

SOEN 287

WEB PROGRAMMING

PHP – 4

Sessions & Cookies



PHP Cookie



Cookies are used to often to perform following tasks

1. Session management:
Ex: shopping cart
2. User identification – a way of remembering a user
3. Tracking/Analytics – used to track the user

PHP Cookie



- A *cookie* is a name/value pair that is passed between a browser and a server in the HTTP header
- A small file that the server embeds on the **user's computer** (client side).
- Each time the same computer requests a page with a browser, it will send the cookie too.
- With PHP, you can both create and retrieve cookie values.

PHP Cookie ...



- **Creating a cookie:** `setcookie()`
- Return a boolean
- Cookies are implicitly deleted when their lifetimes are over
- Must appear before html tag!

- **Syntax:**

```
setcookie(name, value, expire);  
(name is the only one that is required)
```

- **Example:**

```
setcookie("user", "Anna S.", time()+1800);  
where 1800 sec = 30 minutes
```

PHP Cookie ...

- ```
<?php
$cookie_name = "user";
$cookie_value = "John Doe";
setcookie($cookie_name, $cookie_value, time() + (86400 * 30), "/"); // 86400 = 1 day
?>

<html>
<body>

<?php
if(!isset($_COOKIE[$cookie_name])) {
 echo "Cookie named '" . $cookie_name . "' is not set!";
} else {
 echo "Cookie '" . $cookie_name . "' is set!
";
 echo "Value is: " . $_COOKIE[$cookie_name];
}
?>

</body>
</html>
```

# PHP Cookie ...



- To retrieve a cookie value use the `$_COOKIE` variable
- Example:

```
<?php
if (isset($_COOKIE["cookie_name"]))
echo "Welcome ".$_COOKIE[" cookie_name
"] . " !
";
else
 echo "Welcome guest !
";
?>
```

# PHP Cookie – Can use to count visits

- Can set a *cookie* for number of visits, and increment this counter.

- Example 1: Setting counter 1<sup>st</sup> time

```
$_COOKIE["numOfVisits"]=0;
$count= $_COOKIE["numOfVisits"]+1;
$length=time()+3600*24*365; // a year
setcookie("numOfVisits", $count, $length);
```



- Example 2: To see if a cookie has already been set

```
if (isset($_COOKIE["numOfVisits"])) ...
```

# PHP Sessions

- A *session* is a conversation between the server and a client. A conversation consists of series of continuous request and response.
- Stored on server side and closed once user closes browser.
- Session variables hold information about **one** single user, and are available to all pages in one application.



# PHP Sessions

- When there is a series of continuous request and response from a same client to a server, the server cannot identify from which client it is getting requests.
- Session *variables* last until the user closes the browser.

# When a session is started following things happen ...

([http://www.tutorialspoint.com/php/php\\_sessions.htm](http://www.tutorialspoint.com/php/php_sessions.htm))

1. PHP first creates a unique identifier for that particular session which is a random string of 32 hexadecimal
2. A cookie called **PHPSESSID** is automatically sent to the user's computer to store unique session identification string.
3. A file is automatically created on the server in the designated temporary directory.

# When a session is started following things happen ...

([http://www.tutorialspoint.com/php/php\\_sessions.htm](http://www.tutorialspoint.com/php/php_sessions.htm))

4. When a PHP script wants to retrieve the value from a session variable, PHP automatically gets the unique session identifier string from the `PHPSESSID` cookie and then looks in its temporary directory for the file bearing that name and a validation can be done by comparing both values.
5. A session ends when the user closes the browser or after leaving the site, the server will terminate the session after a predetermined period of time, commonly 30 minutes duration.

# PHP Session Support

- Start a *session*:

To be able to store information about a user, need to 1<sup>st</sup> start a session.

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

- Checks to see if a session has already been started, then if none currently exists starts one.
- Also alerts PHP engine to expect *session* variables.

Must appear before 1<sup>st</sup> html tag!

# PHP Session Support ...

- `$_SESSION`:  
a global associative array which holds the session states/variables
- To create/store a session state/variable  
`$_SESSION[ 'count' ]=$counter;`
- To delete/free a session state/variable  
`unset($_SESSION[ 'count' ] );`
- To destroy a session (lose all states)  
`session_destroy();`

# A first example of sessions

- `session1.php`  
Starts the session
- `session2.php`  
Displays info from started session
- `session3.php`  
Changes one of the session variables value
- `session4.php`  
Ends and clears a session

