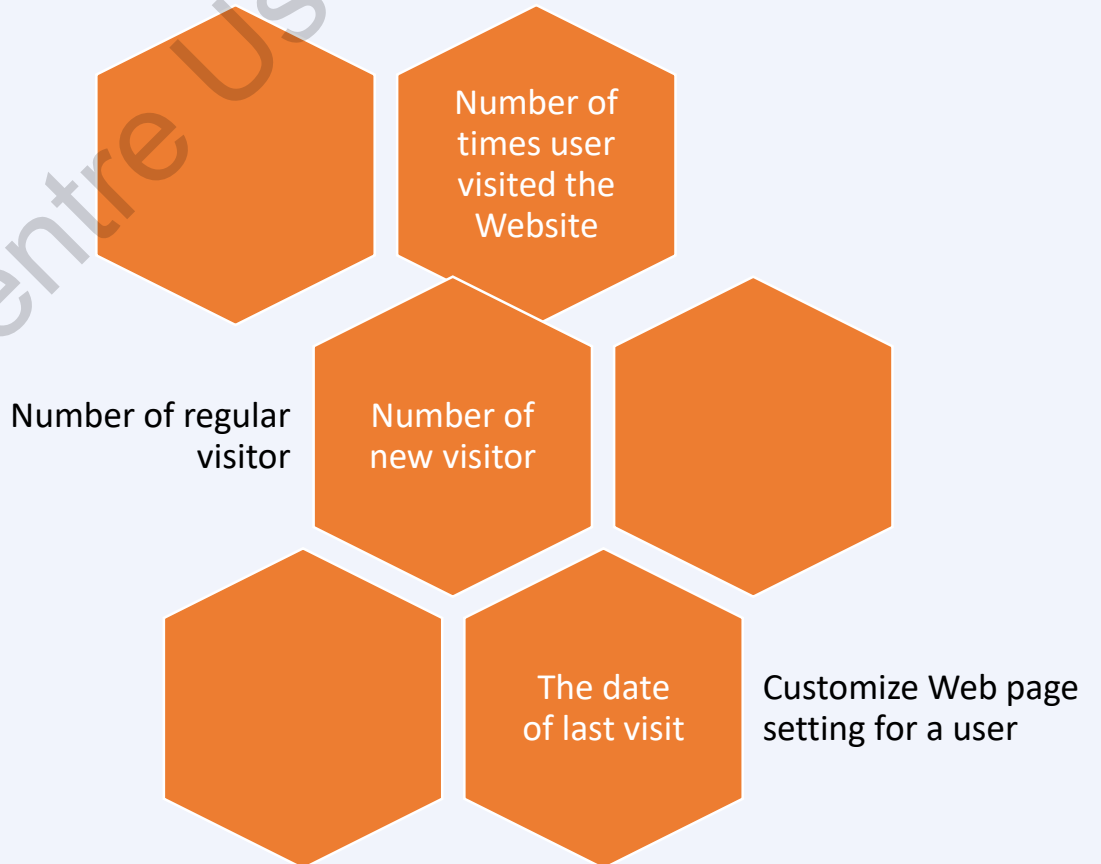


The Web browser saves the data that is received on the user's local machine



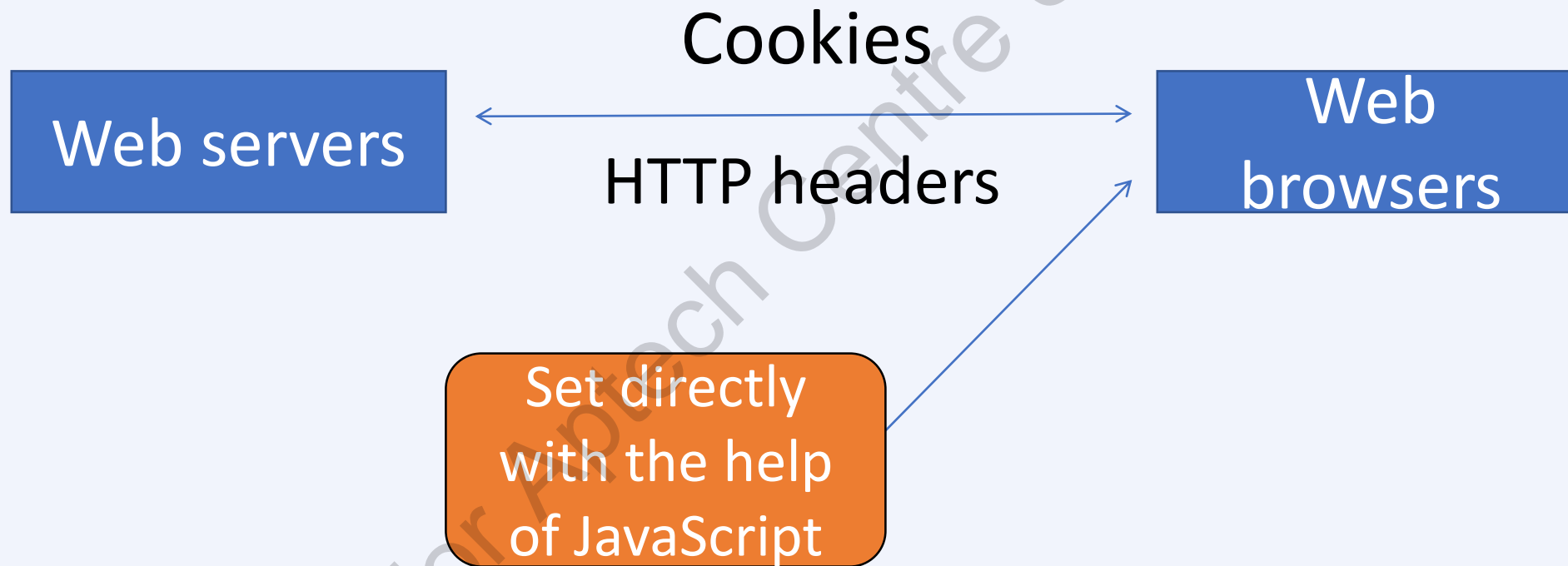
When the Web server receives a request from the Web browser, it identifies the user with the saved cookies

