Maintaining state through multiple forms

ONE LOVE. ONE FUTURE.

HTTP – stateless protocol

HTTP is a **stateless protocol**

- Once a web server completes a client's request for a web page, the connection between the two goes away.
- There is no way for a server to recognize that a sequence of requests all originate from the same client.

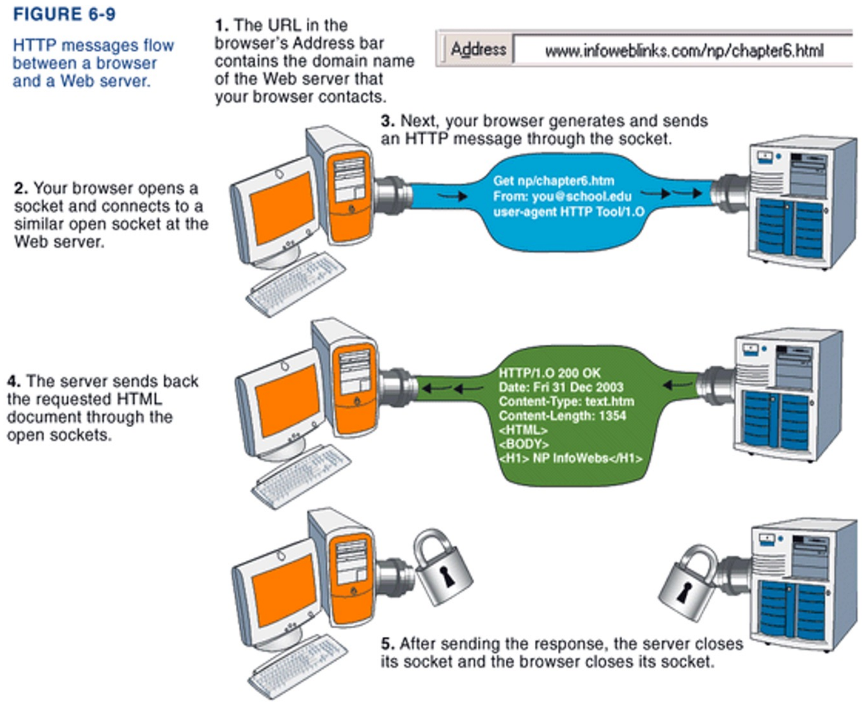


FIGURE 6-9

HTTP messages flow between a browser and a Web server.

1. The URL in the browser's Address bar contains the domain name of the Web server that your browser contacts.

2. Your browser opens a socket and connects to a similar open socket at the Web server.

3. Next, your browser generates and sends an HTTP message through the socket.

```
Get np/chapter6.htm
From: you@school.edu
user-agent HTTP Tool/1.O
```

