

## Managing Sessions

### 1) Using Session Variables to Save Values to the Current Session

- a) When session is turned on, a session will be automatically created by the PHP engine. To save any values to the session, use the pre-defined associative array `$_SESSION`. The following PHP script shows you how to save values to the session:

```
<?php
    session_start();
    echo "<html>";

    $_SESSION["myLogin"] = "myWebsite";
    echo "A value saved in the session named as myLogin.<br>";

    $_SESSION["myColor"] = "Blue";
    echo "A value saved in the session named as myColor.<br>";

    echo "Click <a href=next_page.php>Next Page</a>"
        ." to retrieve the values.<br>";
    echo "</html><br>";
?>
```

- b) Type in this script and save it to the Web server as **first\_page.php**. Run it with a browser.

### 2) To Retrieve Values from the Current Session

- a) The following PHP script shows you how to retrieve values from the session:

```
<?php
    session_start();
    echo "<html>";

    $myLogin = $_SESSION["myLogin"];
    echo "Value of myLogin has been retrieved: ".$myLogin."\n";

    $myColor = $_SESSION["myColor"];
    echo "Value of MyColor has been retrieved: ".$myColor."\n";

    echo "</html>\n";
?>
```

- b) Save this script as **next\_page.php**. Run the page and click the "Next Page" link.

### 3) To Retrieve the Session ID of the Current Session

The session ID is created by the PHP engine, there are two ways to retrieve it:

- Calling `session_id()` function. It will return the session ID value.
- Using built-in constant `SID`. It will contains a string of session ID name and value.

a) Open **next\_page.php** file and the code shown in red text as shown in the script below. The script shows you how to retrieve the session ID using the two methods:

```
<?php
    session_start();
    echo "<html>";

    $sid = session_id();
    echo "Session ID returned by session_id(): ".$sid."\n";
    $sid = SID;
    echo "Session ID returned by SID: ".$sid."\n";

    $myLogin = $_SESSION["myLogin"];
    print("Value of myLogin has been retrieved: ".$myLogin."\n");
    $myColor = $_SESSION["myColor"];
    echo "Value of myColor has been retrieved: ".$myColor."\n";

    echo "</html>\n";
?>
```

b) Save this script. Now run **first\_page.php** and click the "Next Page" link to get the output that looks like the following:

```
Session ID returned by session_id(): rfnql7ui6c7g6pjbt46n0vi97
Session ID returned by SID: PHPSESSID=rfnql7ui6c7g6pjbt46n0vi97
Value of myLogin has been retrieved: FYICenter
Value of mColor has been retrieved: Blue
```

Now you know that the session ID created by the PHP engine is 26 characters long with alphanumeric characters only.

### 4) To Find out How the Session IDs are Transferred to the Client Browser

Once a new session is created, its session ID must be transferred to the client browser and included in the next client request, so that the PHP engine can find the same session created by the same visitor. The PHP engine has two options to transfer the session ID to the client browser:

- As URL parameter (default option) - The Session ID will be embedded in all URLs in the HTML document delivered to the client browser. When the visitor clicks any of those URLs, the session ID will be returned back to the Web server as part of the requesting URL.
- As a cookie - The session ID will be delivered as a cookie to the client browser. When visitor requests any other pages on the Web server, the session ID will be returned back to the Web server also as a cookie.

a) Type in the following script:

```
<?php
    session_start();
    echo "<html>";

    $queryString = $_SERVER["QUERY_STRING"];
    echo "Query string of the incoming URL: ".$queryString."\n";

    echo "Cookies received:\n";
    foreach ($_COOKIE as $name => $value) {
        echo "    $name = $value\n";
    }

    $myLogin = $_SESSION["myLogin"];
    echo "Value of myLogin has been retrieved: ".$myLogin."\n";
    $myColor = $_SESSION["myColor"];
    echo "Value of myColor has been retrieved: ".$myColor."\n";

    echo "</html>\n";
?>
```

b) Save this script to the Web server as **next\_page.php**. Now visit **first\_page.php** and click the "Next Page" link, you will get something like this:

```
Query string of the incoming URL: PHPSESSID=mem1483hk4dvm1n2ii8k9hvjj1
Cookies received:
Value of myLogin has been retrieved: myWebsite
Value of myColor has been retrieved: Blue
```

Base on the output, the PHP engine is using URL parameters to transfer session IDs, if you can see the session ID parameter in the query string of the incoming URL, or it is transferred as a cookie if there is any cookies related to session ID.

Another way to confirm that your PHP engine is using URL parameters to transfer session IDs is to look at the address field of your browser, it will show something like:

```
http://localhost/next_page.php?PHPSESSID=mem1483hk4dvm1n2ii8k9hvjj1
```

In a special file named like: \temp\sess\_r66hq1bcg8o79e5i5gd52p26g3, if you open this file, you will see:

```
myLogin|s:9:"myWebsite";myColor|s:4:"Blue";
```

Now you know that session values are stored on the Web server as text files, and values are formatted with value names and lengths.

## 5) To Close a Session Properly

Let's say you site requires users to login. When a logged in user clicks the logout button, you need to close the session associated with this user properly in 3 steps:

1. Remove all session values with `$_SESSION = array()`.

2. Remove the session ID cookie with the *setcookie()* function.
3. Destroy the session object with the *session\_destroy()* function.

