

Sessions

Objectives

- Understand what it means a web session with multiple forms
- Learn how to use hidden fields to manage multi-form sessions
- Learn how to use cookies to track user data
- Learn how to use PHP session functions for tracking user data

This lecture covers

- Multi-Form Session
- Hidden fields
- Sending e-mail
- Cookies
- Sessions

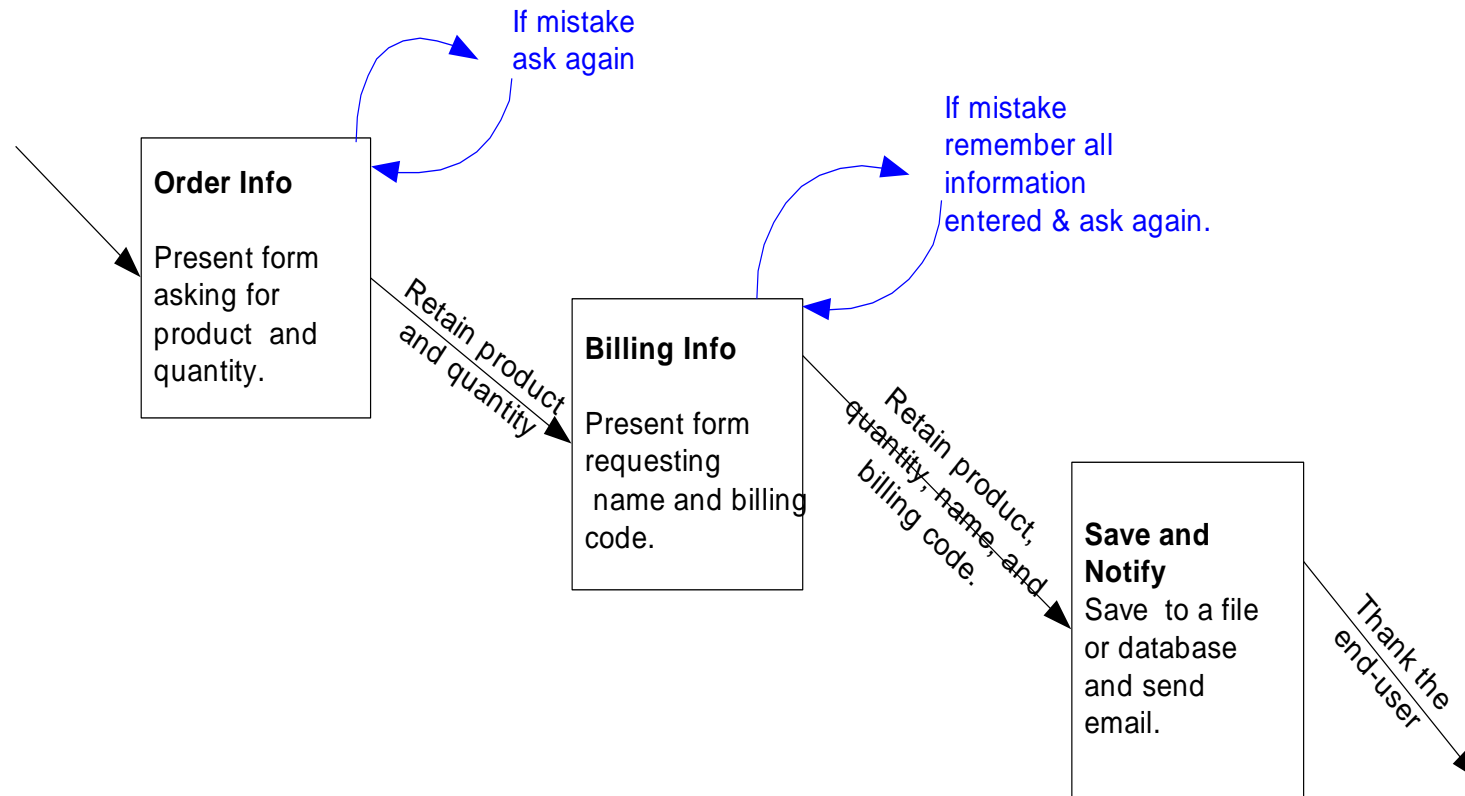
This lecture covers

- **Multi-Form Session**
- Hidden fields
- Sending e-mail
- Cookies
- Sessions

Multi-Form Session

- user navigates through a series of forms
- working together
- exchanging data
- example
 - online shopping

Example : multi-screen session



This lecture covers


- Multi-Form Session
- **Hidden fields**
- Sending e-mail
- Cookies
- Sessions

Hidden Fields

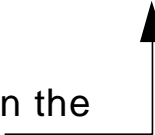
- use hidden fields to transfer data from one page to another
- not visible but are part of the form
- will be received by the action script
- still visible in source file

```
<input type="hidden" name="preference" value="Likes Power Tools">
```

Variable name
that will be available
to your PHP program



Variable's value in the
PHP program



Order Product x

← → ↻ ⓘ fsciences.ul.edu.lb/ch4/order.html ☆ ⬆

Happy Harry's Hardware Product Order Form

We have hammers, handsaws, and wrenches on special today!