4. The server sends back the requested HTML document through the open sockets.

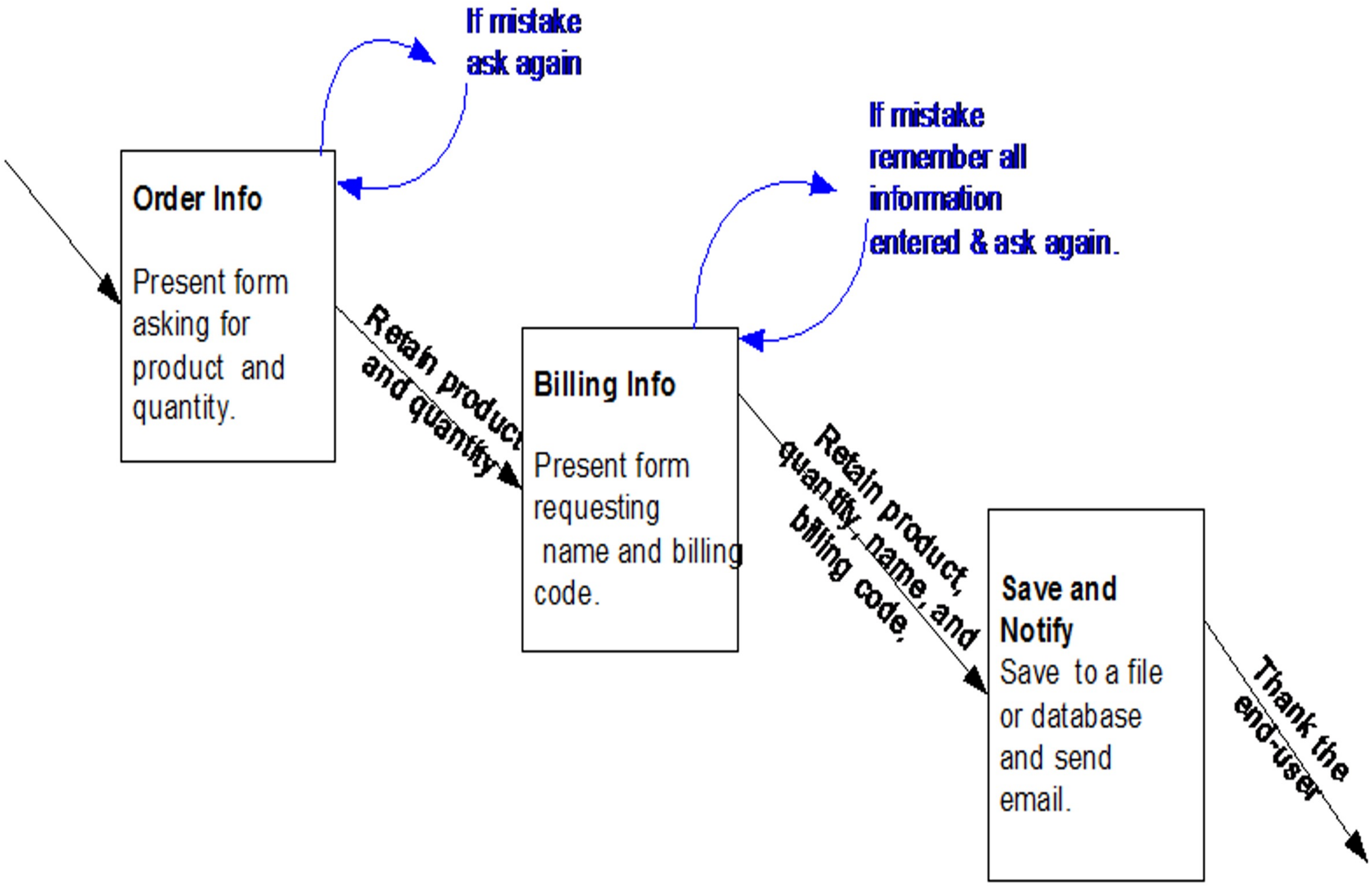
```
HTTP/1.O 200 OK
Date: Fri 31 Dec 2003
Content-Type: text.htm
Content-Length: 1354
<HTML>
<BODY>
<H1> NP InfoWebs</H1>
```

5. After sending the response, the server closes its socket and the browser closes its socket.

What Are Multiple-Form Web Sessions?

- A multiple-form Web session leads the user through a series of HTML forms that work together and pass data from form to form.
- E.g.
  - To build a shopping cart or on-line survey.
  - To save user authentication information from page to page
  - To store persistent user preferences on a site

Example Multiple Screen Session



If mistake
ask again
Order Info
Present form asking for product and quantity.
Retain product and quantity
If mistake remember all information entered & ask again.
Billing Info
Present form requesting name and billing code.
Retain product, quantity, name, and billing code.
Save and Notify
Save to a file or database and send email.
Thank the end-user

How to maintain the state through multiform?

