

# SOEN 287

## Tutorial 09 – PHP session

Jing Li



# PHP Sessions

- HTTP is stateless, any data is forgotten when the connection is closed.
- A PHP session variable holds information about one single user, and are available to all pages in one application. It allows you to store user information on the server for later use .
- Sessions can be destroyed by PHP pages. Sessions expire after a time of inactivity.

# Start a session

```
<?php session_start();?>
```

```
<html>
```

```
    <body>
```

```
        . . . .
```

```
    </body>
```

```
</html>
```

**Note:** The session\_start() function must appear **BEFORE** the <html> tag.

# Store session data

**store data into a session**

```
$_SESSION['user name'] = 'jimmy';
```

**Retrieving session data**

You can retrieve session data as well. Again, you have to start the session first.

```
session_start();
```

```
if(isset($_SESSION['user name ']))
```

```
    . . . .
```

# Destroy a session

The unset() function is used to free the specified session variable

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

The session\_destroy() is used to destroy all your stored session data.

```
<?php
    session_destroy();
?>
```

# PHP Cookies

- A cookie is used to identify a user.
- A cookie is a small file that the server embeds on the user's computer.
- Each time the same computer requests a page with a browser, it will send the cookie too.

# PHP Cookies

## Create a Cookie:

```
<?php  
setcookie("user", "Alex Porter", time()+3600);  
?>
```

## Retrieve a Cookie Value:

```
if (isset($_COOKIE["user"]))
```

## Delete a Cookie:

```
setcookie("user", "", time()-3600);
```

# Cookies vs. Sessions

- Cookies are stored on the client
- web browsers have a limit on how big cookies
- Sessions are stored on the server
- sessions can be any size you want

PHP session actually stores the Session ID by cookie



# PHP arrays

Array is an ordered map.

```
$old = array('a', 'b', 'c');
```

```
$foo = ['a', 'b', 'c'];
```

## Associative Arrays

```
$bar = [0 => 'a', 1 => 'b', 2 => 'c'];
```



named keys

# Practice 1

- Create a new php page that opens a session.
- If the session['views'] is not exist, create the session. If the session exists, count and show how many times the user visits this page.
- If the session has been visited more than 10 times, clear the session.

# Practice 2

## ❑ Most frequent

Parameter(s): String Array

Description : Outputs the string (element) that appears the most frequently within the array. Strings should be case insensitive.

# Practice 2

## ❑ Function: **uppercaseFirst**

Parameter(s): String

description : For each word in the string, capitalize (uppercase) the first letter and return the modified string.

## ❑ Function: **splitSort**

Parameter(s): String

description : Given the string, place each word into an array index and sort it alphabetically before returning the filled array.