Enter Item: Enter Quantity:

Order Product 2 x

← → ↻ ⓘ localhost/ch4/order2.php ⬆

Hidden value=Welcome!

You selected product=handsaws and quantity=4

Please enter your name: and billing code: (5 digits)

Order Product 3 x

← → ↻ ⓘ fsciences.ul.edu.lb/ch4/order3.php ☆ ⬆

Sending e-mail to order handling department at orders@hardwareville.com

The e-mail body is : *New Order: Product=handsaws Number= Cust= moh Code=12345.*

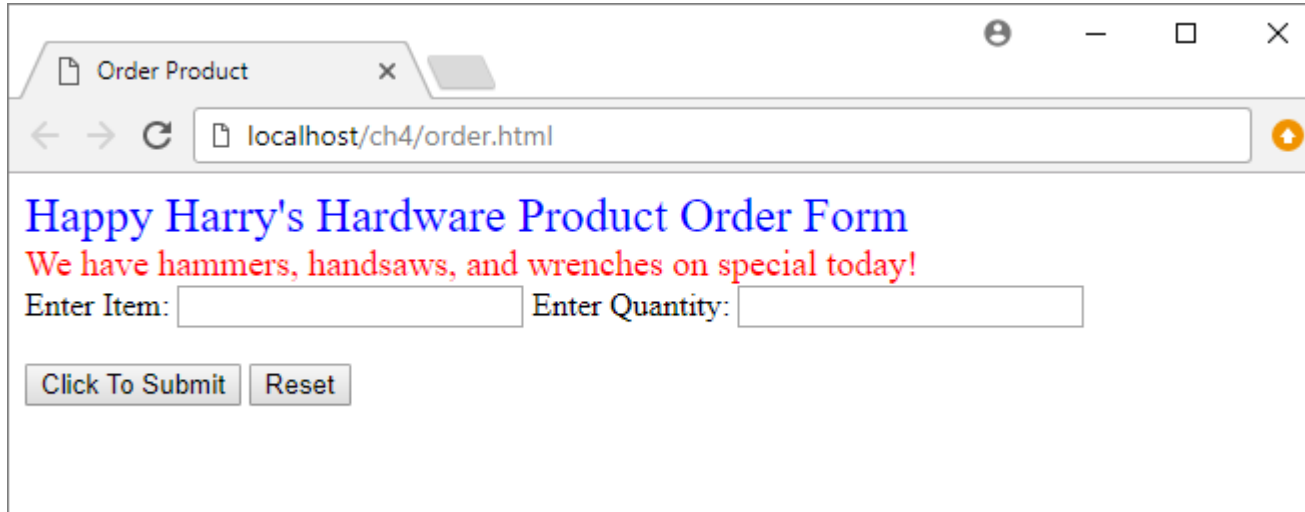
[E-mail sent. Thanks for ordering.](#)

By the way, sample hidden=Welcome!

Example

```
<html>
<head><title>Order Product</title></head>
<body>
  <form action="order2.php" method="post">
    <font color="blue" size="5">Happy Harry's Hardware Product
      Order Form</font><br>
    <font color="red" size="4"> We have hammers, handsaws,
      and wrenches on special today!</font>
    <input type="hidden" name="sample_hidden" value="Welcome!"> <br>
    Enter Item: <input type="text" name="product">
    Enter Quantity: <input type="text" name="quantity"><br><br>
    <input type="submit" value="Click To Submit">
    <input type="reset" value="Reset">
  </form>
</body>
</html>
```

Example



A screenshot of a web browser window. The title bar shows a single tab labeled "Order Product". The address bar contains the URL "localhost/ch4/order.html". The page content includes a blue heading "Happy Harry's Hardware Product Order Form", a red promotional message "We have hammers, handsaws, and wrenches on special today!", and a form with two input fields labeled "Enter Item:" and "Enter Quantity:". Below the form are two buttons: "Click To Submit" and "Reset".

Order Product

localhost/ch4/order.html

Happy Harry's Hardware Product Order Form

We have hammers, handsaws, and wrenches on special today!

Enter Item: Enter Quantity:

Click To Submit Reset

Receiving Hidden Fields in Web Sessions – order2.php

```
<html>
<head><title>Order Product 2 </title></head>
<body>

    <form action="order3.php" method="post">
        <?php  $sample_hidden = $_POST["sample_hidden"];
                $product = $_POST["product"];
                $quantity = $_POST["quantity"];

                echo "<font size=5 color=blue>";
                echo "Hidden value=$sample_hidden <br></font>";
                echo "You selected product=$product and quantity=$quantity<br /><br>";
                echo "<input type=\"hidden\" name=\"product\" value=\"\$product\">";
                echo "<input type=\"hidden\" name=\"quantity\" value=\"\$quantity\">";
                echo "<input type=\"hidden\" name=\"sample_hidden\" value=\"\$sample_hidden\">";
                echo 'Please enter your name: <input type="text" name="name">';
                echo ' and billing code: (5 digits) <input type="text" name="code">';
                echo '<br> <input type=submit value="Process Order">';
                echo '<input type=reset>'; ?>

    </form>
</body></html>
```