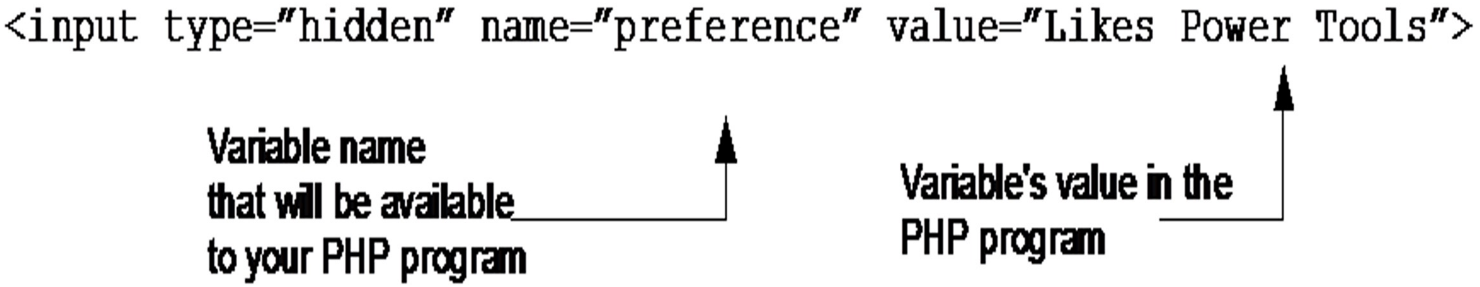
- Use tricks to keep track of state information between requests (session tracking)
  - Using hidden form fields
  - URL rewriting: every local URL on which the user might click is dynamically modified to include extra information
  - Using cookies: a bit of information that the server give to a client
  - Using session

Content

1. Hidden fields
2. User browser cookies
3. PHP session

1. Hidden fields

- Hidden fields are part of HTML forms
  - Not displayed but value can be accessed in receiving script like any other variable.
  - Can still be viewed by user’s who view source.



```
<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

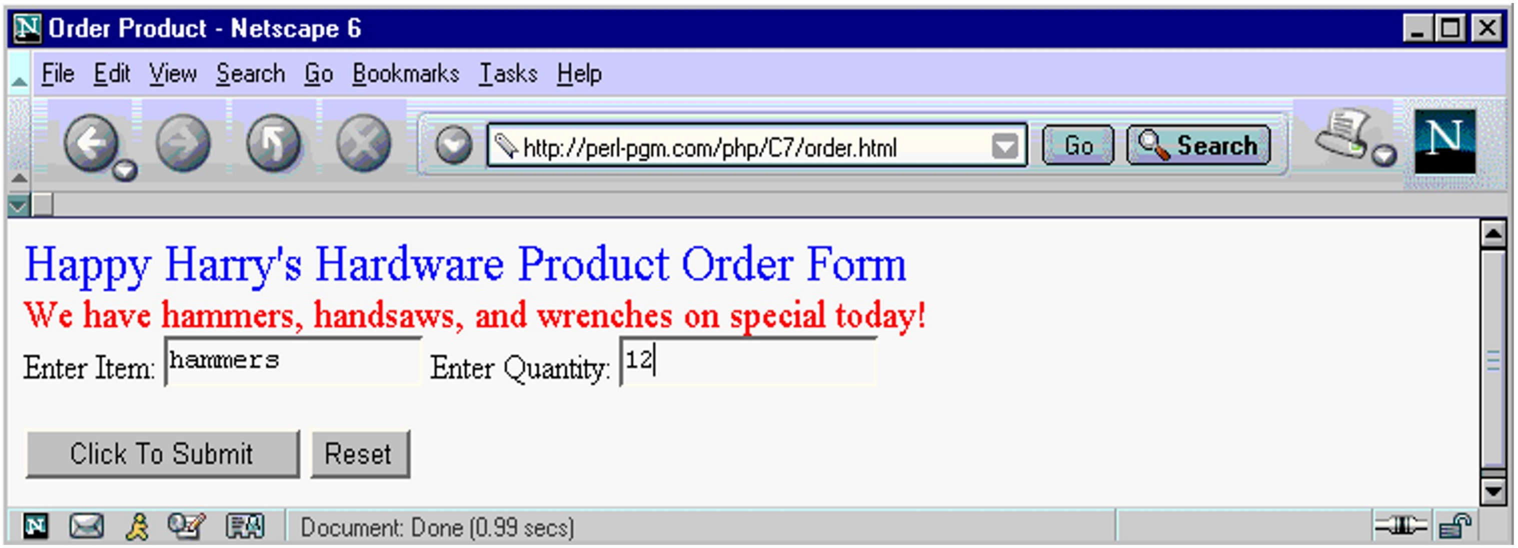
An example

- Consider an example script sets a hidden field
  - Implements the Order Info form e.g., order.html
  - On submit sends data to other pages e.g., order2.php, order3.php

order.html

