

INTRODUCTION TO PHP SESSIONS AND COOKIES

What you Benefit ???

By the end of this session you will learn

- How to use Sessions and Cookies to maintain the state among multiple requests.

TASK OF THE DAY

LOGIN PAGE

Create a session when a user log in to his account. When user logout from his account the session should expire

Log into your account

Username

Password

Login

☐ Remember me

[New user](#)

INTRODUCTION TO PHP SESSIONS AND COOKIES

Introduction To PHP Sessions And Cookies

We had already tried passing data to a server .

But..how the server knows the user from which the requests are received...?



COOKIES

Cookies

- HTTP is a stateless protocol; this means that the web server does not know (or care) whether two requests comes from the same user or not; it just handles each request without regard to the context in which it happens.
- Cookies are used to maintain the state in between requests—even when they occur at large time intervals from each other.
- Cookies allow your applications to store a small amount of textual data (typically, 4-6kB) on a Web client browser.
- There are a number of possible uses for cookies, although their most common one is maintaining state of a user

Creating A Cookie

- `setcookie("userid", "100", time() + 86400);`

↓
Cookie variable name

↓
Expiration time.

↓
variable value

- This simply sets a cookie variable named "userid" with value "100" and this variable value will be available till next 86400 seconds from current time

Accessing a Cookie

- `echo $_COOKIE['userid'];` `// prints 100`
- **Cookie as array**
 - `setcookie("test_cookie[0]", "foo");`
 - `setcookie("test_cookie[1]", "bar");`
 - `setcookie("test_cookie[2]", "bar");`
- `var_dump($_COOKIE['test_cookie']);`

Destroying A Cookie

- There is no special methods to destroy a cookie, We achieve it by setting the cookie time into a past time so that it destroys it
 - Eg : `setcookie('userid',100,time()-100);`

cookies.html

```
1 <html>
2 <head> <title>Setting Cookie</title> </head>
3 <body>
4 <form method="post" action="file1.php">
5     Enter your first name:
6     <input type="text" name="user"> <br />
7     Enter your country: <br />
8     <input type="radio" name="country" value="India">India <br />
9     <input type="radio" name="country" value="Thailand">Thailand <br />
10    <input type="radio" name="country" value="Korea">South Korea <br />
11    <input type="radio" name="country" value="USA">USA <br />
12    <input type="radio" name="country" value="Netherlands">Netherlands <br />
13    <br /> <br />
14    <input type="submit" name="submit_button" value="Submit">
15 </form>
16 </body>
17 </html>
```

Form data is posted to file1.php.

Setting Cookie - Windows Internet Explorer



http://localhost/cookies.html



Favorites



Suggested Sites



FactChecking Day 2 of ...



Setting Cookie

Enter your first name:

Enter your country:

- ☐ India
- ☐ Thailand
- ☐ South Korea
- ☐ USA
- ☐ Netherlands

Submit

File1.php

```
1 <?php
2 if(isset($_GET['submit_button'])) {
3     $user = $_GET['user'];
4     $country = $_GET['country'];
5
6     if( $user && $country ) {
7         setcookie("firstname", $user, time()+3600);
8         setcookie("where", $country, time()+3600);
9         echo 'welcome '. $user . ' from '. $country;
10        echo '<br>please go to <a href="getcookies.php">getcookies.php</a>';
11
12    }
13 }
14 else
15 {
16     if(isset($_COOKIE['submit_button']))
17     {
18         $user = $_COOKIE['user'];
19         $country = $_COOKIE['country'];
20         echo "your name is $name this is done using cookies ;)";
21     }
22     else
23     {
24         echo 'please enter your name in the first <a href="cookies.html">page</a>';
25     }
26 }
27
28 ?>
```

Function **setcookie** takes the name of the cookie to be set as the first argument, followed by the value to be stored in the cookie. The optional third argument is the expiration time of the cookie.

Hyperlink to **getCookies.php**.

output

welcome ahmed from USA

please go to [page 2](#) to test the cookies

getcookie.php

```
1 <html>
2 <head> <title>Get Cookie Data</title> </head>
3 <body>
4 <?php
5     $userName = $_COOKIE['firstname'];
6     $countryName = $_COOKIE['where'];
7     echo 'Hello ' . $userName . ' from ' . $countryName . ' this is another page and the
8     cookie is still alive';
9 ?>
10 </form>
11 </body>
12 </html>
```

output

Hello ahmed from USA this is another page and the cookie is still alive

SESSIONS

WHY PHP – Sessions ?

Whenever you want to create a [website](#) that allows you to store and display information about a user, determine which user groups a person belongs to, utilize permissions on your [website](#) or you just want to do something cool on your site, [PHP's Sessions](#) are vital to [each](#) of these features.

Cookies are about 30% unreliable right now and it's getting worse every day. More and more web browsers are starting to come with security and privacy settings and people browsing the net these days are starting to frown upon Cookies because they store information on their local computer that they do not want stored there.

PHP has a great set of functions that can achieve the same results of Cookies and more without storing information on the user's computer. PHP Sessions store the information on the web server in a location that you chose in special files. These files are connected to the user's web browser via the server and a special ID called a "Session ID". This is nearly 99% flawless in operation and it is virtually invisible to the user.

Creating and accessing session

- Once session is started you can create and access session variables like any other arrays in PHP

– `$_SESSION['userid'] = 100;`



Session variable name



variable value

– `echo $_SESSION['userid'];`

//prints 100

PHP - Sessions

- Sessions store their identifier in a cookie in the client's browser
- Every page that uses session data must be preceded by the `session_start()` function
- Session variables are then set and retrieved by accessing the global `$_SESSION[]`
- Save it as `session.php`

```
1  <?php
2      session_start();
3      if (!$_SESSION["count"])
4          $_SESSION["count"] = 0;
5      if ($_GET["count"] == "yes")
6          $_SESSION["count"] = $_SESSION["count"] + 1;
7      echo "<h1>".$_SESSION["count"]."</h1>";
8  ?>
9  <a href="session.php?count=yes">Click here to count</a>
```

Comparison

Cookies

Cookies are stored in the user's browser

A cookie can keep information in the user's browser until deleted by user or set as per the timer. It will not be destroyed even if you close the browser.

Cookies can only store string

We can save cookie for future reference

Session

Sessions are stored in server

A session is available as long as the browser is opened. User cant disable the session. It will be destroyed if you close the browser

Can store not only strings but also objects

session cant be.