This lecture covers

- Multi-Form Session
- Hidden fields
- **Sending e-mail**
- Cookies
- Sessions

Send an Email

- send an email from a PHP script
- `mail()`
 - uses Simple Mail Transfer Protocol (SMTP)
 - `mail(to_address, subject, message, extra_headers);`

Example

```
$dest='orders@hardwareville.com';  
$subject = 'New Hardware Order';  
$message = 'Enclosed is a new order for 12 hammers.\n Thanks.';  
$extra = 'From: harry@hardwareville.com';  
  
mail( $dest, $subject, $message, $extra );
```

Complete example

- Save and notify the client
- order3.php
- Access to variables
 - \$product
 - \$quantity
 - \$sample_hidden


```
<html>
<head><title>Order Product 3</title> </head>
<body>
<?php
    $sample_hidden = $_POST["sample_hidden"];
    $quantity=$_POST["$quantity"];
    $product = $_POST["product"];
    $name=$_POST["name"];
    $code=$_POST["code"];

    $email='orders@hardwareville.com';
    $body = "New Order: Product=$product ";
    $body .= "Number=$quantity Cust=$name Code=$code";

    echo "<font size=4> <br>Sending e-mail to order handling department at $email</font>";
    echo "<br>The e-mail body is <i>: $body. </i>";
    $from = 'harry@hardwareville.com';
    $subject = "New order from $name";
    mail($email, $subject, $body, "From: $from");

    echo '<br><font color="blue"> E-mail sent. Thanks for ordering. </font>';
    echo "<br>By the way, sample hidden=$sample_hidden";
?>
</body></html>
```

order3.php

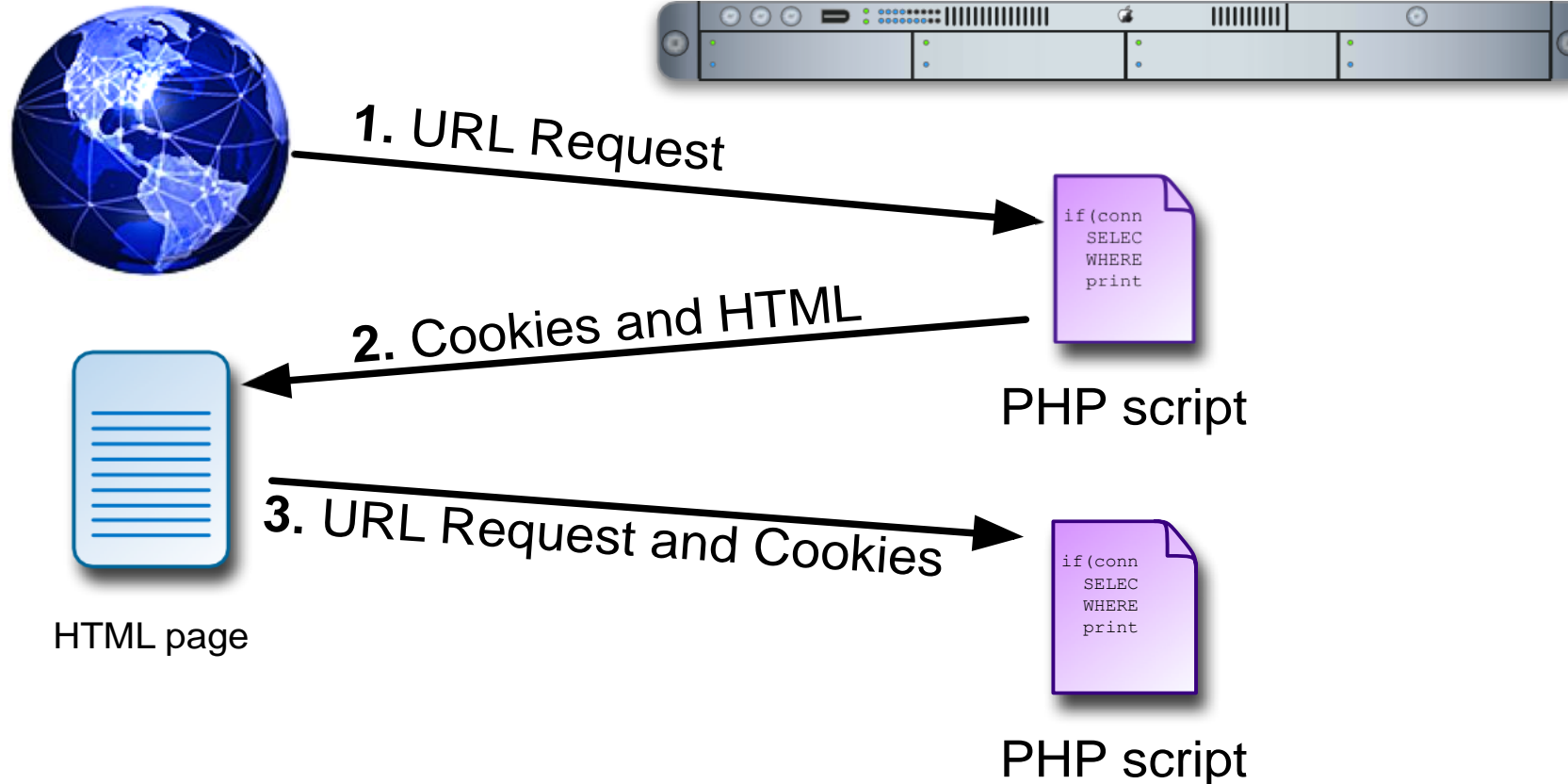
This lecture covers

- Multi-Form Session
- Hidden fields
- Sending e-mail
- **Cookies**
- Sessions

Using Cookies

Web Browser (client)