```
<!DOCTYPE html><html><head><title>Order Product</title></head><body>
<form action="order2.php“ method="post">
<h1> Hardware Product Order Form</h1>
<p class="highlight">
We have hammers, handsaws, and wrenches on special today!
</p>
<input type="hidden" name="sample_hidden" value="Welcome"><br/>
Enter Item: <input text type="text" size="15” maxlength="20" name="product">
Enter Quantity: <input text type="text" size="15” maxlength="20" name="quantity"><br/><br/>
<input type="submit" value="Click To Submit">
<input type = "reset" value="Reset">
</form></body></html>
```

The Output


Order Product - Netscape 6
File Edit View Search Go Bookmarks Tasks Help
http://perl-pgm.com/php/C7/order.html
Go
Search

Happy Harry's Hardware Product Order Form

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

Enter Item: hammers Enter Quantity: 12

Click To Submit
Reset
Document: Done (0.99 secs)

order2.php

- Your scripts can receive data from hidden fields like any other data.

```
<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 =
$_POST[“quantity”];
print "<p class=“highlight”>";
print "Hidden value=$sample_hidden </p><br>";
print "You selected product=$product and
quantity=$quantity";
```

order2.php

```
print "<br><br><input type=\"hidden\" name=\"product\“ value=\"$product\"> ";
