

Today's Goals

Today's lecture will cover:

- An introduction to sessions, their purpose, and their use
- · Starting and stopping a session
- Using session variables

Purpose of a Session

- In general, HTTP is a "stateless" system, i.e., clients access documents through links without regard to past interactions
- · This is not acceptable when it comes to managing a complex interaction with a client such as:
 - the use of a shopping cart;

CSCI 2910 – Client/Server-Side Programming

- logging into a database or other secure site; or
- tracking a user's settings/data values as he or she progresses through a site.

Mechanics of a Session

Sessions in PHP – Page 2 of

Sessions in PHP – Page 4 of 1

Sessions in PHP – Page 6 of

- Session is identified using a session ID (32) digit hexadecimal value)
- The session ID is transmitted between the client and server with each HTTP request and response
- · Client keeps track of a session through the use of a cookie
- · Server keeps track of a session through locally stored text files or a database

CSCI 2910 – Client/Server-Side Programming

Mechanics of a Session (continued)

- · Databases are used for large traffic applications while text files are used for lower traffic.
- The server maintains the session variables. in the text file or database.
- To prevent security risks due to someone hijacking an old session and to avoid clogging the server with unused sessions. the server will clean up old sessions after a specified timeout period.

CSCI 2910 – Client/Server-Side Programming

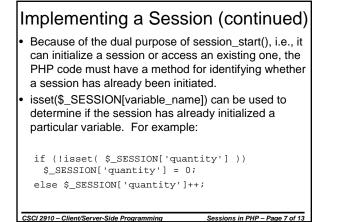
Sessions in PHP – Page 5 of

Sessions in PHP – Page 3 of 1

Implementing a Session

- session_start() creates a new session or finds an existing session. Basically, it identifies a session and accesses the session's variables if it is an existing session.
- Once a session has been started, the session's variables are accessed through a superglobal associative array called \$_SESSION. (This is the same sort of array as \$_GET and \$_POST.)
- Example: \$ SESSION[variable name]

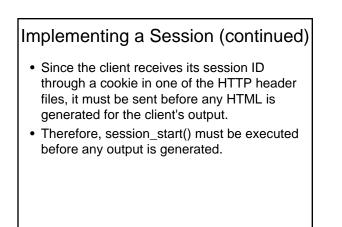
CSCI 2910 – Client/Server-Side Programming



Implementing a Session (continued)

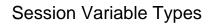
- A variable can be removed from a session using the unset() function
- Example: unset(\$_SESSION['quantity']);
- All session variables can be removed by simply re-initializing the \$_SESSION array
- Example: \$_SESSION = array();

CSCI 2910 – Client/Server-Side Programming



CSCI 2910 – Client/Server-Side Programming

Sessions in PHP – Page 9 of 13

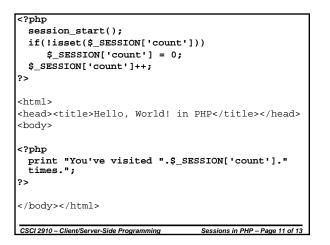


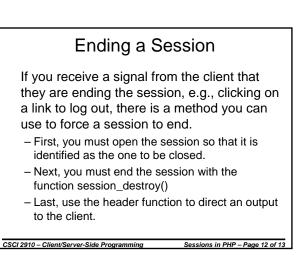
Sessions in PHP - Page 8 of 1

Sessions in PHP – Page 10 of 1

- A session variable can be of any type or object
- If a session variable is an object, be sure to define the object before running session start().
- If an existing session that uses an object is opened before the object is defined, it will cause problems.
- The following slide presents an example

CSCI 2910 – Client/Server-Side Programming





```
Ending a Session (continued)
<?php
// Begin by accessing the session
```

```
session_start();
// Close the session
session_destroy();
// Direct output to the client
header("Location: logout.html");
?>
```

CSCI 2910 – Client/Server-Side Programming Sessions in PHP – Page 13 of 13