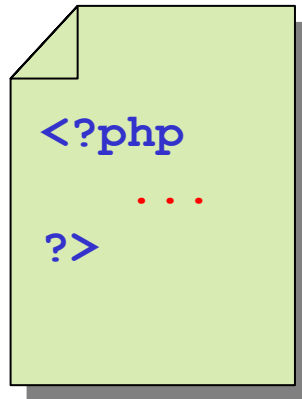


COSC 2206 Internet Tools

More PHP Web Applications





More PHP WEB applications

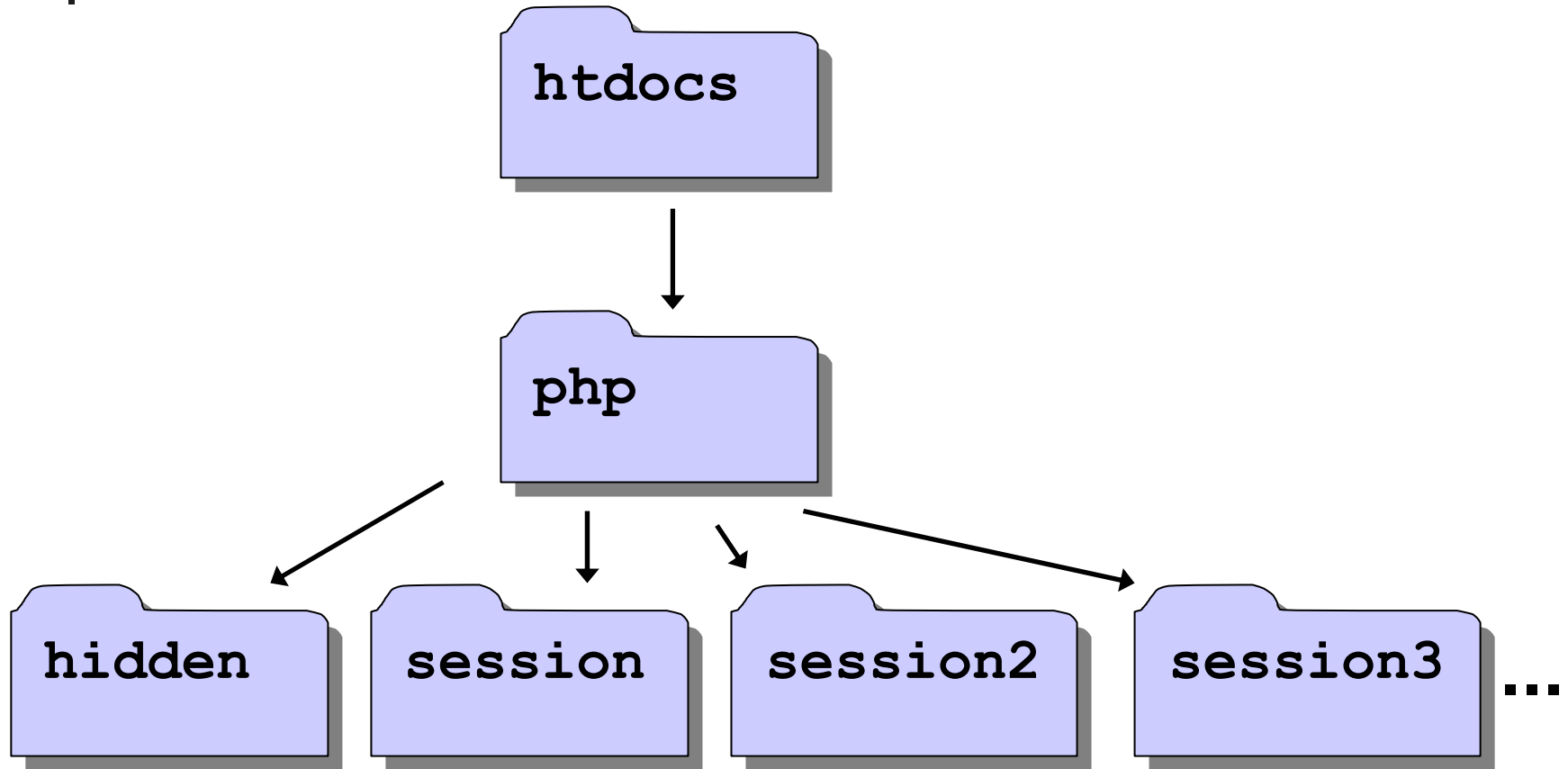
Hidden fields

Survey Example

Session Management

Authentication

Web directory structure





Hidden Fields (1)

- Web server/client communication is stateless
- When a client makes an HTTP request to a server, the server makes an HTTP response, delivering a document for example, and then closes the connection and forgets who the client was.
- A very simple way to maintain state across multiple connections is to use hidden fields



Hidden fields (2)

- This method only works across a chain of forms (submission of one form followed by another form and so on)
- Use the hidden input field of the form to save state information

```
<input type="hidden" name="name" value="value">
```

- It is not displayed but its name-value pair is submitted with the form data.

Example with two forms

Page 1

Name:

Age:

**Page 1 has
two fields**

Page 2

Favourite book:

**Page 2 has
one field and
two hidden fields
from first form**

Results

Name	Fred
Age	37
Favourite book	Dune

**Page 2
remembers the
results from Page
1 so results of all
three fields can
be displayed**



hidden_demo.php (1)

```
<?php

if ( isset($_REQUEST['page1']) )
{
    // First form submitted so display page 2
    display_page_two();
}
elseif ( isset($_REQUEST['page2']) )
{
    // Second form submitted so display results
    display_result_page();
}
else // no forms submitted so display page 1
{
    display_page_one();
}
```



hidden_demo.php (2)

```
function display_page_one ()
{
    ?>

    <html>
    <head><title>Page 1</title></head>
    <body>
        <h1>Page 1</h1>
