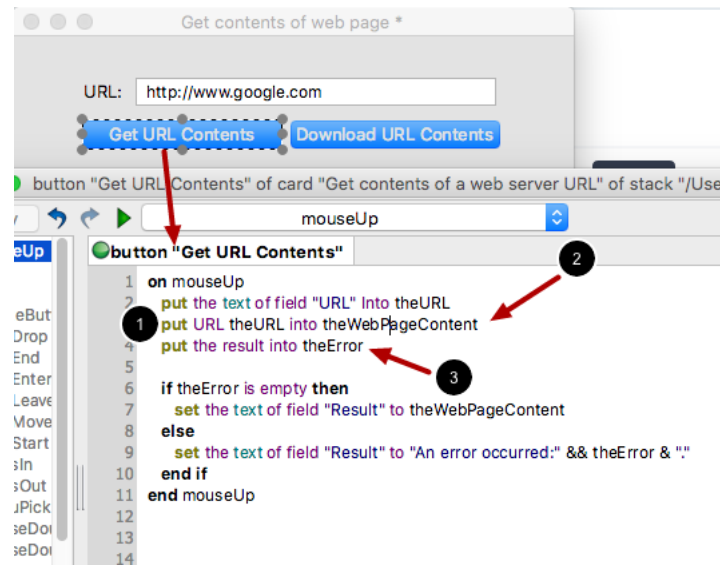


Using put



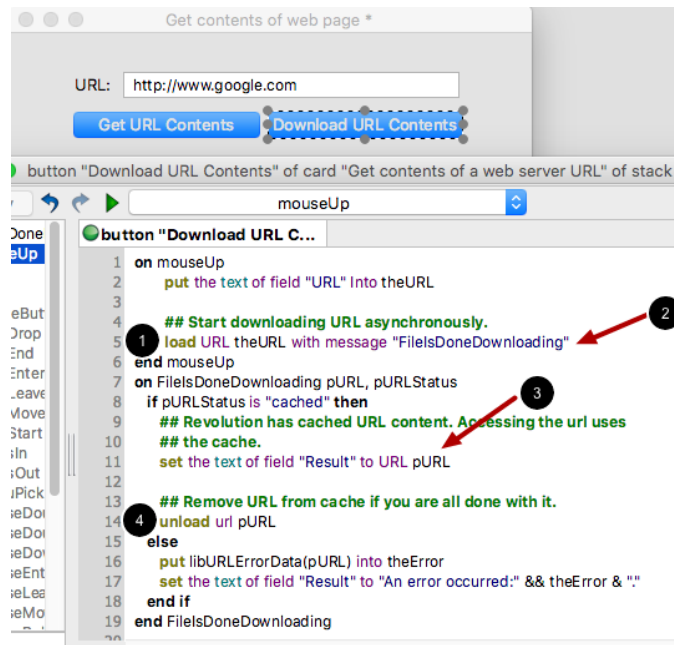
First we will look at how to download a URL using the **put** command. In particular we will use the **put URL** form (1). *put URL* is an easy way to synchronously download a URL into LiveCode. You just provide a URL and LiveCode will download the URL contents and store them. In this case the contents are being stored in the variable **theWebPageContent** (2).

The fact that *put URL* is synchronous means that your LiveCode code will not continue executing until the URL has finished downloading. For URLs that take a long time to download your UI will become unresponsive until the download process is complete. Because of this we can check for any errors that may have occurred while downloading the URL right after the *put URL* line (3).

```
on mouseUp
    put the text of field "URL" into theURL
    put URL theURL into theWebPageContent
    put the result into theError

    if theError is empty then
        set the text of field "Result" to theWebPageContent
    else
        set the text of field "Result" to "An error occurred:" && theError & "."
    end if
end mouseUp
```

Using load



Now we will look at how to download a URL using the **load** command (1). The *load* command requires a little more work than using **put URL** but has the added benefit of being an asynchronous action.

The fact that *load* is asynchronous means that your LiveCode will continue executing while the URL is being downloaded. For long download operations your application will remain responsive. When calling an asynchronous command we cannot get the results of the download operation right after making the call. Instead we make use of a feature in LiveCode called **callbacks**. A callback is a message that will be sent when the *load* operation has finished. In this example we tell LiveCode that the message **FilesDoneDownloading** should be sent when the url finishes downloading (2).

When the *FilesDoneDownloading* message is sent by the engine we can get the contents of the URL. LiveCode actually caches the contents of the URL on the computer so that the next time you access the URL the cached data is returned (3). When we no longer need the URL contents any longer we clear the cache by calling **unload url** (4).

```
on mouseUp
    put the text of field "URL" into theURL

    ## Start downloading URL asynchronously.
    load URL theURL with message "FileIsDoneDownloading"
end mouseUp
```

```
on FileIsDoneDownloading pURL, pURLStatus
if pURLStatus is "cached" then
    ## LiveCode has cached URL content. Accessing the url uses
    ## the cache.
    set the text of field "Result" to URL pURL

    ## Remove URL from cache if you are all done with it.
    unload url pURL
else
    put libURLErrorData(pURL) into theError
    set the text of field "Result" to "An error occurred:" && theError & "."
end if
end FileIsDoneDownloading
```

2 Comments

student_developer Wednesday Mar 19 2014 at 10:53 AM

Hi, I want to use a functionality that loads a google map on the screen. Can you please help me

Thanks

Elanor Buchanan Thursday Mar 27 2014 at 09:12 AM

Hi

Have a look at this lesson

<http://lessons.runrev.com/s/lessons/m/4069/l/32356-how-do-i-access-maps-on-ios> (<http://lessons.runrev.com/s/lessons/m/4069/l/32356-how-do-i-access-maps-on-ios>)

This will work on mobile or desktop, on desktop google maps will be opened in your default browser.

I hope that helps.

Elanor