

COOKIES , SESSION

COOKIES

- Cookies are text files stored on the client computer and they are kept of use tracking purpose. PHP transparently supports HTTP cookies.
- There are three steps involved in identifying returning users :
 - ❖ Server script sends a set of cookies to the browser. For example name, age, or identification number etc.
 - ❖ Browser stores this information on local machine for future use.
 - ❖ When next time browser sends any request to web server then it sends those cookies information to the server and server uses that information to identify the user.

The Anatomy of a Cookie

- Cookies are usually set in an HTTP header (although JavaScript can also set a cookie directly on a browser).
- A PHP script that sets a cookie might send headers that look something like this:

```
HTTP/1.1 200 OK
Date: Fri, 04 Feb 2000 21:03:38 GMT
Server: Apache/1.3.9 (UNIX) PHP/4.0b3
Set-Cookie: name=xyz; expires=Friday, 04-Feb-07 22:03:38 GMT;
           path=/; domain=tutorialspoint.com
Connection: close
Content-Type: text/html
```

- **Cookie header** contains a name value pair, a GMT date, a path and a domain
- **The name and value** will be URL encoded.
- **The expires field** is an instruction to the browser to "forget" the cookie after the given time and date.

The Anatomy of a Cookie

- If the browser is configured to store cookies, it will then keep this information until the expiry date.
- If the user points the browser at any page that matches the path and domain of the cookie, it will resend the cookie to the server.
- The browser's headers might look something like this:

```
GET / HTTP/1.0
Connection: Keep-Alive
User-Agent: Mozilla/4.6 (X11; I; Linux 2.2.6-15apmac ppc)
Host: zink.demon.co.uk:1126
Accept: image/gif, */*
Accept-Encoding: gzip
Accept-Language: en
Accept-Charset: iso-8859-1,*,utf-8
Cookie: name=xyz
```

CONTOH COOKIES pada FIREFOX (1)

- Cookie adalah serangkaian teks yang disimpan pada komputer Anda oleh situs web yang Anda kunjungi.
- Pada umumnya cookie menyimpan pengaturan atau preferensi Anda untuk suatu situs web tertentu, misalnya bahasa yang dipilih, atau lokasi (negara) Anda.
- Ketika Anda kembali ke situs web tersebut, Firefox akan mengirimkan cookie yang bersesuaian kepada situs web yang bersangkutan.
- Dengan cara ini, situs dapat menampilkan informasi yang sesuai dengan pengaturan atau preferensi Anda.

CONTOH COOKIES pada FIREFOX (2)

- Cookie dapat menyimpan berbagai jenis informasi, termasuk di antaranya informasi pribadi seperti nama, alamat rumah, alamat email, atau nomor telepon Anda.
- Informasi ini hanya akan disimpan jika Anda pernah memberikan informasi ini kepada situs tersebut.
- Situs web tidak dapat mengakses informasi yang tidak pernah Anda berikan kepada situs web tersebut, dan situs web juga tidak dapat mengakses berkas lainnya pada komputer Anda.

Setting Cookies with PHP

- PHP provided **setcookie()** function to set a cookie.
- This function requires up to six arguments and should be called before <html> tag.
- For each cookie this function has to be called separately.

```
setcookie(name, value, expire, path, domain, security);
```

The detail of all the arguments **setcookie**

- **Name** – This sets the name of the cookie and is stored in an environment variable called HTTP_COOKIE_VARS. This variable is used while accessing cookies.
- **Value** – This sets the value of the named variable and is the content that you actually want to store.
- **Expiry** – This specifies a future time in seconds since 00:00:00 GMT on 1st Jan 1970. After this time cookie will become inaccessible. If this parameter is not set then cookie will automatically expire when the Web Browser is closed.
- **Path** – This specifies the directories for which the cookie is valid. A single forward slash character permits the cookie to be valid for all directories.
- **Domain** – This can be used to specify the domain name in very large domains and must contain at least two periods to be valid. All cookies are only valid for the host and domain which created them.
- **Security** – This can be set to 1 to specify that the cookie should only be sent by secure transmission using HTTPS otherwise set to 0 which means cookie can be sent by regular HTTP.