Websites use cookies to determine the following:



# Setting Cookies with PHP 1-2

- Setting a cookie means sending the cookie to the browser



# Setting Cookies with PHP 2-2

---

**PHP  
setcookie()  
function**

**Retrieving Cookie  
values using PHP  
\$\_COOKIE**

For Aptech Centre Use Only

# Working with Cookies in PHP [1-3]

- Modifying cookies

`setcookie()` function is not only used to create a cookie, but also to modify it.

Code snippet:

```
<?php
setcookie("name", "George Miller", time()+1800, "/", "", 0);
setcookie("city", "New York", time()+1800, "/", "", 0);
setcookie("name", "David Smith", time()+86400, "/", "", 0);
?>

<html>
<body>
<?php
echo "Cookies have been set.";
echo $_COOKIE["name"]. "<br />";
?>
</body>
</html>
```

For Apteck Centre Use Only



# Working with Cookies in PHP [2-3]

- Deleting a cookie

After a certain period of time, the cookie mentioned inside the `setcookie` function will automatically be deleted.

Code snippet:

```
<?php
    setcookie("name", "", time()- 30, "/", "", 0);
    setcookie("city", "", time()- 30, "/", "", 0);
?>
<html>
    <body>
        <?php
            echo "Cookies are deleted" ;
        ?>
    </body>
</html>
```