Below is a good sample script:

```
<?php
    session_start();

    $_SESSION = array();

    if (isset($_COOKIE[session_name()])) {
        setcookie(session_name(), '', time()-42000, '/');
    }

    session_destroy();

    echo "<html>";
    echo "Thank you for visiting FYICenter.com.<br>";
    echo "  <a href=login.php>Login Again.</a><br>";
    echo "</html><br>";
?>
```

## Managing Cookies

### 1) To Set and Retrieve Cookies

a) Type in the following script in a text editor:

```
<?php
    setcookie("myLoginName", "MyWebsite");
    setcookie("myPreferredColor", "Blue");
    echo "<br>";
    echo "2 cookies were delivered.<br>";

    if (isset($_COOKIE["myLoginName"])) {
        $loginName = $_COOKIE["myLoginName"];
        echo "Received a cookie named as LoginName: ".$loginName."<br>";
    } else {
        echo "Did not received any cookie named as LoginName.<br>";
    }

    $count = count($_COOKIE);
    echo "$count cookies received.<br>";
    foreach ($_COOKIE as $name => $value) {
        echo "  $name = $value<br>";
    }
    echo "<br>";
?>
```

b) Save the file as **setting\_receiving\_cookies.php**. Run the file on the browser, you will get:

```
2 cookies were delivered.
Did not received any cookie named as myLoginName.
0 cookies received.
```

"0 cookies received" is because there was no previous visit from this browser. But if you click the refresh button of your browser, you will get:

```
2 cookies were delivered.  
Received a cookie named as myLoginName: myWebsite  
2 cookies received.  
myLoginName = myWebsite  
myPreferredColor = Blue
```

## 2) To Set a Persistent Cookie

A persistent cookie is a cookie which is stored in a cookie file permanently on the browser's computer. By default, cookies are created as temporary cookies which stored only in the browser's memory. When the browser is closed, temporary cookies will be erased. You should decide when to use temporary cookies and when to use persistent cookies based on their differences:

- Temporary cookies can not be used for tracking long-term information.
- Persistent cookies can be used for tracking long-term information.
- Temporary cookies are safer because no programs other than the browser can access them.
- Persistent cookies are less secure because users can open cookie files see the cookie values.

a) Type in the following script and save it as **setting\_persistent\_cookies.php**.

```
<?php  
setcookie("myLoginName","myWebsite");  
setcookie("myPreferredColor","Blue");  
setcookie("CouponNumber","07470433",time()+60*60*24*7);  
setcookie("CouponValue","100.00",time()+60*60*24*7);  
  
print("\n");  
print("2 temporary cookies were delivered.\n");  
print("2 consistent cookies were delivered.\n");  
  
if (isset($_COOKIE["myLoginName"])) {  
    $loginName = $_COOKIE["myLoginName"];  
    print("Received a cookie named as myLoginName: ".$loginName."\n");  
} else {  
    print("Did not received any cookie named as myLoginName.\n");  
}  
  
$count = count($_COOKIE);  
print("$count cookies received.\n");  
foreach ($_COOKIE as $name => $value) {  
    print "  $name = $value\n";  
}  
print("\n");  
?>
```

b) Open the browser and run the file. You will see:

```
2 temporary cookies were delivered.
2 consistent cookies were delivered.
Did not received any cookie named as myLoginName.
0 cookies received.
```

Click the refresh button, you will see:

```
2 temporary cookies were delivered.
2 consistent cookies were delivered.
Received a cookie named as myLoginName: myWebsite
4 cookies received.
  myLoginName = myWebsite
  myPreferredColor = Blue
  CouponNumber = 07470433
  CouponValue = 100.00
```

Close your browser and open it again to the same page. You will see:

```
2 temporary cookies were delivered.
2 consistent cookies were delivered.
Did not received any cookie named as myLoginName.
2 cookies received.
  CouponNumber = 07470433
  CouponValue = 100.00
```

This proves that "CouponNumber" and CouponValue" persisted outside the browser.

## 2) To Remove a Cookie

a) Type in the following script

```
<?php
    setcookie("CouponNumber","",time()-1);
    setcookie("CouponValue","",time()-1);
    print("<pre>\n");
    print("2 cookies were delivered with past times.\n");

    $count = count($_COOKIE);
    print("$count cookies received.\n");
    foreach ($_COOKIE as $name => $value) {
        print "  $name = $value\n";
    }
    print("</pre>\n");
?>
```

b) Save the file as **removing\_cookies.php**. Run it on the browser, you will see:

```
2 cookies were delivered with past times.
2 cookies received.
CouponNumber = 07470433
CouponValue = 100.00
```

Click the refresh button, you will see:

```
2 cookies were delivered with past times.
```

```
0 cookies received.
```

As you can see, both cookies are removed.

### 3) To View the Content of a Cookie File

Go to \Documents and Settings\%user%\Cookies directory and open the cookie file, %user%localhost.txt. You will see:

```
CouponNumber
07470433
localhost/
1024
3084847744
29787636
2404950512
29786228
*
CouponValue
100.00
localhost/
1024
3084847744
29787636
2405150512
29786228
*
```