Example

```
<?php
    setcookie("name", "John Watkin", time()+3600, "/", "", 0);
    setcookie("age", "36", time()+3600, "/", "", 0);
?>
<html>

    <head>
        <title>Setting Cookies with PHP</title>
    </head>

    <body>
        <?php echo "Set Cookies"?>
    </body>

</html>
```

Accessing Cookies with PHP

- PHP provides many ways to access cookies. Simplest way is to use either **\$_COOKIE** or **\$HTTP_COOKIE_VARS** variables.
- Following example will access all the cookies set in above example.

```
<html>

<head>
  <title>Accessing Cookies with PHP</title>
</head>

<body>

  <?php
    echo $_COOKIE["name"]. "<br />";

    /* is equivalent to */
    echo $HTTP_COOKIE_VARS["name"]. "<br />";

    echo $_COOKIE["age"] . "<br />";

    /* is equivalent to */
    echo $HTTP_COOKIE_VARS["age"] . "<br />";
  ?>

</body>
</html>
```

You can use **isset()** function to check if a cookie is set or not.

```
<html>

<head>
  <title>Accessing Cookies with PHP</title>
</head>

<body>

  <?php
    if( isset($_COOKIE["name"]))
      echo "Welcome " . $_COOKIE["name"] . "<br />";

    else
      echo "Sorry... Not recognized" . "<br />";
  ?>

</body>
</html>
```

Deleting Cookie with PHP

- To delete a cookie : setcookie() with the name argument only but this does not always work well, however, and should not be relied on.

```
<?php
    setcookie( "name", "", time()- 60, "/", "", 0);
    setcookie( "age", "", time()- 60, "/", "", 0);
?>
<html>

    <head>
        <title>Deleting Cookies with PHP</title>
    </head>

    <body>
        <?php echo "Deleted Cookies" ?>
    </body>

</html>
```

SESSION

SESSION

- A session is a way to store information (in variables) to be used across multiple pages.
- Unlike a cookie, the information is not stored on the users computer.

PHP Session

- When you work with an application, you open it, do some changes, and then you close it. This is much like a **Session**. The computer knows who you are. It knows when you start the application and when you end. But on the internet there is one problem: the web server does not know who you are or what you do, because the HTTP address doesn't maintain state.
- Session variables solve this problem by storing user information to be used across multiple pages (e.g. username, favorite color, etc.). By default, session variables last until the user closes the browser.
- Session variables hold information about one single user, and are available to all pages in one application.

Start a PHP Session

- A session is started with the `session_start()` function.
- The `session_start()` function must be the very first thing in your document. Before any HTML tags.
- Session variables are set with the PHP global variable: `$_SESSION`.

- Example:

```
<?php
// Start the session
session_start();
?>
<!DOCTYPE html>
<html>
<body>

<?php
// Set session variables
$_SESSION["favcolor"] = "green";
$_SESSION["favanimal"] = "cat";
echo "Session variables are set.";
?>

</body>
</html>
```

Get PHP Session Variable Values

- Notice that session variables are not passed individually to each new page, instead they are retrieved from the session we open at the beginning of each page (session_start()).

```
<?php
session_start();
?>
<!DOCTYPE html>
<html>
<body>

<?php
// Echo session variables that were set on previous page
echo "Favorite color is " . $_SESSION["favcolor"] . "<br>";
echo "Favorite animal is " . $_SESSION["favanimal"] . ".";
?>

</body>
</html>
```


Another way to show all the session variable values for a user session

- ```
<?php
session_start();
?>
<!DOCTYPE html>
<html>
<body>

<?php
print_r($_SESSION);
?>

</body>
</html>
```

# Modify a PHP Session Variable

- ```
<?php
session_start();
?>
<!DOCTYPE html>
<html>
<body>

<?php
// to change a session variable, just overwrite it
$_SESSION["favcolor"] = "yellow";
print_r($_SESSION);
?>

</body>
</html>
```

Destroy a PHP Session

- ```
<?php
session_start();
?>
<!DOCTYPE html>
<html>
<body>

<?php
// remove all session variables
session_unset();

// destroy the session
session_destroy();
?>

</body>
</html>
```

# Reference

- <https://www.w3schools.com/php>