print "<input type=\"hidden\" name=\"quantity\“ value=\"$quantity\">";
print "<input type=\"hidden\" name=\"sample_hidden\"value=\"$sample_hidden\">";
print 'Please enter your name:';
print '<input type="text" size="15" maxlength="20“ name="name">';
print ' and billing code: (5 digits)';
print '<input type="text" size="5" maxlength="5” name="code">';
print ‘<br/> <input type=submit value="Process Order">';
print '<input type=reset>';
?></form></body></html>
```

Content

1. Hidden fields
2. User browser cookies
3. PHP session

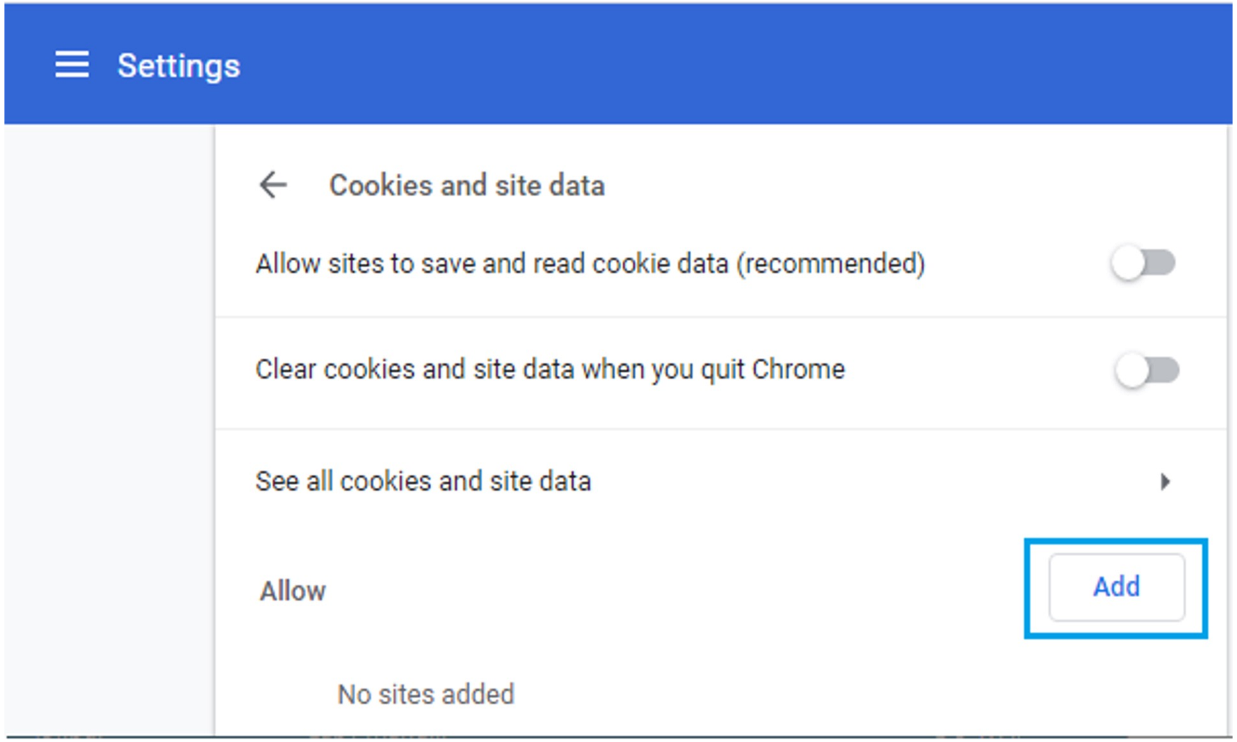
Using Browser Cookies ...

- 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

Understanding Cookie Limitations

- Users can easily disable the cookies feature.
- Users may delete cookies.
- PHP sets limit on cookies

The disable cookie screen



Settings

Cookies and site data

Allow sites to save and read cookie data (recommended)

Clear cookies and site data when you quit Chrome

See all cookies and site data

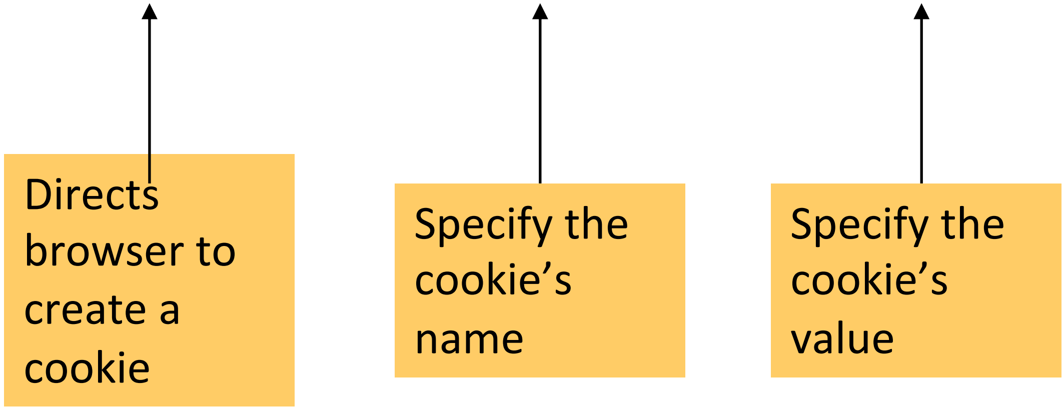
Allow

No sites added

Setting and Reading Cookies

- A cookie is often 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.

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



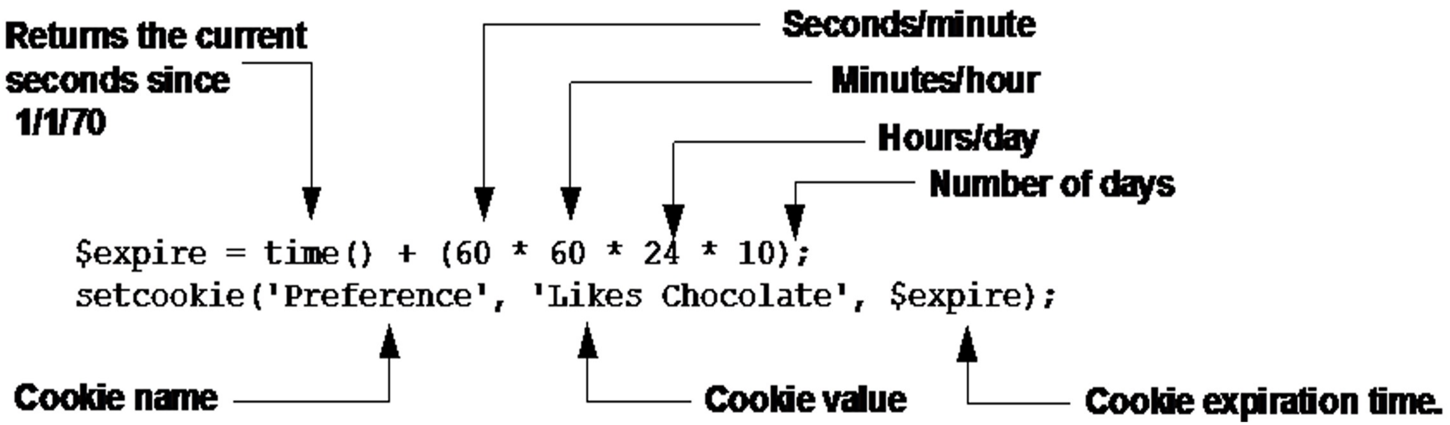
Directs browser to create a cookie

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.