Server



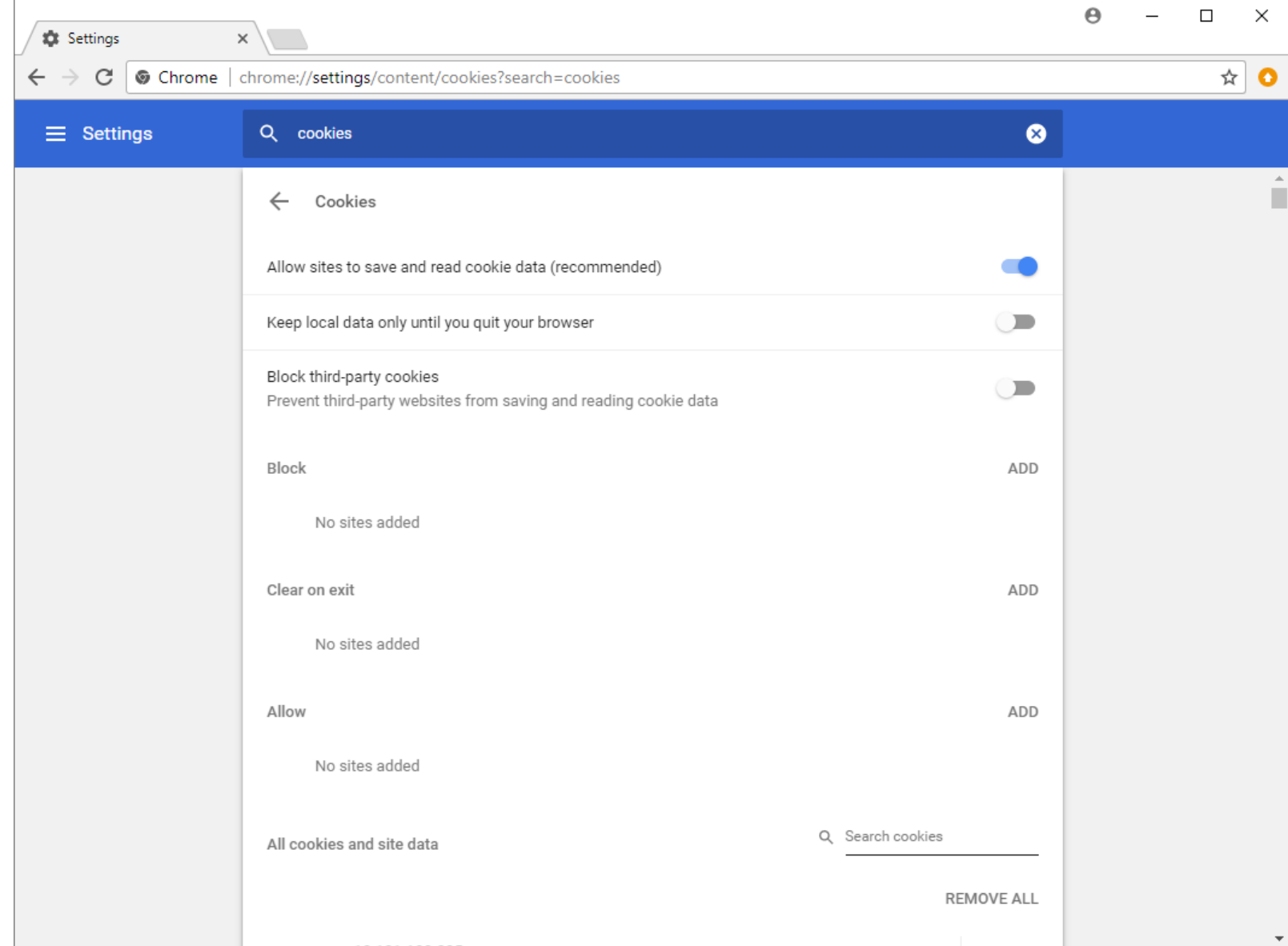
Cookies

- small files saved on the client's hard disk
- by a web application
- during a visit
- to be read during subsequent visits

Limitation of Cookies

- Cookies are small pieces of data that a Web application can save when a user visits the Web page.
 - Stored on the visitor's hard drive
 - a Web page script can read the previously stored browser cookie data
- Users can easily disable the cookies feature.
- People move around.
- Users may delete cookies.
- PHP sets limit on cookies

