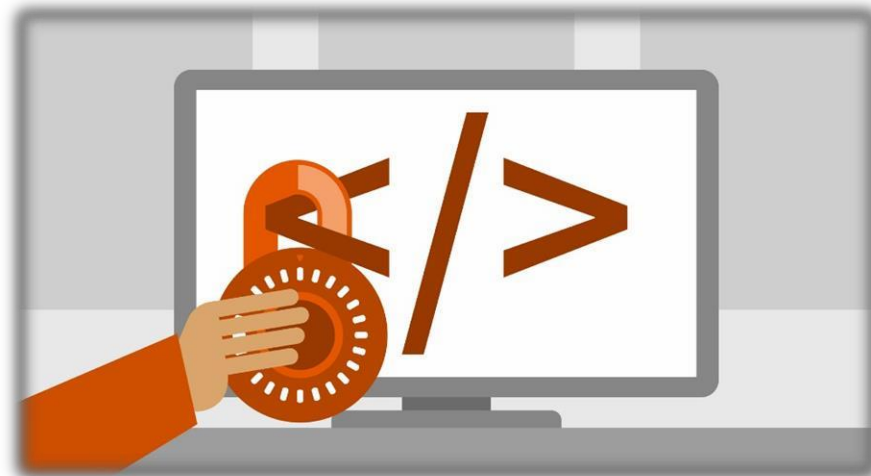


Lecture 9 - ICT1233

PHP Sessions



DEPARTMENT OF ICT

FACULTY OF TECHNOLOGY

Objectives

After the successful completion of this lecture, students should be able to,

- Use sessions with php



The need for persistence

- Consider these examples
 - Counting the number of “**hits**” on a website
 - *i.e.*
 - how many times does a client load your web page source
- Somehow your **.php** needs to remember previous instances of it being requested by a client

Sessions

- **Session:** You work with an application, you open it, do some changes, and then you close it
- Session variables store user information (in the form of variables) to be used across multiple pages (e.g. username, items selected, etc)
- You can store this information in the server side for later use, using PHP session



When should you use sessions?

- When there is a need for data to be stored on the server
- Unique session information for each user
- Transient data, only relevant for short time
- Data does not contain secret information
- Similar to Cookies, but it is stored on the server

Sessions are great for storing temporary data about your visitors, particularly when you do not want that data to be accessible from outside of your server

How PHP Sessions Work

- Assign a session ID number
- Store session variables in a file on the server
- Passes the session ID number to every page
- Gets the variables from session file for each new session page

Starting a Session

- When you call `session_start()`, PHP will check to see whether the visitor sent a session,
 - If it did, PHP will load the session data
 - Otherwise, PHP will create a new session file on the server, and send an ID back to the visitor to associate the visitor with the new file

Starting a Session

- Each visitor has their own data locked away in their unique session file, you need to call ***session_start()*** before you try to read session variables
 - Otherwise you will not have access to their data
- The ***session_start()*** function must appear at the top of EVERY page, even BEFORE the <html> tag

Adding Session Data

- All of session data is stored in the session super global array, **\$_SESSION**, which means that each session variable is one element in that array, combined with its value
- Adding variables to this array is done in the same way as adding variables to any array
- Session variables will still be there when the user visits to another page

Reading Session Data

- Once you have put your data safely away, it becomes immediately available in the `$_SESSION` super global array with the key of the variable name you gave it
- Session variables are not passed individually to each new page, instead they are retrieved from the session we open at the beginning of each page (`session_start()`)

Exercise

- Write down the php code to print the session variable which was created by the previous script

Removing Session Data

- To remove all global session variables and destroy the session, use ***session_unset()*** and ***session_destroy()***
- **unset()** function is used to free the specified session variable
- Completely destroy the session by calling the ***session_destroy()*** function
- *session_destroy()* reset your session and you will lose all your stored session data

unset()

```
<?php
    session_start();
    if(isset($_SESSION['views']))
        unset($_SESSION['views']);
?>
```

Summary

- PHP sessions pass data from one page to another
- However, this session information is temporary and is usually deleted very quickly after the user has left the website that uses sessions
- PHP Cookies are used to store information about the user and their visits on the user's computer

Questions.....

