

Cookies in PHP

Cookies are used to store the information of a web page in a remote browser, so that when the same user comes back to that page, that information can be retrieved from the browser itself.

They are typically used to keep track of information such as a username that the site can retrieve to personalize the page when the user visits the website next time.

Cookies are often used to perform following tasks:

- **Session management:** Cookies are widely used to manage user sessions. For example, when you use an online shopping cart, you keep adding items in the cart and finally when you checkout, all of those items are added to the list of items you have purchased. This can be achieved using cookies.
- **User identification:** Once a user visits a web page, using cookies, that user can be remembered. And later on, depending upon the search/visit pattern of the user, content which the user likely to be visited are served. A good example of this is 'Retargetting'. A concept used in online marketing, where depending upon the user's choice of content, advertisements of the relevant product, which the user may buy, are served.
- **Tracking / Analytics:** Cookies are used to track the user. Which, in turn, is used to analyze and serve various kind of data of great value, like location, technologies (e.g. browser, OS) form where the user visited, how long (s)he stayed on various pages etc.

How to create a cookie in PHP

PHP has a **setcookie()** function to create a cookie.

```
<?php
```

```
setcookie(cookie_name, cookie_value, [expiry_time], [cookie_path], [domain], [secure], [httponly]);
```

```
?>
```

Here

- Php“setcookie” is the PHP function used to create the cookie.
- “cookie_name” is the name of the cookie that the server will use when retrieving its value from the \$_COOKIE array variable. It’s mandatory.
- “cookie_value” is the value of the cookie and its mandatory
- “[expiry_time]” is optional; it can be used to set the expiry time for the cookie such as 1 hour. The time is set using the PHP time() functions plus or minus a number of seconds greater than 0 i.e. time() + 3600 for 1 hour.
- “[cookie_path]” is optional; it can be used to set the cookie path on the server. The forward slash “/” means that the cookie will be made available on the entire domain. Sub directories limit the cookie access to the subdomain.
- “[domain]” is optional, it can be used to define the cookie access hierarchy.
- “[secure]” is optional, the default is false. It is used to determine whether the cookie is sent via https if it is set to true or http if it is set to false.
- “[Httponly]” is optional. If it is set to true, then only client side scripting languages.

Creating Cookie:

```
<?php
```

```
    setcookie("user_name", "karishma", time()+ 60, '/'); // expires after 60 seconds
```

```
    echo 'the cookie has been set for 60 seconds';
```

```
?>
```

Accessing Cookies Values

The PHP `$_COOKIE` superglobal variable is used to retrieve a cookie value. It is typically an associative array that contains a list of all the cookies values sent by the browser in the current request, keyed by cookie name. The individual cookie value can be accessed using standard array notation,

```
<?php
```

```
// Accessing an individual cookie value
```

```
echo $_COOKIE["user_name"];
```

```
?>
```

You can use the PHP **isset()** function to check whether a cookie is set or not before accessing its value.

```
<?php

// Verifying whether a cookie is set or not

if(isset($_COOKIE["user_name"])){

    echo "Hi " . $_COOKIE["user_name"];

} else{

    echo "Welcome Guest!";

}

?>
```

Removing Cookies

You can delete a cookie by calling the same setcookie() function with the cookie name and any value (such as an empty string) however this time you need to set the expiration date in the past.

```
<?php

// Deleting a cookie

setcookie("user_name", "", time()-3600);

?>
```

Difference Between Session and Cookie in PHP

Cookie	Session
Cookies are client-side files that can store some data on browsers	Sessions are server-side files that can store some data on servers.
You don't need any function to start Cookie as It is stored in your local machine.	Before using <code>\$_SESSION</code> , you have to call <code>session_start()</code> ; This function will start the session and then, you can access <code>\$_SESSION</code> data on that page.
Most browser's maximum Cookie size is 4KB (4096 Bytes)	You can store as much data as you like within sessions. The only limits you can reach is the maximum memory a script can consume at one time, which by default is 128MB.