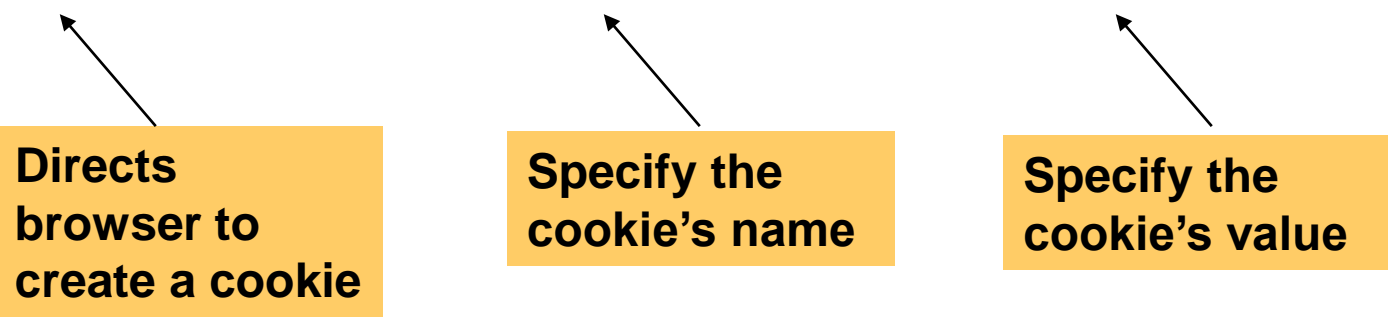
Deactivate cookies



Create and read a cookie

- Cookies can be set in memory or on hard disk
 - Set on hard disk are deleted when browser closes
 - Can use the `setcookie()` script

```
setcookie('Customer_name', 'Denise');
```



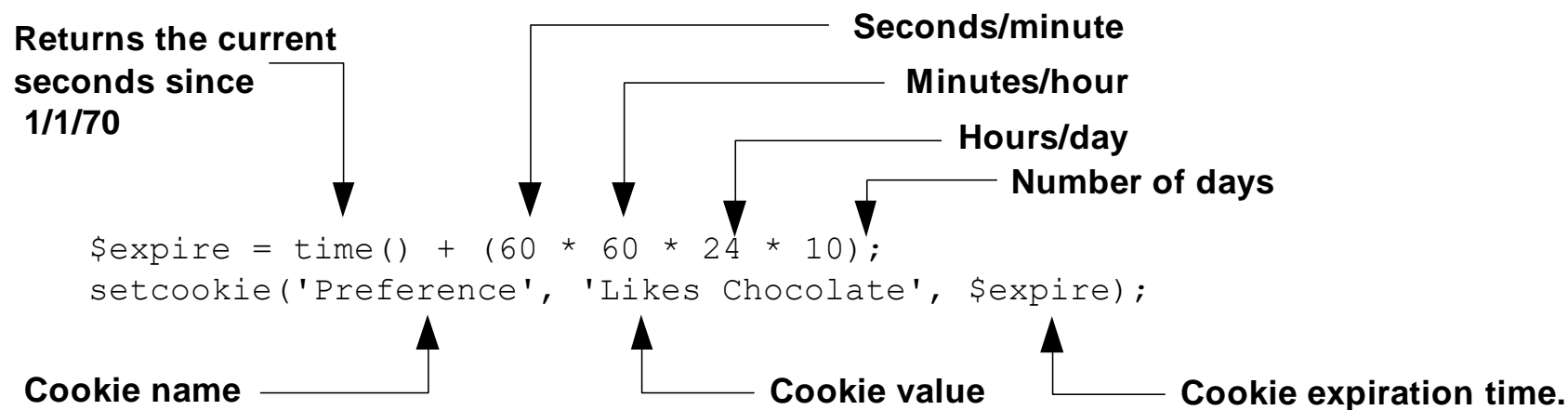
**Directs
browser to
create a cookie**

**Specify the
cookie's name**

**Specify the
cookie's value**

Setting A Cookie on a Hard Drive

- You need to use the `time()` function when want to set a cookie on a hard drive.



Example : survey (setcookie1.html)

- Suppose a front-end web page asks for some survey information:

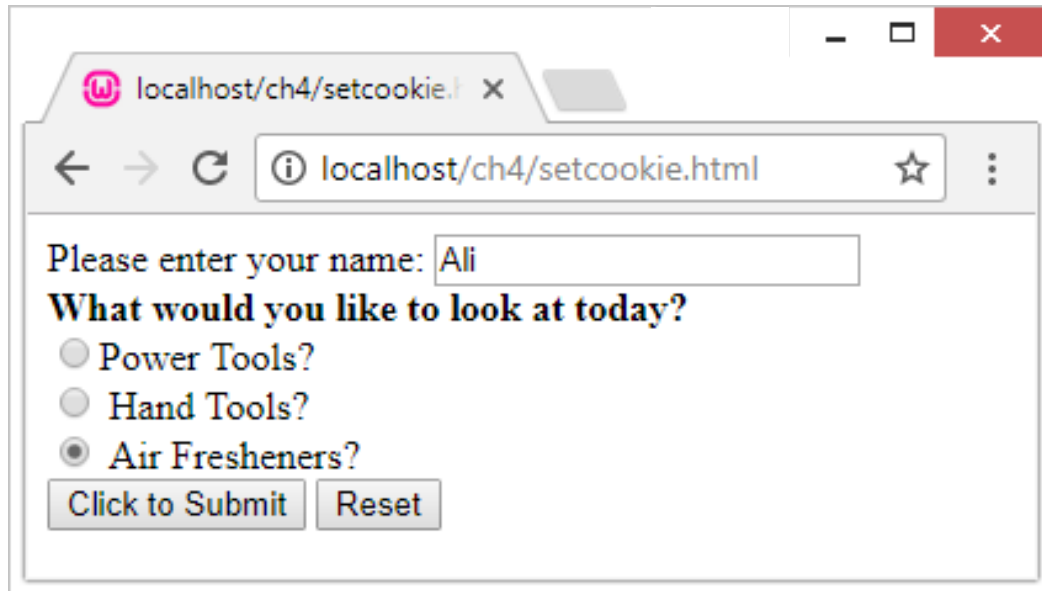
```
<form action="setcookie.php" method="POST">
Please enter your name: <input type="text" name="custname"><br>
<b>What would you like to look at today?</b><br>

<input type="radio" name="prefers" value="power tools" checked="checked">Power Tools?<br>
<input type="radio" name="prefers" value="hand tools"> Hand Tools? <br>
<input type="radio" name="prefers" value="air fresheners"> Air Fresheners? <br>
<input type="Submit" value="Click to Submit"> <input type="reset" value="Reset">
</form>
```

