Maintaining State in Web Applications

pluralsight
hardcore developer training

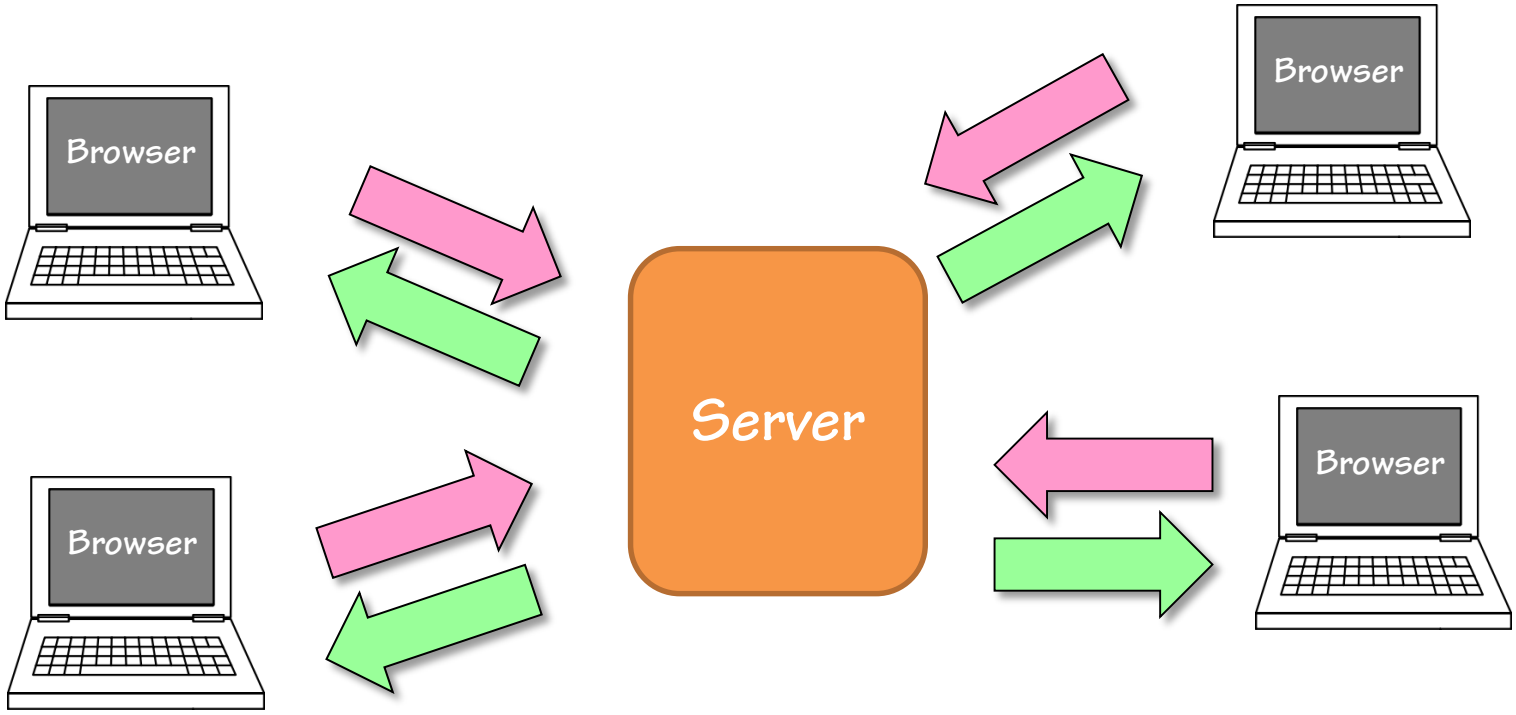
Lesson Objectives

- **In this lesson we will learn what we mean by "state" and why maintaining state is an issue**

- **We'll examine several approaches:**
  - The PHP **`$_SESSION`** object
  - Hidden form fields
  - Cookies

- **We will develop a "shopping cart" for our library so that borrowers can reserve books**

How web applications work!



