

Cookies and sessions are two related concepts in web development, but they serve different purposes.

**Cookies** are small pieces of data that are stored on the user's computer. They can be used to store a variety of information, such as user preferences, login status, or items in a shopping cart. Cookies are created by the web server and sent to the user's browser, which then stores them on the user's computer. When the user visits the same website again, the browser sends the cookies back to the server.

**Sessions** are a way of storing data on the server that is specific to a particular user. Sessions are started when a user visits a website and are terminated when the user closes their browser or logs out. Sessions are typically used to store information about the user's current activity on the website, such as the items in their shopping cart or the pages they have visited.

In PHP, cookies and sessions can be used together to implement a variety of features, such as user authentication, shopping carts, and personalized content.

## Cookies in PHP

To create a cookie in PHP, you can use the `setcookie()` function. This function takes three arguments: the name of the cookie, the value of the cookie, and the expiration time of the cookie. The expiration time is the date and time after which the cookie will be deleted. If you do not specify an expiration time, the cookie will be deleted when the user closes their browser.

To retrieve a cookie in PHP, you can use the `$_COOKIE` superglobal variable. This variable contains an array of all of the cookies that have been set for the current page request.

## Sessions in PHP

To start a session in PHP, you can use the `session_start()` function. This function must be called before any output is sent to the browser.

To store data in a session, you can use the `$_SESSION` superglobal variable. This variable contains an array of all of the data that is associated with the current session.

To destroy a session, you can use the `session_destroy()` function. This function will delete all of the data that is associated with the current session.

## Example

The following example shows how to use cookies and sessions in PHP to implement a

simple login system:

```
PHP
<?php

// Start the session
session_start();

// If the user is not logged in, redirect them to the login page
if (!isset($_SESSION['logged_in'])) {
    header('Location: login.php');
}

// Display the user's name
echo 'Welcome, ' . $_SESSION['username'] . '!';

?>
```

The login page would then contain a form where the user can enter their username and password. When the user submits the form, the following code would be used to log the user in and create a session:

```
PHP
<?php

// Start the session
session_start();

// Get the user's username and password from the form
$username = $_POST['username'];
$password = $_POST['password'];

// Validate the user's credentials
// ...

// If the user's credentials are valid, log them in and create a session
$_SESSION['logged_in'] = true;
$_SESSION['username'] = $username;

// Redirect the user to the main page
header('Location: index.php');
```

?>

The user would then be able to navigate throughout the website without having to log in again.

## **Conclusion**

Cookies and sessions are two important concepts in web development. Cookies can be used to store a variety of information on the user's computer, while sessions can be used to store data on the server that is specific to a particular user. PHP provides a variety of functions for working with cookies and sessions, making it easy to implement a variety of features on your website.

## **Sources**

1. <https://github.com/AnthonyBahl/WEB3400>