        <form>
            <table>
                <tr>
                    <td>Name: </td>
                    <td><input type="text" name="name"></td>
                </tr>
            </table>
        </form>
    </body>
</html>
```




hidden_demo.php (3)

```
<td>Age: </td>
<td><input type="text" name="age"></td>
</tr>
<tr>
<td><input type="submit" name="page1"
value="Next Page"></td>
</tr>
</table>
</form>
</body>
</html>
<?php
}
```



hidden_demo.php (4)

```
function display_page_two ()
{
    // get the values just submitted from form one
    $name = $_REQUEST['name'];
    $age = $_REQUEST['age'];

    // insert them in form 2 as hidden fields
    ?>
    <html>
    <head><title>Page 2</title></head>
    <body>
    <h1>Page 2</h1>
```



hidden_demo.php (5)

```
<form>
<input type="hidden" name="name"
    value="<?php echo $name ?>">
<input type="hidden" name="age"
    value="<?php echo $age ?>">
<table>
<tr><td>Favourite book: </td>
    <td><input type="text" name="book"></td>
</tr>
<tr><td><input type="submit" name="page2"
    value="Show Results"></td>
</tr>
</table></form><body></html>

<?php
}
```



hidden_demo.php (6)

```
function display_result_page ()
{
    // Form 2 was just submitted so get results
    // for all fields

    $name = $_REQUEST['name']; // hidden field
    $age = $_REQUEST['age'];    // hidden field
    $book = $_REQUEST['book'];

    ?>
```



hidden_demo.php (7)

```
<html>
<head><title>Results</title></head>
<body><h1>Results</h1>
<table border="1">
  <tr><td>Name</td><td>
    <?php echo $name ?></td></tr>
  <tr><td>Age</td><td>
    <?php echo $age ?></td></tr>
  <tr><td>Favourite book</td><td>
    <?php echo $book ?></td></tr>
</table>
</body>
</html>
<?php
}
```