# Crazy Pizza Shop ...

## Example of session control

### **Crazy Pizza Shop**

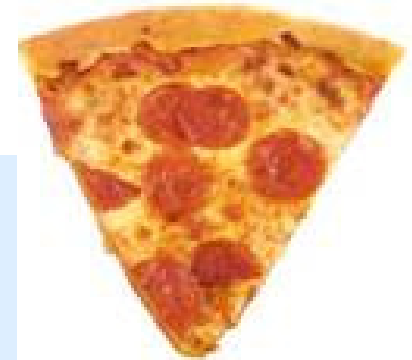
#### **Your Pizza Order:**

Choose Crust: ☐ Thin crust ☐ Thick crust

#### **Add Toppings:**

☐ pepperoni ☐ ham

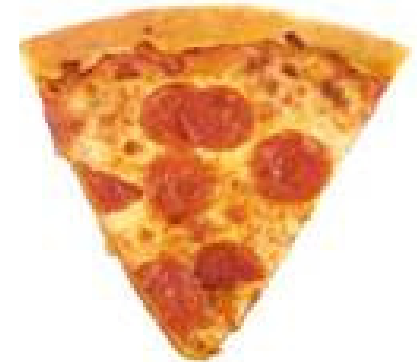
☐ mushroom ☐ onion ☐ Extra Cheese



pizza.php

# Logic of Crazy Pizza Shop

1. If there is no form submission,
  - a. create a new session
  - b. Session variables are initialized
  - c. Form is displayed
2. Else if order form continuing – resume processing
  - a. Order form data is processed
  - b. Display form with current order status  
(user can change crust and/or add another topping)
3. Else if order is done
  - a. Session is terminated
  - b. Order confirmed and link to a new order





# Crazy Pizza Shop – pizza.php



//1<sup>st</sup> time ...

```
if (empty($_POST['submit'])) {
```

//A

```
 session_start();
```

//B

```
 $_SESSION['crust']="";
```

//C

```
 $_SESSION['toppings']="";
```

```
 include("head.php");
```

```
 require("orderform.php"); }
```

//D

...

# Crazy Pizza Shop – pizza.php



//Else if order form continuing – resume processing

```
elseif ($_POST['submit'] == " Enter ") //E
 session_start(); // before any output
 processOrder(); //F
 include("head.php");
 require("orderform.php"); }
```

...

`<input type="submit" name="submit" value="Enter " /> // in orderform.php`

# Crazy Pizza Shop – pizza.php



//Continuing selection

```
elseif($_POST['submit'] == " Done "){ //G
 session_start(); session_end(); //H
 include("head.php");
 echo '<p>Your order is complete.
 Thanks for your business.</p>';
 echo '<p>Another Order</p>';
```

`<input type="submit" name="submit" value="
Done " /> // in orderform.php`

# Crazy Pizza Shop – orderform.php

<h3>Your Pizza Order:</h3>

<?php

```
$cr = $_SESSION["crust"];
$top = $_SESSION["toppings"];
echo '<p style="color: #600;
 font-weight: bold">';
if ($cr) { echo "$cr, "; }
if ($top) { echo "$top</p>"; }
```

?>

<form method="post" action=""><p>Choose  
Crust:

...



# Crazy Pizza Shop ...



Most Visited Getting Started Outlook Web App MyConcordia - Yo

## Crazy Pizza Shop

**Your Pizza Order:**

**Thin crust, mushroom**

Choose Crust: ☐ Thin crust ☐ Thick crust

Add Toppings:

☐ pepperoni ☐ ham

☐ mushroom ☐ onion ☐ Extra Cheese

## processOrder()

```
function processOrder() {
 $cr = $_POST["crust"];
 if ($cr) {
 $_SESSION['crust'] = $cr; }
 $top = $_SESSION['toppings'];
 $it = $_POST["top"];

 if ($it && ! strstr($top,$it)) {
 if ($top) {
 $_SESSION['toppings'] = "$top, $it"; }
 else {
 $_SESSION['toppings'] = $it; }
 }
 } // function
```

# Crazy Pizza Shop ...



## Crazy Pizza Shop

### Your Pizza Order:

Thin crust, mushroom, extra cheese

Choose Crust: ☐ Thin crust ☐ Thick crust

### Add Toppings:

☐ pepperoni ☐ ham

☐ mushroom ☐ onion ☐ Extra Cheese

Enter

Done

# Crazy Pizza Shop ...



## Crazy Pizza Shop

### Your Pizza Order:

**Thick crust, mushroom, extra cheese**

Choose Crust: ☐ Thin crust ☐ Thick crust

### Add Toppings:

☐ pepperoni ☐ ham  
☐ mushroom ☐ onion ☐ Extra Cheese

Enter

Done



# Crazy Pizza Shop ...

- When click on Done



## **Crazy Pizza Shop**

Your order is complete. Thanks for your business.

[Another Order](#)