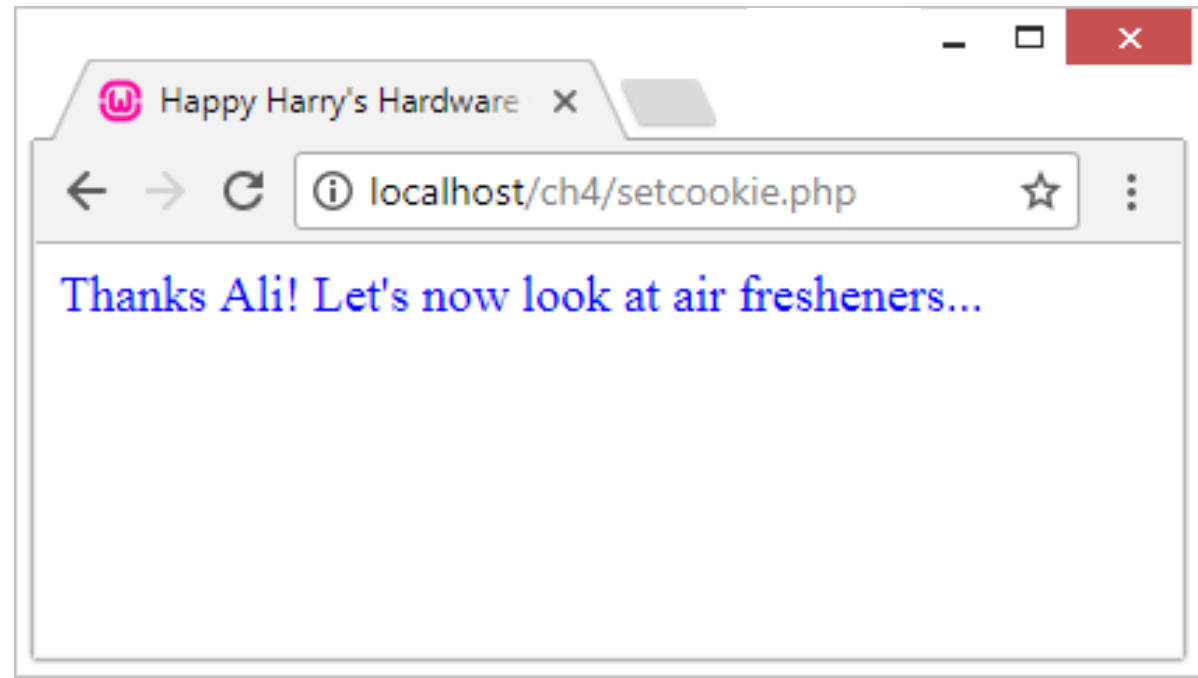
Script PHP (setcookie.php)

```
<?php
    $prefers = $_POST["prefers"];
    $custname=$_POST["custname"];
    $expire = time() + (60 * 60 * 24 * 30);
    setcookie("name", $custname, $expire);
    setcookie("preference", $prefers, $expire);
?>
<html>
<head><title>Happy Harry's Hardware Catalog </title></head>
<body><font size=4 color="blue">
<?php
    echo "Thanks $custname! ";
    echo "Let's now look at $prefers... ";
?> </font></body></html>
```