This doesn't work …

- You cannot simply persist data in variables within your PHP code

```
<body>
<FORM action="counter-global.php" method="GET">
<INPUT type="submit" name="Count" value="Count">
<?php
@   $count = $count + 1;
    echo "count is $count";
?>
</FORM>

</body>
```

Variable is created and initialised to zero every time

Introducing Sessions

- **A *session* is a sequence of related interactions with a web application**
  - Has a specific goal (get a local weather forecast, book a flight, …)
  - Ends when the browser is closed
  - Can be explicitly destroyed

- **PHP makes it easy to maintain state information througout a session, using the `$_SESSION` array**
  - Any data placed into this array is automatically persisted

- **Let's use `$_SESSION` to implement a hit counter that actually works!**

Using the $_SESSION Array

- **$_SESSION is an associative array:**

```
<body>
<form action="counter-session.php" method="GET">
<input type="submit" name="Count" value="Count">
<?php
   session_start();
   if (! isset($_SESSION['counter']))
      $count = 0;
   else
      $count = $_SESSION['counter'];
   $count = $count + 1;
   $_SESSION['counter'] = $count;
   echo "count is $count";
?>
</form>
</body>
```

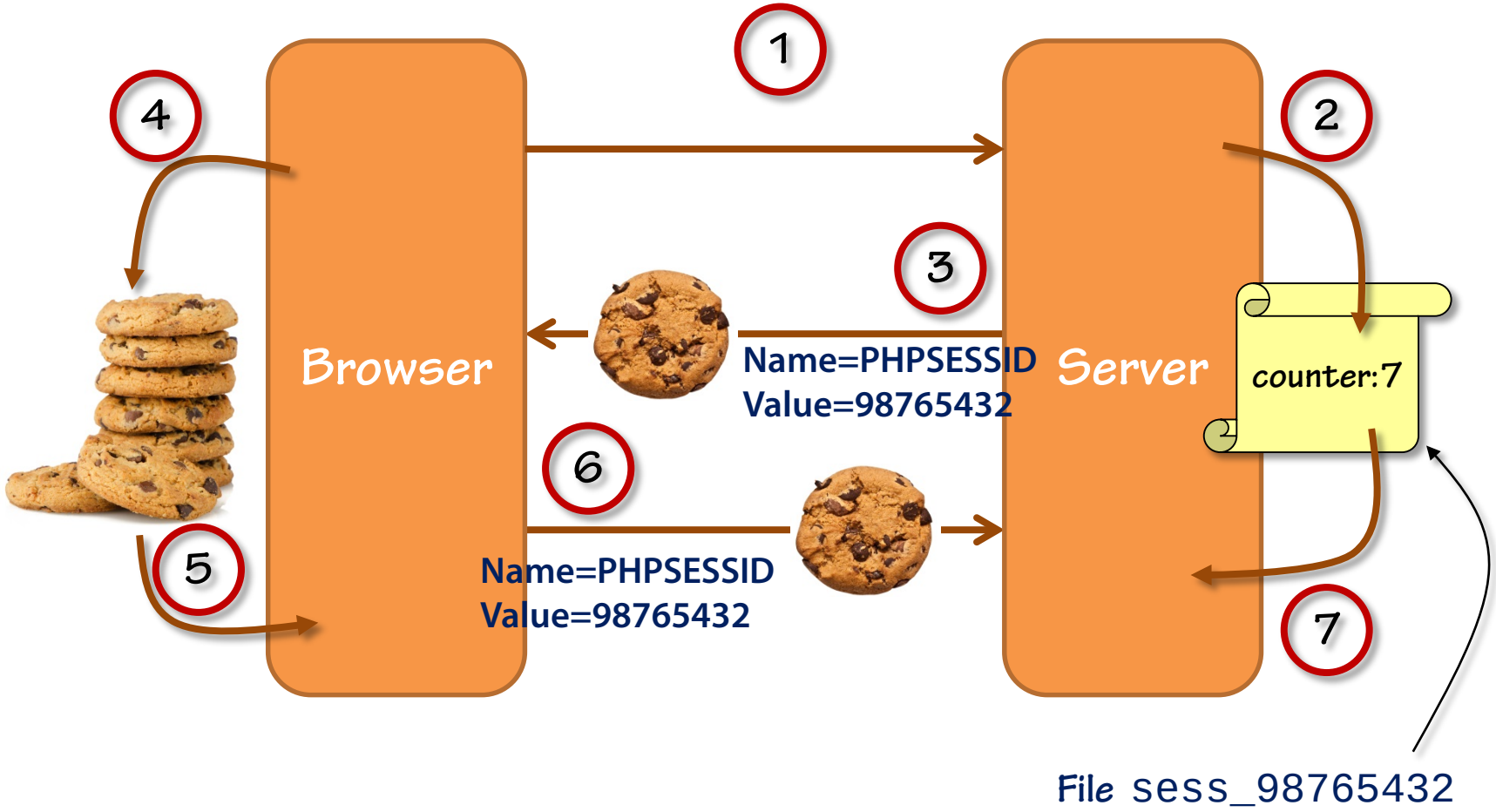
Establish new session or populate array From existing session