```
$expire = time() + (60 * 60 * 24 * 10);
setcookie('Preference', 'Likes Chocolate', $expire);
```

A full example of setting a cookie....

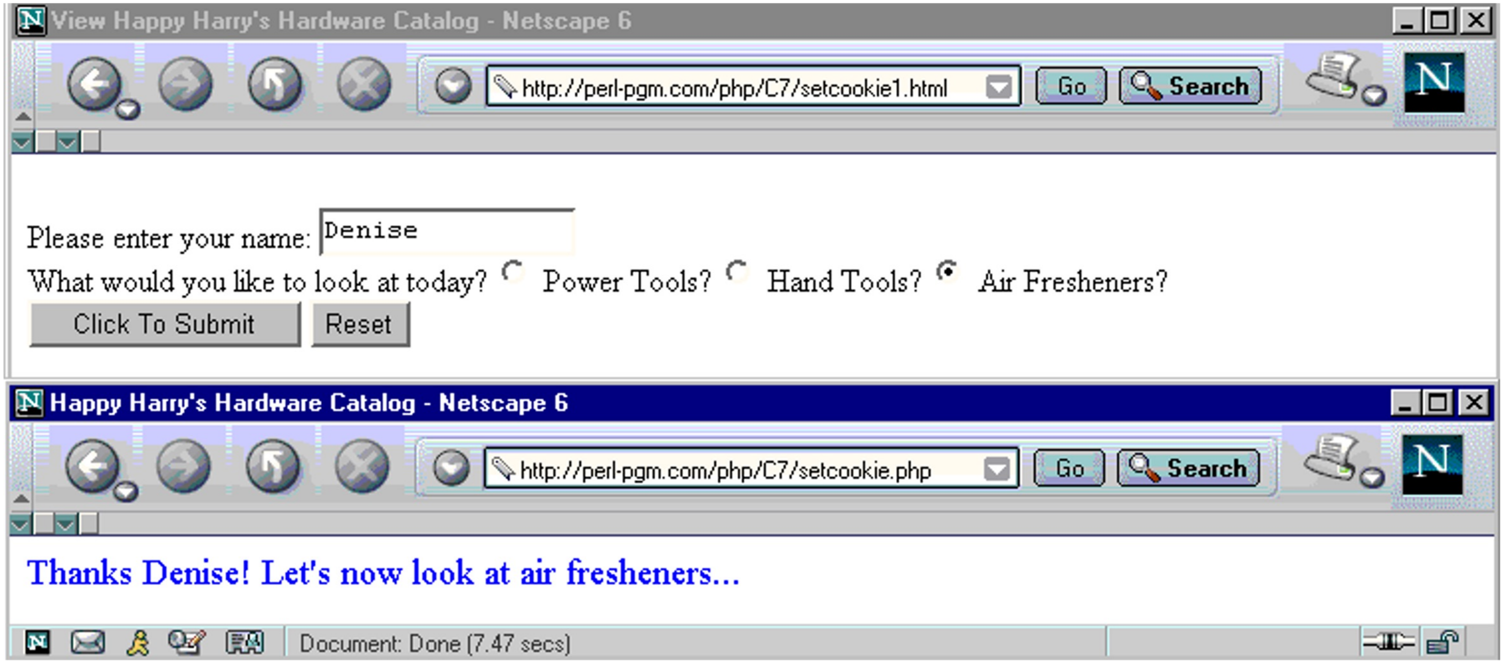
- Suppose a front-end web page asks for some survey information:

