How to implement script variables:

Concepts used in this program:

- Algorithms
- Lists
- Procedures with inputs
- Simple looping (repeat, forever)
- String operators
- Variables - local/global scope
- Variables – numbers
- Variables – strings

The Problem:

Scratch (as of Version 3) allows you to have private variables for a sprite, but not for a custom block or a script. In this article, we will see how to overcome this shortcoming.

The Proposed Solution:

Well, I thought back to my college course in Operating Systems in which I had learned that local variables are memory locations on a stack. So, I said to myself: Why not implement a stack in Scratch? And that's exactly what I did to solve the above problem.

I used the List mechanism to implement my stack. Here is how it works.

I created a list variable called "stack". The top locations in "stack" are used by the custom block as its local variables. The custom block adds these new locations at the beginning and deletes them before returning. (That is similar to how a stack works.) See the algorithm for the custom block below:

```
Add a new item at the top of stack
<Do whatever>
Delete the top item of stack
```

Version 1:

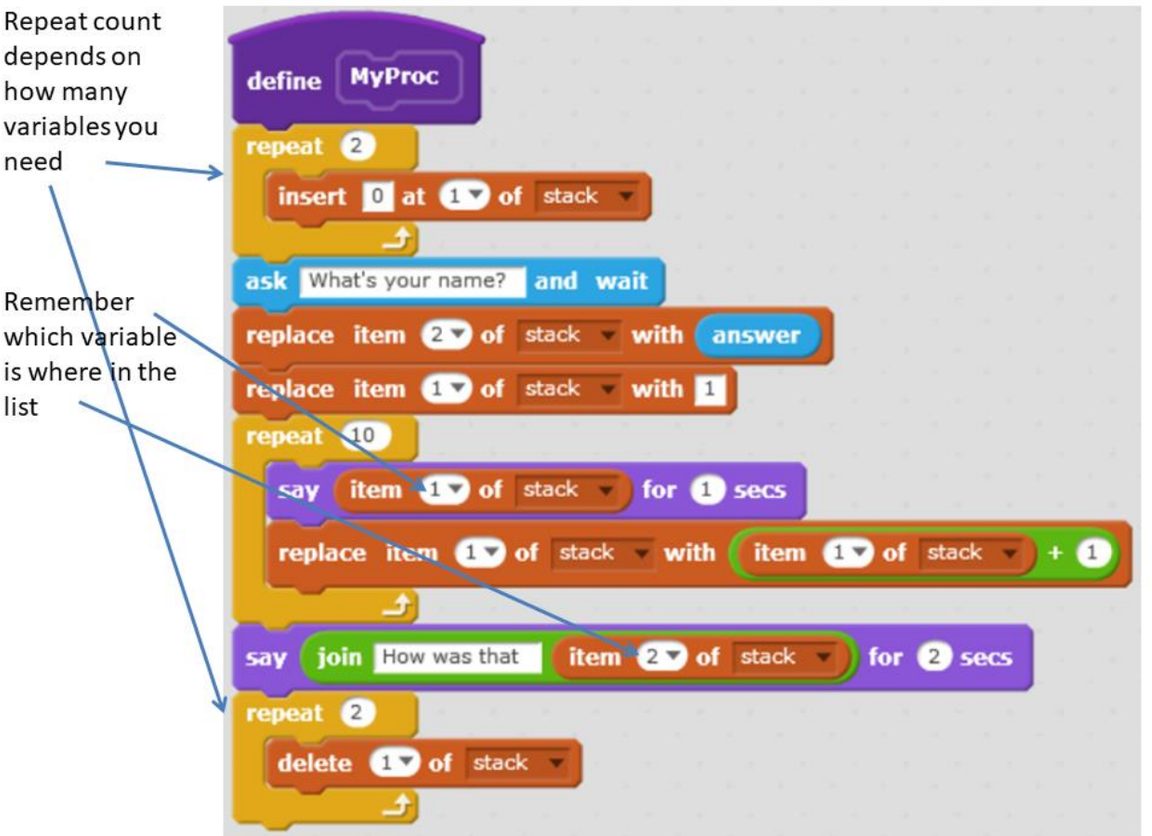
Let's take a trivial example to understand how this might work:

Let's say I have this new custom block:

```
MyProc1:
Local variable myname
Local variable counter
Ask user his/her name and save it in "myname"
```

```
Count 1 to 10 using variable counter
```

Here is how this can be implemented using our "stack" list.



The first repeat loop allocates 2 local variables by adding items to "stack". The last repeat loop de-allocates these local variables.

Within the program you must be consistent about which item in "stack" is to be used as which local variable. For instance, in our script above, item 1 is used as "counter" and item 2 is used "myname".

Multiple functions and nested calls:

What if we need multiple custom blocks (or scripts) each of which needs local variables? Well, the above scheme works perfectly in that scenario too. See the algorithm below for another custom block called "MyProc2".

```
MyProc2:
Local integer variables one, two, three, result.
Ask user for 3 numbers and save them in these variables.
Add the 3 numbers and announce "result".
```

Check out my program at the link below.

Solution: scriptvariables-1.sb2
https://scratch.mit.edu/projects/319734067/

Version 2:

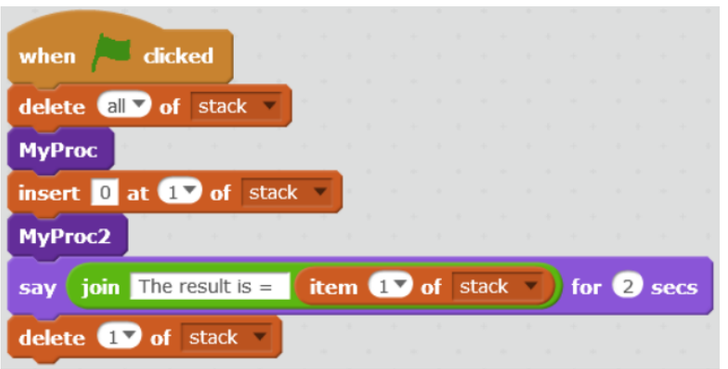
While we are at it, we can address another shortcoming in Scratch – which is that Scratch custom blocks don't allow a "return value". We can use the same stack to address this lacuna.

Let's modify MyProc2 above as follows:

```
MyProc2:
Local integer variables one, two, three.
Ask user for 3 numbers and save them in these variables.
Add the 3 numbers and return the result to the caller.
```

Design:

The trick for this is for the caller to allocate a slot on the stack for the return value, which the MyProc2 will use. See the calling script below.



Notice the "insert" command to allocate room for the return value. After calling MyProc2 the return value would be in the top item, which must be deleted after it has been used.

The script for MyProc2 would save the sum of the 3 numbers at item #4 of stack since the top 3 are its own local variables. Check out the solution below to see the entire script.

Solution: scriptvariables-2.sb2
https://scratch.mit.edu/projects/319734307/

Version 3:

Although this is a nifty scheme to have script-local variables and return values, the implementation looks kind of bulky. When you read the scripts of the above program, they contain really long statements, such as:

Assuming this relates to a local variable I, the above statement is equivalent to:
I = I + 1

We can make our stack implementation appear less bulky by implementing custom blocks for these statements. For instance, the above block can be made as compact as:

by using the following custom block:

We will define similar blocks to:

- Create local variables on the stack
- Delete local variables
- Set a local variable with a value

Solution: scriptvariables-final.sb2
https://scratch.mit.edu/projects/320175133/

Author: Abhay B. Joshi (abjoshi@yahoo.com)
Last updated: 8 July 2019