First time

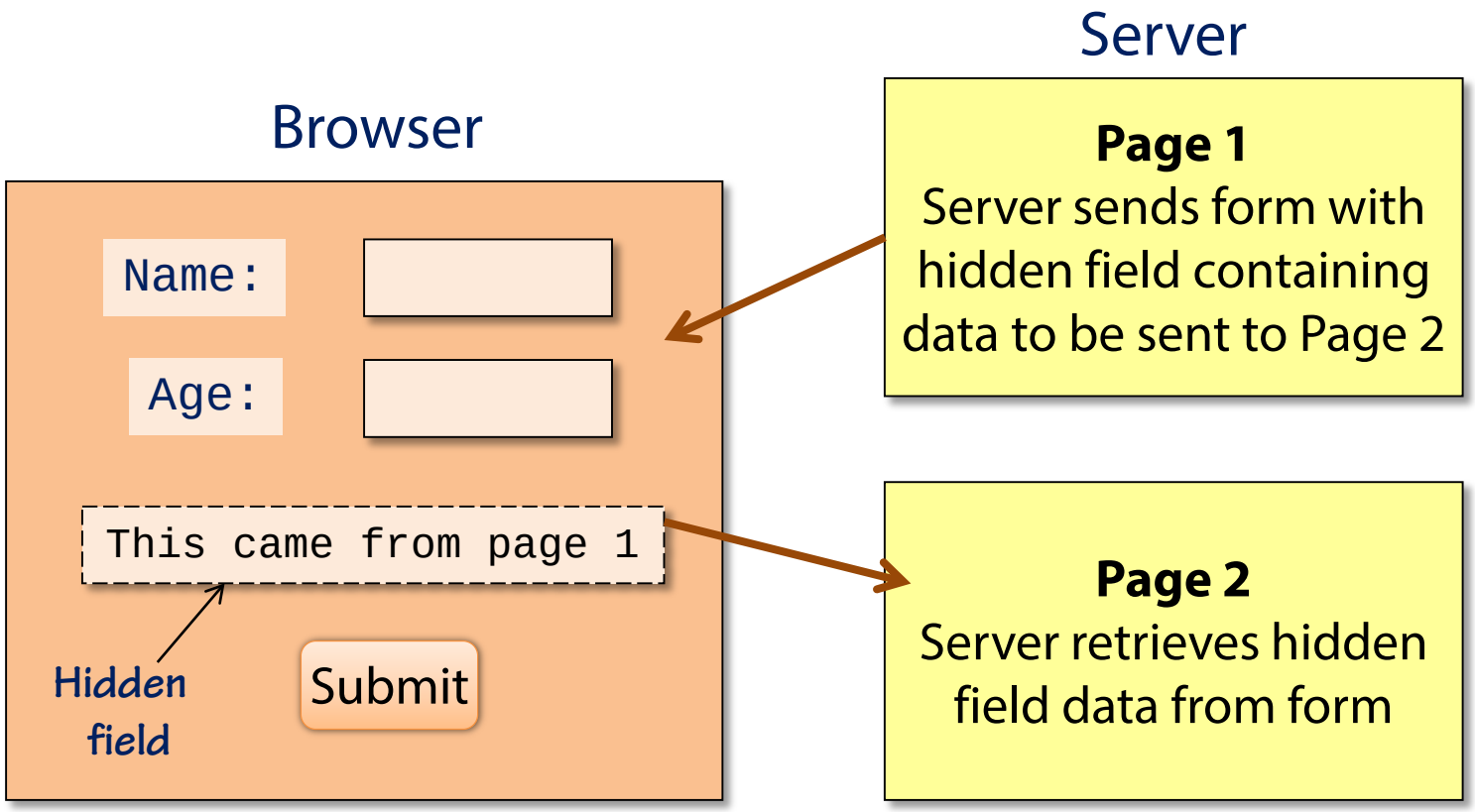
Get data from session

Put updated data into the session

How Sessions Work



Hidden Form Fields



Using Hidden Form Fields

```
<body>
  <form action="counter-hidden.php" method="GET">
  <input type="submit" name="Count" value="Count">
  <?php
   if (! isset($_GET['hiddencounter']))
      $count = 0;
   else
      $count = $_GET['hiddencounter'];
   $count = $count + 1;
   echo '<input type="hidden" value=' . $count .
          'name="hiddencounter">';
   echo "count is $count";
  ?>
  </form>
</body>
```

Cookies

- **A cookie is an item of information created on the server and sent to the browser for safe keeping**
- **The browser sends the cookie back to the server when it visits a related page on the site**
- **Cookies are transmitted using fields in the HTTP request and response headers**

```
Set-Cookie: counter=5
```

- **Session cookies**
  - Short term, destroyed when a browsing "session" ends
- **Persistent cookies**
  - Long term, stored in browser's file system, persist data between sessions

What's in a Cookie?

- **The attributes of a cookie include:**

Attribute	Description
`name=value`	A programmer-defined name and associated value
`expires`	The time and date when the browser should forget the cookie. If not set, it's a session cookie
`domain`	The domain name (web site) that the cookie relates to
`path`	The browser will send the cookie only for pages at (or below) this path
`secure`	If set, the browser will only send the cookie if it's using a secure (SSL) connection