```
<input type="text" size="15" maxlength="20"
  name="custname">
<input type="radio" name="prefers" value="power tools"
  checked > Power Tools?
<input type="radio" name="prefers" value="hand tools">
  Hand Tools?
<input type="radio" name="prefers″ value="air
  fresheners"> Air Fresheners?
```

The script runs when submitted setcookie.php

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

Would output:



View Happy Harry's Hardware Catalog - Netscape 6

http://perl-pgm.com/php/C7/setcookie1.html

Please enter your name: Denise

What would you like to look at today? ○ Power Tools? ○ Hand Tools? ● Air Fresheners?

Happy Harry's Hardware Catalog - Netscape 6

http://perl-pgm.com/php/C7/setcookie.php

Thanks Denise! Let's now look at air fresheners...

Document: Done (7.47 secs)

Reading Cookies

- To read a cookie value use the $_COOKIE[] associative array to get the cookie function
- $cust_name= $_COOKIE[“cust_name”];

Example Script that read a cookie

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

Content

1. Hidden fields

2. User browser cookies

3. PHP session

PHP Sessions

- PHP enables you to retain data between forms
  - session_start()
    - starts a new session
    - This function must be the very first thing in your document. Before any HTML tags.

  - Global varialbe $_SESSION
    - $_SESSION[‘name ‘]=$name;

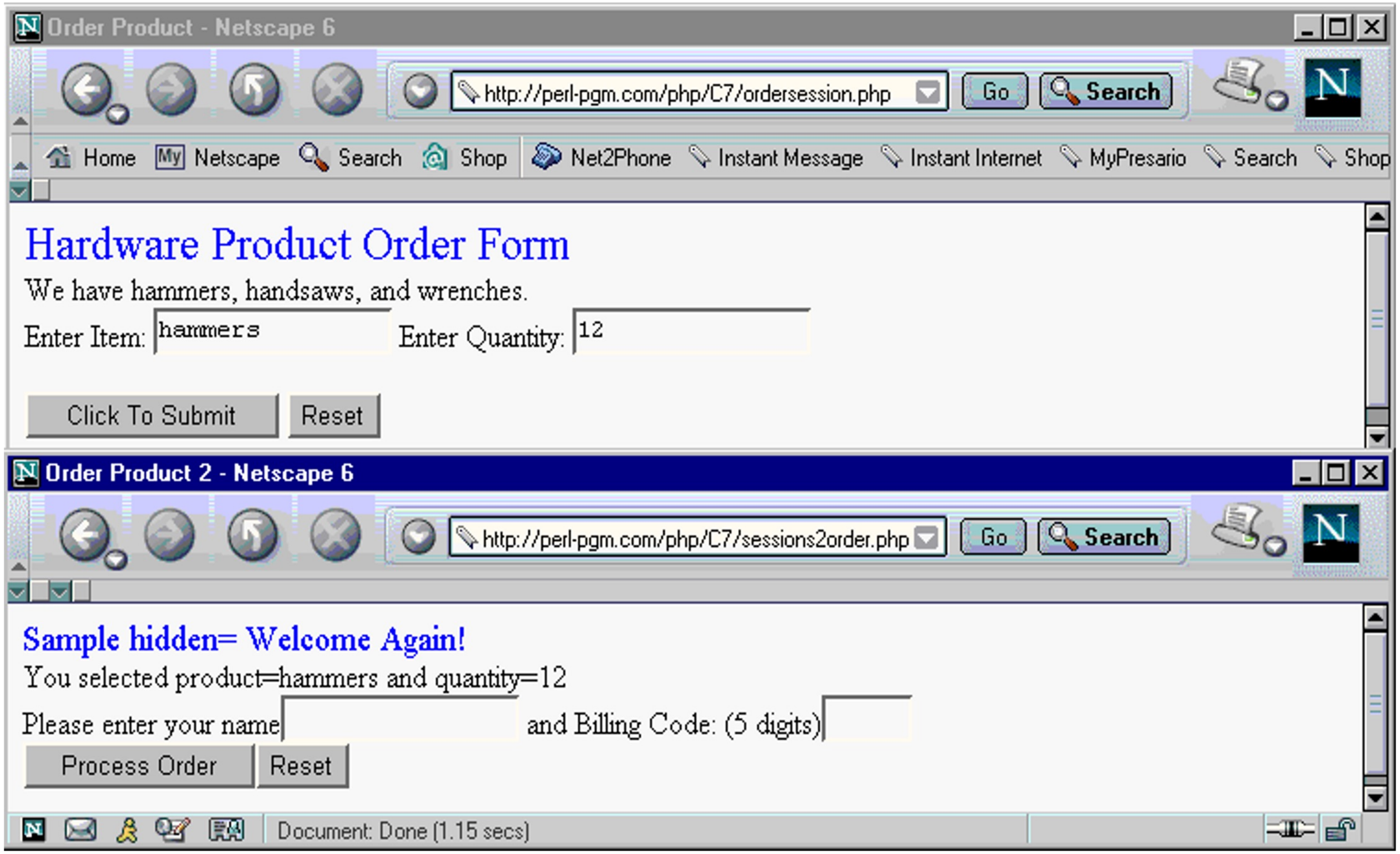
Example PHP Code

