

# LiveCode Lessons (copy)

## Topics

**Installing LiveCode Server** 6

**Using LiveCode Server** 4

**Interacting with LiveCode Server** 6

[How do I pass information to LiveCode Server scripts?](#)

[How do I handle user input using LiveCode Server?](#)

[How Do I Use AJAX with LiveCode Server?](#)

[How to upload a file with LiveCode Server](#)

[How do I use Cookies on LiveCode Server?](#)  
(Current Article)

[Accessing Web Services using LiveCode](#)

## Last Updated

Apr 04, 2016

[Print Article](#)

## Other Resources

### Getting Started with LiveCode

[Get Up and Running with LiveCode](#)

[Getting Started with LiveCode Development](#)

### LiveCode Lessons

[How To - Step-By-Step Guides To Tasks In LiveCode](#)

[How To - LiveCode Server Tasks](#)

[How To - LiveCode Mobile Tasks](#)

[How To - LiveCode Sample Scripts](#)

[How to - LiveCode Marketplace Products](#)

[How to Purchase and License LiveCode](#)

### Tutorials

[Creating a Video Library Application](#)

### Data Grid

[LiveCode Data Grid](#)

[Data Grid Tips & Tricks](#)

[Converting the Stock Program](#)

### LiveCode Releases

[LiveCode 6.5](#)

[LiveCode 6.7](#)

[LiveCode 8](#)

## Comments

[Liquid error: internal](#) for this article

[LiveCode Lessons \(copy\)](#) / [LiveCode Lessons](#) / [How To - LiveCode Server Tasks](#) / [Interacting with LiveCode Server](#) / [How do I use Cookies on LiveCode Server?](#)

## How do I use Cookies on LiveCode Server?

The LiveCode Server product brings our english like language to the server environment. The server engine is a separate build of the LiveCode engine with specific syntax and functionality that makes it suitable for use in command-line contexts and, in particular, as a CGI processor.

This lesson describes how to process cookies in your LiveCode Server scripts. Code samples are provided

### Introduction

Cookies provide a convenient way to store session specific information client side.

The information stored in cookies varies and often covers user names, encrypted passwords or shopping trolley content.

Cookies are widely used, and their management is largely down to the configuration of a user's web browser and cookie clean-up policy. The user can enable and disable the support for cookies and remove stored cookies from their machine at any time.

### Specifying the Cookie Duration

As we discussed in the Introduction, the user can control if cookies are supported and when cookies are to be removed from their system. As a webmaster, this is not a means on which to rely when considering the time a cookie should be available.

Cookies can be configured to have a lifespan that is set when the cookie is created by the web page. The following examples demonstrate the creation of two cookies, called "yearcookie" and "sessioncookie":

```
<?lc

// Cookie lasts 1 year

put cookie "yearcookie" for "/" with "This cookie will stay on your system for a year" until (the seconds + 60 * 60 * 24 * 365)

// Cookie lasts until browser closes

// The default expiry time of a cookie is set to when the user's browser closes.
```

```
put cookie "sessioncookie" for "/" with "This cookie will be cleared when you close your browser closes"
```

```
?>
```

### Reading Cookies

Cookies can be read from the user's computer using the `$_COOKIE` keyword. The following code prints the contents of the cookies that were created in the previous examples to the screen:

```
<?lc

// Output cookies

put "PRINTING COOKIES:" & "<br /><br />"

put $_COOKIE into tCookiesArray

if tCookiesArray is an array then

printArray "$_COOKIE", tCookiesArray

else

put "no cookies found"

end if

put "<br /><br />" & "FINISHED"

?>
```

### Supporting Code

The code shown so far provides the basic mechanism to save and read cookie content. The array printing command that formats the output of the cookie content is provided here. This code is provided separately in this lesson, as it only serves to present the content in a readable form for this lesson. Normally you would write LiveCode commands to process the cookie content and extract the login information, shopping cart data or other content stored.

```
<?lc

// Commands to display an array in readable form

command printArray pArrayName, pArray

if pArray is an array then

put pArrayname & " = {" & "<br />"

dumpArray pArray, " ", " "

put "}"

end if

end printArray

command dumpArray pArray, plndent, plndentString

local tKeys

if pArray is an array then

put the keys of pArray into tKeys

sort lines of tKeys

repeat for each line tKey in tKeys

if pArray[tKey] is an array then

put plndent & tKey & " = {" & "<br />" & return

dumpArray pArray[tKey], (plndent & plndentString), plndentString

put plndent & "}" & "<br />" & return

else

put plndent & tKey & " = " & pArray[tKey] & "<br />" & return

end if

end repeat

end if

end dumpArray

?>
```

### Output - No Cookies

The first time you run the code listed in this lesson, you may get the following output:

PRINTING COOKIES:

no cookies found

FINISHED

This may be because the printing code is executed before you have saved the cookies or because of a latency that exists when writing a cookie to a file system.

If no cookie data is displayed, then refresh your web browser screen. You should then receive the output that is shown in the next step.

### Output - Cookies Found

If your code was successful, then you should see the following output in your web browser window:

PRINTING COOKIES:

```
$_COOKIE = {

sessioncookie = This cookie will be cleared when you close your browser closes

yearcookie = This cookie will stay on your system for a year

}
```

FINISHED

Both cookie examples are listed and the content of the two cookies is also displayed.

[Prev: How to upload a file with LiveCode Server](#) [Next: Accessing Web Services using LiveCode](#)

## 0 Comments

### Add your comment

Name\*


Email\*

Comment\*

☐ **Subscribe** E-Mail me when someone replies to this comment

Are you human?

☐ I'm not a robot

  
reCAPTCHA  
[Privacy](#) - [Terms](#)