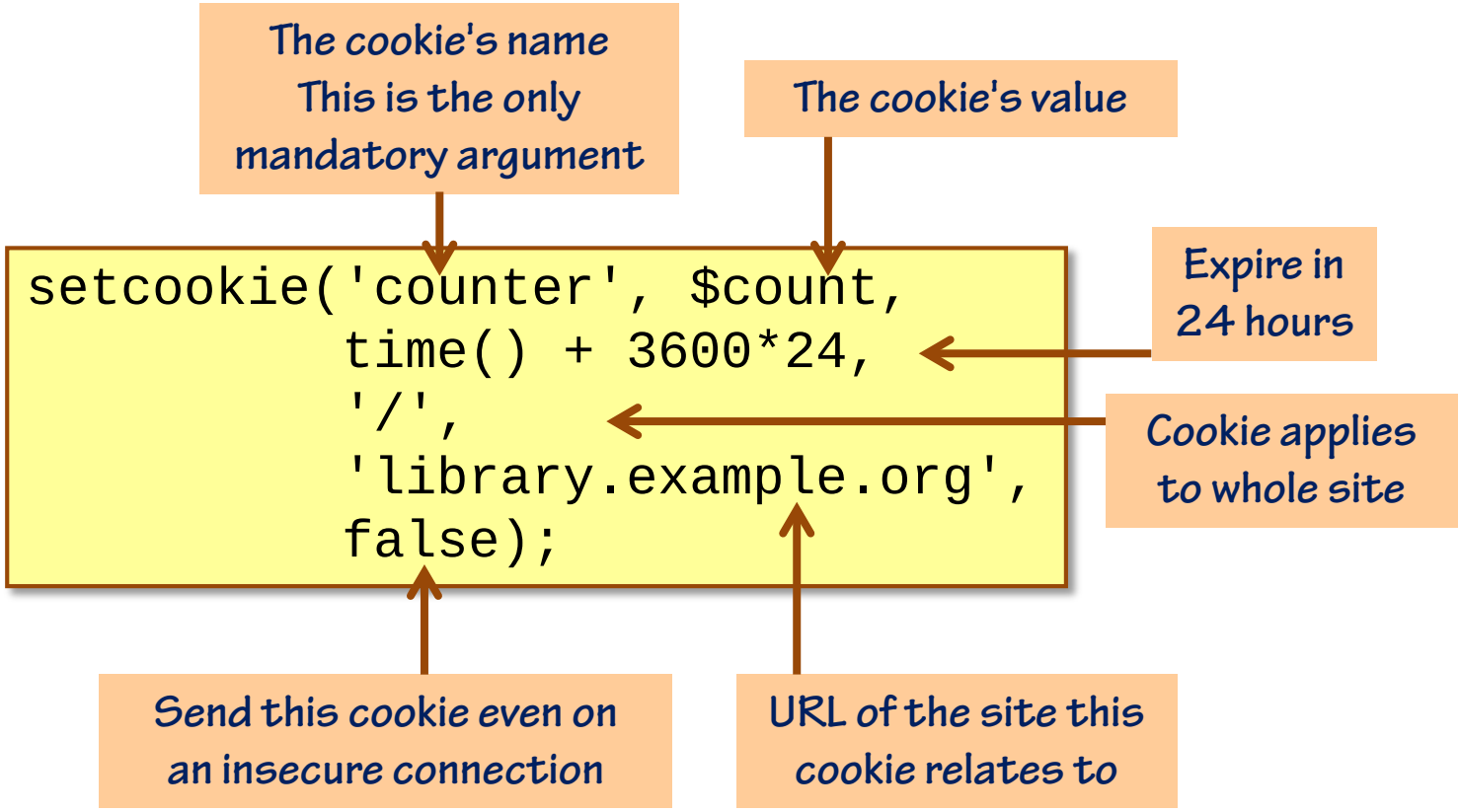
Sample Cookie

HTTP response header

```
Set-Cookie: counter=5; domain=library.example.org;
path=/; expires=Tue 1 Jan 2016 00:00:01 2036 GMT
```

Creating a Cookie

- Using PHP, create a cookie using `setcookie()`
- Example:



Retrieving a Cookie

- **PHP retrieves the cookies sent to a page and places them into the `$_COOKIE` array**
  - Cookie name is array index
  - Also available in $_REQUEST

- **Example:**

```
if (isset($_COOKIE['counter']))
    $count = $_COOKIE['counter'];
```