```
<?php session_start(); ?>
<html>
<head><title>Order Product</title></head>
<body>
   <form action="session.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" size="15" maxlength="20" name="product">
   Enter Quantity: <input text type="text" size="15" maxlength="20"
   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>
```

sessions.php

```
<?php session_start();?>
<html><head><title> Order Product 2 </title> </head>
<body>
<form action="session2.php" method="post">
<?php
   $sample_hidden = $_SESSION["sample_hidden"];
   print "<h1> Sample hidden= $sample_hidden</h1>";
   $product = $_POST["product"];
   $quantity = $_POST["quantity"];
   print "<br>You selected product=$product and quantity=$quantity";
   $_SESSION["product"] = $product;
   $_SESSION["quantity"] = $quantity;
   print '<br>Please enter your name';
   print '<input text type="text" size="15" maxlength="20"
   name="name">';
   print ' and Billing Code: (5 digits)';
   print '<input text type="text" size="5" maxlength="5" name="code">';
   print '<br> <input type=submit value="Process Order">';
   print '<input type=reset>’;
?>
```

Example output



Order Product - Netscape 6
http://perl-pgm.com/php/C7/ordersession.php
Go
Search
Home
Netscape
Search
Shop
Net2Phone
Instant Message
Instant Internet
MyPresario
Search
Shop

Hardware Product Order Form

We have hammers, handsaws, and wrenches.

Enter Item: hammers Enter Quantity: 12

Click To Submit Reset

Order Product 2 - Netscape 6
http://perl-pgm.com/php/C7/sessions2order.php
Go
Search

Sample hidden= Welcome Again!

You selected product=hammers and quantity=12

Please enter your name and Billing Code: (5 digits)

Process Order Reset

Document: Done (1.15 secs)

Some session extras

- isset- can be used to determine if a variable comes from a session:

```
if (isset($_SESSION['name’])){
    print "got name=$name from session";
} else {
    print "name=$name not set from session";
}
```

Summary

- Hidden fields are HTML form fields you can use to set a variable name and variable value without displaying them on a form.
- Cookies provide a way for Web server applications to store small pieces of data on the user's hard disk.
- PHP provides a convenient way to retain data between PHP scripts.
  - Use session_start() and $_SESSION to start sessions and define session variables, respectively

Question?