[view source hidden/hidden_demo.php](http://localhost/php/hidden/hidden_demo.php)
http://localhost/php/hidden/hidden_demo.php



Survey Example

- Several linked forms make up the survey
- Some forms are conditional (form depends on previously submitted answers)
- [view script hidden/survey.php](#)
- <http://localhost/php/hidden/index.html>
- Things to do:
 - save survey results in a file or database
 - write scripts to do statistics using this file



Session Management

- Session management keeps track of users across multiple pages, in an online shopping application for example
- There are three ways to do session management
 - hidden fields (restricted to set of linked forms)
 - cookies (general method that works unless user does not accept cookies)
 - URL rewriting using a session id in URL



What is a session?

- A session is a set of connections that a specific user makes to a server.
- For example in a shopping cart application each item in the cart is entered in a separate connection.
- The requests that a shopper makes in browsing the catalog, filling the cart and checking out constitute a session



PHP session management (1)

- `session_start()`
 - call this function at top of each page before any data is sent to the browser. This will create a session if one doesn't exist.
- `session_destroy()`
 - call this function to terminate a session
- By default, cookies are valid until the browser is closed.
- Longer times can be set in the `php.ini` file.



PHP session management (2)

- Each session user receives a 32 hex digit id which is associated with the name PHPSESSID (cookie name or query name)
- The value is available in PHP using the function call `session_id()`.



PHP session variables (1)

- `$_SESSION['name'] = $value ;`
 - In the latest versions of PHP this is used to set a session variable and give it a value that will be remembered throughout the session
- `unset ($_SESSION['name']) ;`
 - In the latest versions of PHP use this to unregister a session variable: remove it from the `$_SESSION` array



PHP session variables (2)

- Session variables are stored in files.
- You must specify the default directory in the `php.ini` file. For example, I use
 - `session_save_path="c:\apache\sessions"`
- `session_save_path()`
 - returns the current path
- `session_save_path(path)`
 - sets the save path to the given path



Example

- The following example consists of three pages.
 - **page1.php**: start a session and set a session variable
 - **page2.php**: continue session but unset the session variable
 - **page3.php**: try to display session variable and destroy the session
 - <http://localhost/php/session/page1.php>
 - (must reload page when using Power Point)

session/page1.php

```
<?php
    session_start();
    $_SESSION['greeting'] = "Hello World";
?>
<html>
<head><title>Session Management Page 1</title></head>
<body><h1>Session Management Page 1</h1>
<p>Value of the session variable is
    <?php echo $_SESSION['greeting'] ?></p>
<p>The session id is <?php echo session_id() ?></p>
<p>The session save path is
    <?php echo session_save_path() ?></p>
<p><a href="page2.php?<?php echo SID ?>">
    Go to page 2</a></p></body></html>
```

■ [view script session/page1.php](#)

URL rewriting



session/page2.php

```
<?php
session_start();
$greeting = $_SESSION['greeting'];
unset($_SESSION['greeting']);
?>
<html>
<head><title>Session Management Page 2</title></head>
<body><h1>Session Management Page 2</h1>
<p>Value if the session variable is
    <?php echo $_SESSION['greeting']?></p>
<p><a href="page3.php?<?php echo SID ?>">
Go to page 3</a></p>
</body>
</html>
```

■ [view script session/page2.php](#)



session/page3.php

■ [view script session/page3.php](#)

```
<?php
session_start();
if ( isset($_SESSION['greeting']) )
    $result = "Value of session variable is
               $_SESSION['greeting']";
else
    $result =
        "The session variable is no longer set";
?>
<html>
<head><title>Session Management Page 3</title></head>
<body><h1>Session Management Page 3</h1>
<p><?php echo $result?></p>
</body></html>
```