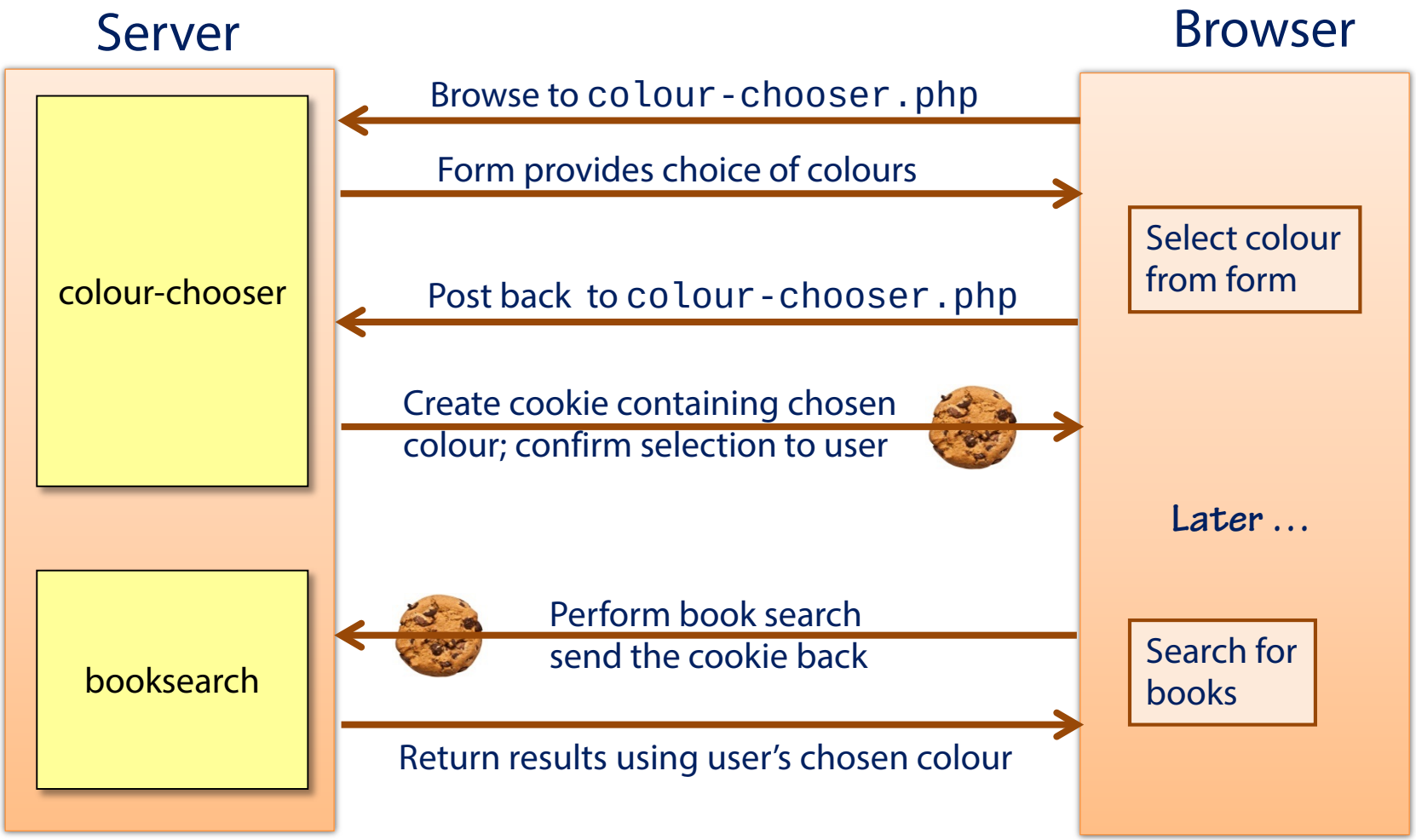
The Library Case Study

- **We 'll use what we've learned to implement two new features in our library:**

