



Office of Digital Humanities

[Back](#) [BYU LiveCode Lessons Gateway](#)

Web Services

Useful Terms

As we talk about using web services in LiveCode stacks here are a few definitions and LiveCode commands and functions that will be useful.

Definitions

parse

To analyze a string of text or data to in order to break it up into usable components. For example, suppose I have some html text:

```
<p>This is line one.</p>
<p>This is some <b>important</b> information.</p>
```

I could parse the text to get just the text without html tags, or to get just the words that are in bold face, or any number of other ways. I just need to figure out how to isolate the components I need.

client

the computer of the person visiting a web site

host

the computer that runs the web site

LiveCode language tokens

the URLEncode function - Returns a string that has been transformed so that it can be safely posted to an HTTP server. You may have noticed that often web browsers transform URLs that have non-alphanumeric characters in them. This is because many of these symbols have a specific meaning in the http protocol. By transforming them before sending them you ensure that they arrive intact at the server, which decodes them before using them.

```
URLEncode(argString)
```

Examples:

```
URLEncode("ABC123") -- returns "ABC123" (alphanumerics are untouched)
URLEncode("Test string $$") -- returns "Test+string+%24%24"
-- notice that spaces are turned to + and other symbols are turned
-- to a % followed by that symbol's standard character code.
```

replace command - replaces all occurrences of first string with second string in the container.

```
replace x with y in container
replace x with empty in container -- deletes all instances of x in container.
```

Example:

```

put "In my best interests." into tString
replace "in" with "out" in tString
--> tString is now "out my best outterests."

put "<p>A line of html text.</p>" tResult
replace "<p>" with empty in tResult
replace "</p>" with empty in tResult
--> tResult is now "A line of html text."

```

the itemDelimiter property - changes the character that determines what an item is from the default comma to any other character you choose.

```
set the itemDelimiter to ":"
```

Examples:

```

put "<p>A line of html text.</p>" tResult
set the itemDelimiter to ">"
put item 2 of tResult into tResult
--> tResult is now "A line of html text.</p>"
set the itemDelimiter to "<"
put item 1 of tResult into tResult
--> tResult is now "A line of html text."

```

You can see that the outcome here is the same as with the replace example. Often using items and the itemDelimiter can be a better solution with complex strings.

the menuHistory property - Use the menuHistory property to change the selected item in a menu, or to find out which menu item is currently selected. Gives you the line number of the currently selected menu item.

```
put the menuHistory of [menu button] into container
```

Examples:

```

put the menuHistory of btn "languageChoice" into tLangNum
set the menuHistory of btn "languageChoice" to 3

```

the merge function - Lets you build complex text strings using variables and other expressions.

```
put merge(string containing [[expression to evaluate]])
```

Examples:

```

merge("1+1 equals [[1+1]]") -- returns "1+1 equals 2"
merge("The current folder is [[the defaultFolder]]")

```

For example, if I needed to build an HTML tag with embedded quotes I could concatenate strings like this:

```
put "<a href=" & quote & tURL & quote & ">" into tTag
```

Instead I could use the merge function, in a format that some find easier to read:

```
put merge("<a href= [[quote]] [[tURL]] [[quote]] >") into tTag
```

This turns out to be really useful in constructing strings of arguments to send to web services, as you frequently need to do when querying web service urls. For example, in building a URL for a GET request I could do the following:

```
put merge("http://my.server.com?term=[[URLencode(fld "term")]]") into tURL
-- tURL value is: http://my.server.com?term=text+from+field
```

Close window

[Back](#) [BYU LiveCode Lessons Gateway](#)

Maintained by Devin Asay.

Copyright © 2005 Brigham Young University.

This page last updated on March 06, 2020 17:02:37.