Session counter example

- Each time the script page is accessed a counter is incremented and displayed
- The count is remembered from one connection to the next until the browser is closed (the default)



session_counter.php (1)

```
<?php
session_start();
if (! isset($_SESSION['count']) )
{
    $_SESSION['count'] = 0;
    $_SESSION['start'] = time();
}
$_SESSION['count']++;
$session_time = time() - $_SESSION['start'];
$cookie_value = isset($_COOKIE['PHPSESSID']) ?
    $_COOKIE['PHPSESSID'] : '';
?>
```



session_counter.php (2)

```
<html>
<head><title>Session Counter Example</title></head>
<body>
<h1>Session Counter</h1>
<p>The session id is <?php echo session_id() ?></p>
<p>The cookie name is <?php echo $cookie_value?></p>
<p>The count is <?php echo $_SESSION['count'] ?></p>
<p>Session has been active for <?php echo
    $session_time?> seconds</p>
<p>The session save path is
    <?php echo session_save_path() ?></p>
<p><a href="session_counter.php?<?php echo SID?>">
    load page again</a></p>
</body>
</html>
```



Session counter example

- The following version times out after 30 seconds and destroys the session.
- A reload will now begin a new session but using the same session id

session_counter2.php (1)

```
<?php
session_start();
if (! isset($_SESSION['count']) )
{
    $_SESSION['count'] = 0;
    $_SESSION['start'] = time();
}
$_SESSION['count']++;
$session_time = time() - $_SESSION['start'];
$cookie_value = isset($_COOKIE['PHPSESSID']) ?
    $_COOKIE['PHPSESSID'] : '';

if ($session_time > 30)
    session_destroy();

?>
```



session_counter2.php (2)

```
<html>
<head><title>Session Counter Example</title></head>
<body>
<h1>Session Counter</h1>
<p>The session id is <?php echo session_id() ?></p>
<p>The cookie name is <?php echo $cookie_value?></p>
<p>The count is <?php echo $_SESSION['count'] ?></p>
<p>Session has been active for <?php echo
    $session_time?> seconds</p>
<p>The session save path is
    <?php echo session_save_path() ?></p>
<p><a href="session_counter2.php?<?php echo SID?>">
    load page again</a></p>
</body>
</html>
```



Cookies / URL rewriting

- If cookies are allowed PHP will use a cookie, otherwise URL rewriting is used
- To see this set explorer security preferences for local access in local internet security and set cookies for prompt
- Now open browser for page1.php, answer no to cookies to see on page2.php a URL like
`http://localhost/php/session/page2.php?PHPSESSID=772bd400b802f81d8dfed45f577e2720`



Another example

- In this example a session variable is used to hold the user's background color preference.
 - [page1.php](#): use a form to get user's color preference or show first page if color has already been specified
 - [page2.php](#): show a page then unregister the session variable
 - [page3.php](#): show a page and destroy session
 - <http://localhost/php/session2/page1.php>
 - must reload page when using PowerPoint



session2/page1.php (1)

```
<?php
// ini_set("session.cookie_lifetime", "30000");
session_start();
if (isset($_REQUEST['bgcolor']))
{
    $_SESSION['bgcolor'] = $_REQUEST['bgcolor'];
    display_page_one();
}
else
{
    display_form_page();
}
?>
```



session2/page1.php (2)

```
<?php
function display_form_page ()
{   $self = $_SERVER['PHP_SELF'];
?>
<html>
<head><title>Session bgcolor page 1</title></head>
<body><h1>Session bgcolor page 1</h1>
<form method="post"
    action="<?php echo $self?>?<?php echo SID?>">
Enter color code as <code>#HHHHHH</code>:
<input type="text" name="bgcolor" value="">
<input type="submit" value="Submit Color">
</form>
</body></html>
<?php } ?>
```



session2/page1.php (3)

```
<?php
function display__page_one ()
{
?>
<html>
<head><title>Session bgcolor page 1</title></head>
<body bgcolor="<?php echo $_REQUEST['bgcolor'] ?>">
<h1>Session bgcolor page 1</h1>
Background color is <?php echo $_SESSION['bgcolor'] ?>
<p><a href="page2.php?<?php echo SID ?>">
Go to page 2</a></p>
</body></html>
<?php } ?>
```