Output



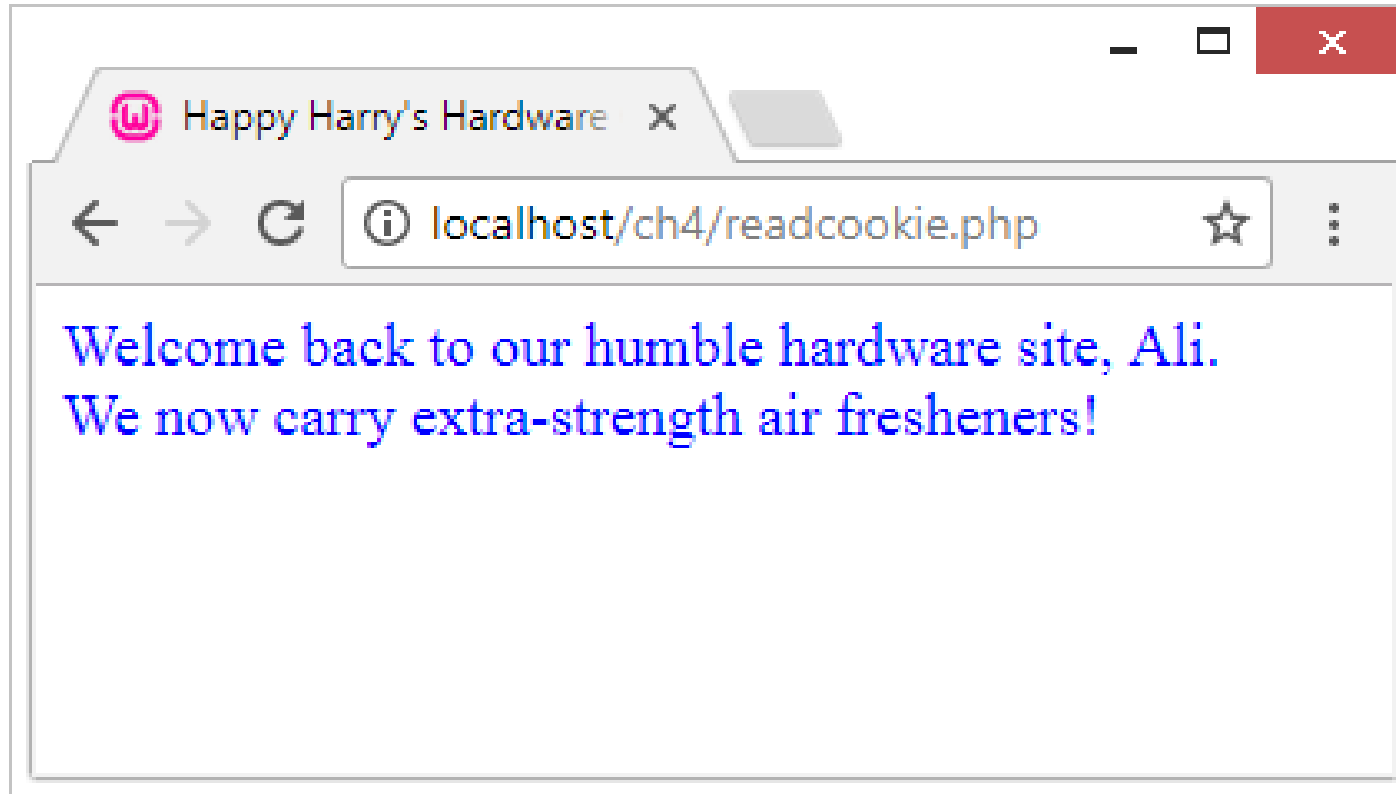
A screenshot of a web browser window. The address bar shows 'localhost/ch4/setcookie.html'. The page content includes a text input field with the value 'Ali' and the label 'Please enter your name:'. Below this is a section titled 'What would you like to look at today?' with three radio button options: 'Power Tools?', 'Hand Tools?', and 'Air Fresheners?'. The 'Air Fresheners?' option is selected. At the bottom of this section are two buttons: 'Click to Submit' and 'Reset'.



Reading Cookies

- Associative array `$_COOKIE[]`
- `$cust_name= $_COOKIE["name"];`

Example Script that read a cookie



Example Script that read a cookie

```
<html>
<head><title>Happy Harry's Hardware Catalog </title></head>
<body>
<?php
    $name = $_COOKIE["name"];
    $preference = $_COOKIE["preference"];
    echo '<font color="blue" size=4>';
    if (isset($name)){
        echo "Welcome back to our humble hardware site, $name."; }
    else { echo '<font color="red">';
        echo 'Welcome to our humble hardware site.</font>';
    } if ($preference == 'hand tools'){
        echo '<br> We have hammers on sale for 5 dollars!';
    } elseif ($preference == 'power tools'){
        echo '<br> We have power drills on sale for 25 dollars!';
    } elseif ( $preference == 'air fresheners'){
        echo '<br> We now carry extra-strength air fresheners!';
    } else {echo '<br> <font color="red">';
        echo 'We have drills and hammers on special today!';}
?></font>
</body>
</html>
```

This lecture covers

- Multi-Form Session
- Hidden fields
- Sending e-mail
- Cookies
- **Sessions**

Sessions vs. Cookies

- **Sessions**

- Generally more secure
- Able to store more data
- Can be used without cookies

- **Cookies**

- Easier to program
- Require less of the server
- Can be set to last longer

Sessions

- PHP supports two functions that enable you to retain data between forms
 - `session_start()` - either starts a new session or resumes one if a session exists
 - Run at the start of every script
 - By default creates a unique session ID stored as a cookie
 - `session_register()` - registers one or more variables as session variables

```
$name = 'Matthew';  
$preference = 'Soccer Equipment';  
session_register('name', 'preference');
```
- To destroy a session
 - `session_destroy()`