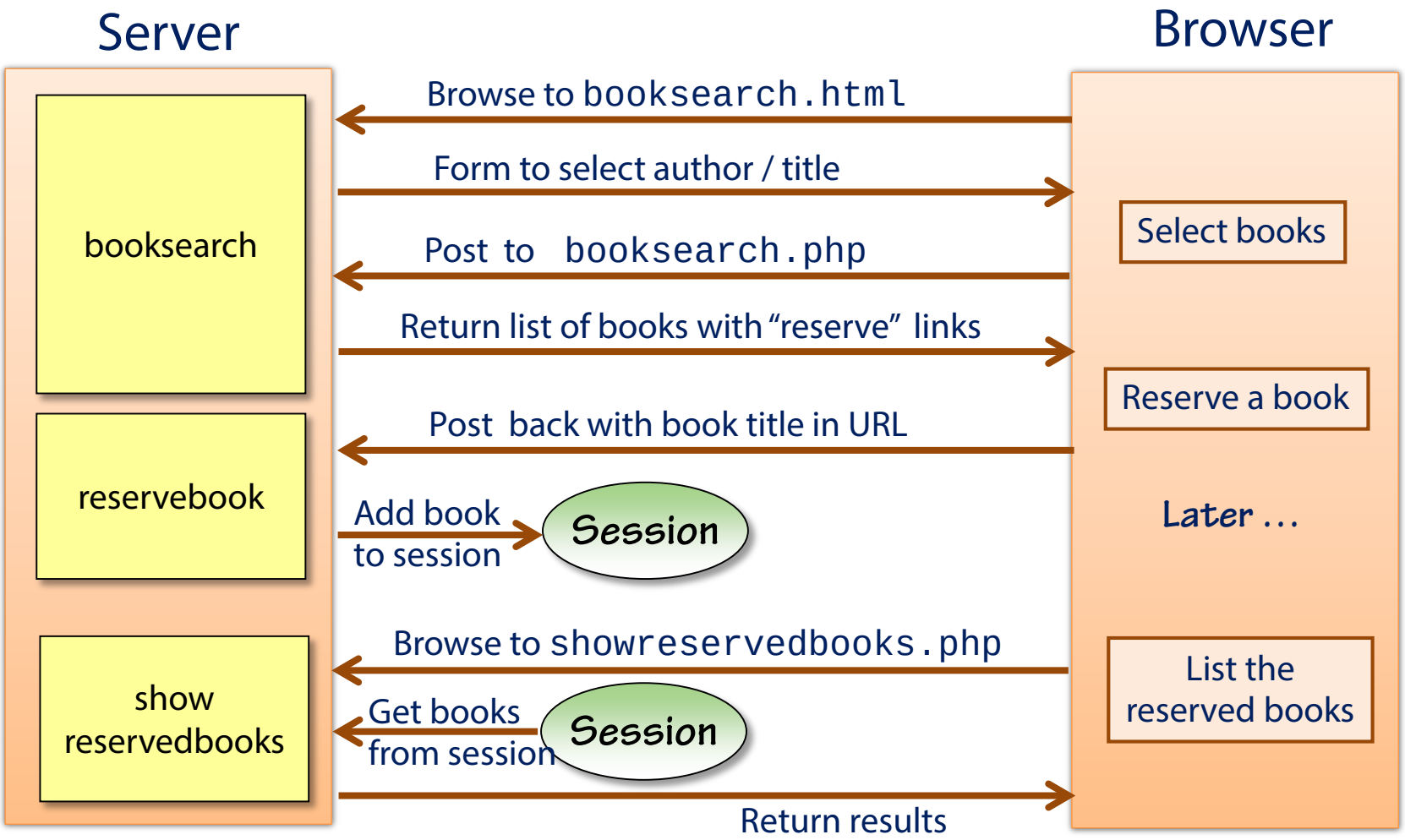
- **"Permanent" storage of user preferences**
  - Colour preference stored in a cookie

- **A "shopping cart" of reserved books**
  - Using a PHP Session

Workflow for Colour Preference Selection

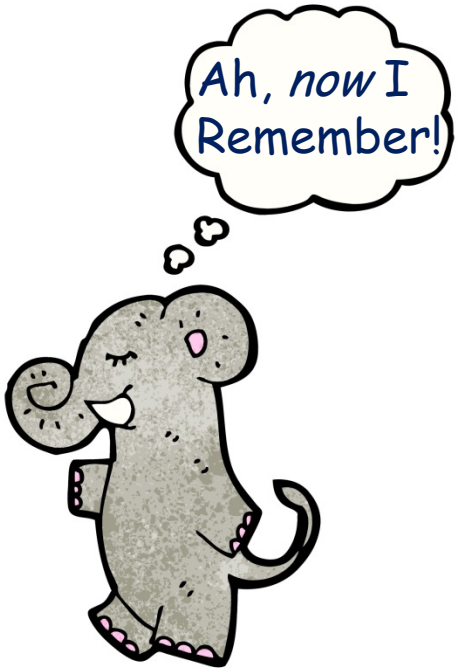


Workflow for Book Reservation List



Lesson Summary

- **We have looked at three ways of maintaining state in a web application**
  - PHP sessions
  - Hidden fields
  - Cookies
- **We have applied these techniques to our library case study**
  - Storing user preferences
  - Building a "shopping cart"



Coming up in Lesson 8:

The access control model in MySQL

Creating user accounts

Granting and revoking privileges