- [view script session2/page1.php](#)



session2/page2.php

■ [view script session2/page2.php](#)

```
<?php
    session_start();
    $bgcolor = isset($_SESSION['bgcolor']) ?
        $_SESSION['bgcolor'] : "#FFFFFF";
    unset($_SESSION['bgcolor']);
?>
<html>
<head><title><Session bgcolor page 2</title></head>
<body bgcolor="<?php echo $bgcolor ?>">
<h1>Session bgcolor page 2</h1>
<?php echo "Background color is $bgcolor" ?>
<p><a href="page3.php?<?php echo SID ?>">
Go to page 3</a></p>
</body>
</html>
```



session2/page3.php

```
<?php
    session_start();
    $bgcolor = isset($_SESSION['bgcolor']) ?
        $_SESSION['bgcolor'] : #FFFFFF;
    session_destroy();
?>
<html>
<head><title><Session bgcolor page 2</title></head>
<body bgcolor="<?php echo $bgcolor ?>">
<h1>Session bgcolor page 2</h1>
<?php echo "Background color is $bgcolor" ?>
</body>
</html>
```

■ [view script session2/page3.php](#)



User Authentication

- Browser displays login dialog box.
 - When browser requests a page server checks headers for a username and password.
 - If none, it sends an authenticate response asking the browser to send a user name and password. Browser uses login dialog
 - PHP can use this name and password to authenticate the user
- Let PHP do the login on a web page
 - server uses form to get name and password



Example

- When a request is made by browser the server needs to send the headers
 - WWW-Authenticate: Basic realm="..."
 - HTTP/1.0 401 Unauthorized
- The browser will get a username and password and send it in the headers
- PHP uses `$_SERVER['PHP_AUTH_USER']` and `$_SERVER['PHP_AUTH_PW']` to get username and password



start.html

```
<html>
<head><title>Start Page</title></head>
<body>
<center>
<p><a href="login.php">Login Page</a>
</center>
</body>
</html>
```




login.php

```
<?php include_once ("auth.php") ; ?>
<html>
<head><title>Login page</title></head>
<body>
<center>
You have successfully logged in and you
can view the<br/>
<a href="secretpage.php"><h1>secret page</h1></a>
</center>
</body>
</html>
```

- [view script authenticate/login.php](#)



secretpage.php

```
<?php include_once ("auth.php") ; ?>
<html>
<head><title>The secret page</title></head>
<body>
<center>
<p>Here is the secret information:</p>
<p>
Username is <?php echo $_SERVER['PHP_AUTH_USER'] ?>
<br/>
Password is <?php echo $_SERVER['PHP_AUTH_PW'] ?>
</p>
</center>
</body>
</html>
```

- [view script authenticate/secretpage.php](#)



auth.php (1)

```
<?php
    $authorized = false;
    $user = isset($_SERVER['PHP_AUTH_USER'])
        ? $_SERVER['PHP_AUTH_USER'] : '';
    $pass = $_SERVER['PHP_AUTH_PW'];
        ? $_SERVER['PHP_AUTH_PW'] : ''

    if ( isset($user) && isset($pass) &&
        $user === "Fred" && $pass === "Jones" )
    {
        $authorized = true;
    }
```



auth.php (2)

```
if (! $authorized)
{
    header('WWW-Authenticate: Basic realm=
        "Top Secret Files"');
    header('HTTP/1.0 401 Unauthorized');
?>

<html>
<head><title>Unauthorized</title></head>
<body><center>
<h1>You are not authorized to view the top
secret files</h1>
<p><a href="login.php"><h1>Login</h1></a></p>
</center></body></html>
<?php exit(); } ?>
```

- [view script authenticate/auth.php](#)

<http://localhost/php/authenticate/start.html>