Sessions

- Support for PHP sessions is a way to preserve data between multiple accesses. This allows you to create custom applications, and increase the appeal of your site.
- Each visitor accessing your web page is assigned a unique identifier, called "session identifier". It can be stored either in a cookie or propagated in the URL.
- Session support allows you to store data between queries in the `$_SESSION` super-global array.
 - When a visitor visits your site, PHP will automatically check (if `session.auto_start` is enabled) or on request (explicitly with `session_start ()` or implicitly with `session_register ()`) if there is a session with the same name.
 - If this is the case, the previously saved environment will be recreated.

session_start()

- `session_start()`
 - starts a new session, in case there is not a session that is already open
 - or connects to the already open session
 - At the beginning of the script (before HTML display itself)
 - by default, the session identifier is saved on the client's hard drive as a cookie

session_register()

- this function can be replaced by a direct access to the global associative array `$_SESSION`
- example
 - `$name = 'Matthew';`
 - `$preference = 'Soccer Equipment';`
 - `$_SESSION['name'] = $name;`
 - `$_SESSION['preference'] = $preference;`

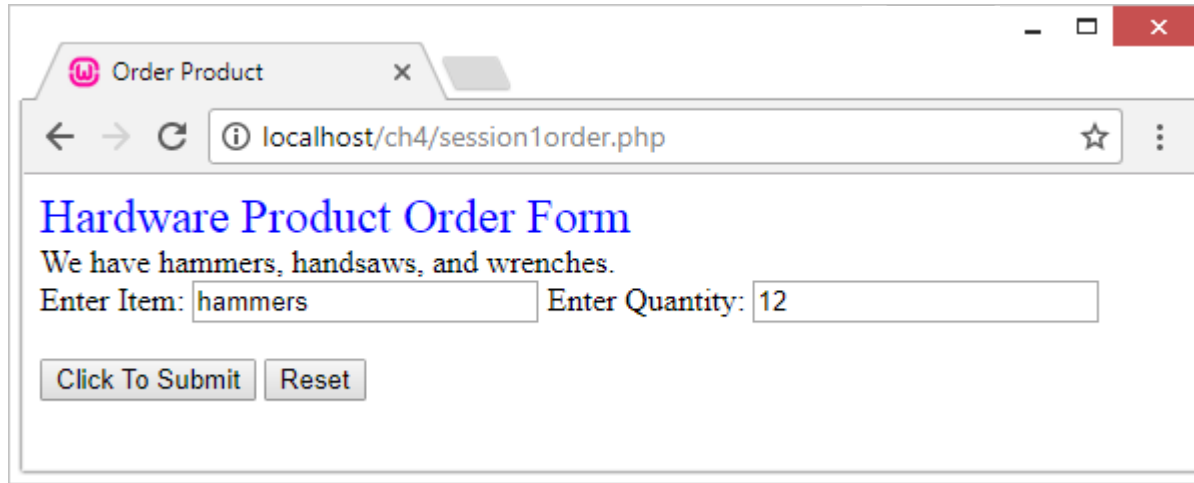
Example PHP Code

```
<?php session_start(); ?>
<html
<head><title>Order Product</title></head>
<body>
    <form action="sessions2order.php" method="post">
    <font color=blue size=5> Hardware Product Order Form </font>
    <br> We have hammers, handsaws, and wrenches.
    <br>Enter Item: <input text type="text" name="product" />
    Enter Quantity: <input text type="text" name="quantity" /><br>
    <?php
        $sample_hidden='Welcome Again!';
        $_SESSION["sample_hidden"]= $sample_hidden;
    ?>
    <br><input type="submit" value="Click To Submit"/>
    <input type = "reset" value = "Reset" />
    </form>
</body></html>
```

sessions2order.php

```
<?php
    session_start() ;
    $sample=$_SESSION['sample_hidden'];
    $product=$_POST['product'];$quantity=$_POST['quantity'];
?>
<html><head><title> Order Product 2 </title> </head>
<body>
    <form action="sessions3order.php " method="post">
        <?php
            echo "<font color=blue size=4> Sample hidden=$sample</font>";
            echo "<br>You selected product=$product and quantity=$quantity";
            $_SESSION['product'] = $product;
            $_SESSION['quantity'] = $quantity;
            echo '<br>Please enter your name';
            echo '<input text type="text" name="name">';
            echo ' and Billing Code: (5 digits)';
        echo '<input text type="text" name="code">';
        echo '<br> <input type=submit value="Process Order">';
        echo '<input type=reset>';
        echo '</form></body></html>'; ?>
```

Example output



A screenshot of a web browser window titled "Order Product". The address bar shows the URL "localhost/ch4/session1order.php". The page content includes a blue heading "Hardware Product Order Form", a text "We have hammers, handsaws, and wrenches.", and a form with two input fields: "Enter Item:" containing the text "hammers" and "Enter Quantity:" containing the text "12". Below the form are two buttons: "Click To Submit" and "Reset".



A screenshot of a web browser window titled "Order Product 2". The address bar shows the URL "localhost/ch4/sessions2order.php". The page content includes a blue heading "Sample hidden=Welcome Again!", a text "You selected product=hammers and quantity=12", and a form with two input fields: "Please enter your name" and "and Billing Code: (5 digits)". Below the form are two buttons: "Process Order" and "Reset".