For Aptech Centre Use Only

# Working with Cookies in PHP [3-3]

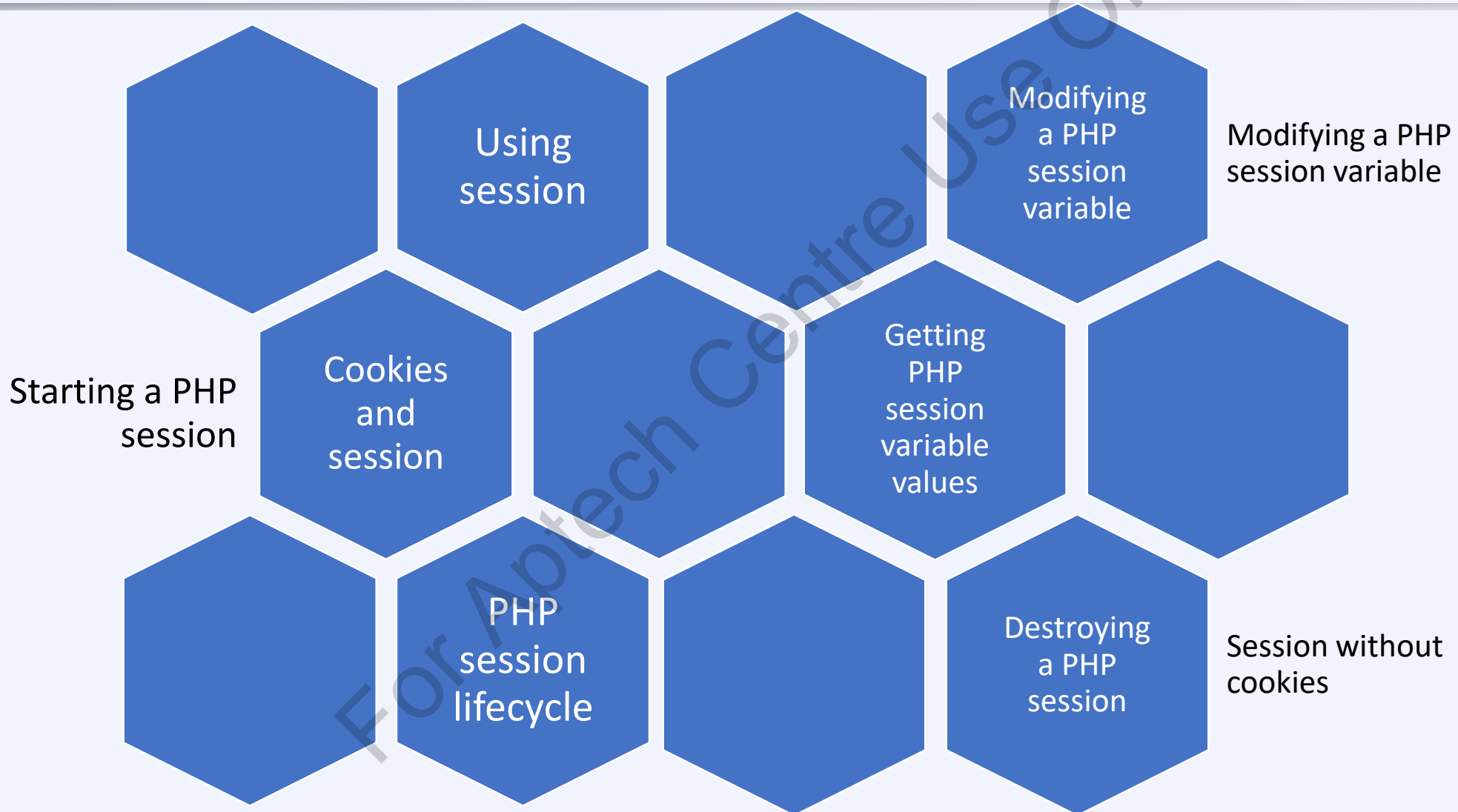
---

- Checking whether cookies are enabled

Browsers can be configured to disable cookies due to privacy concerns.

function `isset()` is used to check whether a cookie is set or not

# Working with Sessions in PHP [1-6]



# Working with Sessions in PHP [2-6]

---

## Disadvantages of using cookies:

Deletion of cookies

Multiple cookies to the same user

Size of the cookies

Cookies disabled

Cookies are prone to risk

# Working with Sessions in PHP [3-6]

Cookies	Sessions
Stores users information on the Web browser even after the user exits the Web browser	Stores user information on the Web server and destroys it once the user exits the Web browser
Allows users to disable cookies	Does not allow users to disable session
Has size limits	Does not have size limits

# Working with Sessions in PHP [4-6]

- Using Session

PHP creates a unique identifier for the session. This is a random string of 32 hexadecimal digits.

The PHPSESSID cookie is automatically sent to the users computer to save the unique session identification string.

An automatically created file on the Web server in the designated temporary directory bears the name of the unique identifier.

For Aptech Centre Use Only

# Working with Sessions in PHP [5-6]

## • PHP session lifecycle

- `Session_unset()` function: Removes all saved values from the session variable.
- `Session_destroy()` function: Destroys the session.

Starting the session

Starts with making a call to the `session_start()` function.

function `isset()` checks whether the session variable is already set.

Registering the session variable

`$_SESSION[]` saves the session variables which can be accessed during a session's lifetime

Ending the session

# Working with Sessions in PHP [6-6]

## Modifying a PHP Session Variable

- Simply overwrite the new value over the old one in the session variable.

## Turning ON Auto Session

- PHP facilitates automatic start of a session through `session.auto_start`

## Using sessions Without Cookies

- Session ID is sent to the browser through an alternate method when user does not want to save the data
- Current session id is retrieved using `session_id()`



# Summary

---

- A cookie is a small file created at the Web server side and stored at the client Web browser.
- PHP cookies can be created, modified, retrieved, and even deleted.
- The `setcookie()` function in PHP enables to create and modify cookies.
- Cookies have some disadvantages; hence, PHP sessions are sometimes used in place of them.
- PHP sessions temporarily store and pass the information from one Web page to another, until the Website is closed.
- The major difference between cookies and sessions is that cookies store information on local computer, whereas a session stores information on the Web server.
- User data is stored in session variables to be utilized across multiple Web pages and they last until the Web browser is closed by the user.
- There are three stages in the lifecycle of a session based on communication between Web browser and Web server, namely, Starting the session, registering the session variable, and ending the session.
- A PHP session starts by making a call to the `session_start()` function.
- The `session_unset()` function removes all the saved values from the session variable and cleans the session variable.
- Session information can be sent from one page to another without cookies, by using SID.