	Document Title	Application Guide	
	Application Name	Macro Maker	
	Document Version	2.0	
	Written By	Vlad Feldfix	**Page 1 of 6**

1. Overview

1.1. This application will automatically complete a set of instructions written in a script file.

1.2. Once the RUN button is pressed, don't touch the computer until the program is finished unless the script file specifies otherwise.

1.3. When the algorithm is done press ENTER.

2. How to use

2.1. Press 1 in the main menu to RUN the script.

2.2. Press 2 to edit the list.

2.3. Press 3 to edit the operation script.

3. Settings

3.1. Script Location > script.txt

4. Related files

4.1. script.txt

4.1.1. `MOUSE_POS_SET(X, Y)` - Sets the new position of the cursor.

4.1.1.1. **X** new x cursor position.

4.1.1.2. **Y** new y cursor position.

4.1.2. `MOUSE_CLICK(BUTTON)` - Click the mouse button.

4.1.2.1. **BUTTON** possible options for this argument are: left, right, and middle.

```
MOUSE_POS_SET(158,125)
MOUSE_CLICK(left)
```

Code 1 – example of function MOUSE_CLICK()

4.1.3.   MOUSE_DOWN(BUTTON) - The difference between MOUSE_CLICK and MOUSE_DOWN is that MOUSE_DOWN clicks without releasing, whilst MOUSE_CLICK clicks and releases immediately. This command together with MOUSE_UP can be used to drag and drop.

4.1.3.1.   **BUTTON** – See argument BUTTON for the MOUSE_CLICK command.

4.1.4.   MOUSE_UP(BUTTON) - This command releases the mouse button. See MOUSE_DOWN to learn more.

4.1.4.1.   **BUTTON** See argument BUTTON for the MOUSE_CLICK command.

4.1.5.   KEYBOARD_CLICK(BUTTON) - This command clicks a given keyboard button.

4.1.5.1.   **BUTTON** This argument can take many values such as alt, shift, enter, home, etc, up, down, etc. For more information see Python library: keyboard 0.13.5.

4.1.6.   KEYBOARD_PRESS(BUTTON) - Similarly to the mouse functions, this command presses (without releasing) any key on the keyboard. This can be used together with the KEYBOARD_RELEASE command to click two keys at the same time. See example below

4.1.6.1.   **BUTTON** see KEYBOARD_CLICK command.

```
KEYBOARD_PRESS(ctrl)
KEYBOARD_PRESS(s)
KEYBOARD_RELEASE(s)
KEYBOARD_RELEASE(ctrl)
```

Code 2 – example of function KEYBOARD_PRESS()

4.1.7.   KEYBOARD_RELEASE(BUTTON) - This command releases a pressed keyboard button. See more information in the KEYBOARD_PRESS command.

4.1.7.1.   **BUTTON** see KEYBOARD_CLICK command.

4.1.8.   SLEEP(SECONDS) - Pauses for a given number of seconds. This can be useful if you need to let something load in the middle of the automation, for example, a web page.

4.1.8.1.   **SECONDS** the number of seconds to wait.

4.1.9.   SET_ALARM() - Allows the user to set their own alarm for when they want to automation to start.

4.1.9.1.   This command allows the user to set an alarm when they need the automation to start (or continue) working.

4.1.9.2.   The alarm command will ask the user to provide a date and time for when the alarm should go off.

4.1.9.3.   Make sure that the application is open until the time the alarm is set.

4.1.10.   INPUT(PROMPT) - Asks the user to input data. It can also be useful to display a message/warning.

4.1.10.1.   **PROMPT** This argument is the message that is displayed for the user on the console screen of the application.

```
INPUT(Hello user, make everything ready and click ENTER to continue...)
INPUT(Insert your birth date as YYYY-MM-DD)
```

Code 3 – example of function INPUT()

4.1.11.   LOOP(CYCLES) - You can run automation in a loop X times.

4.1.11.1.   **CYCLES** is the number of times the loop will run until the command END_LOOP. The CYCLES argument can take 3 possible values:

4.1.11.1.1.   A number 1, 2, 3, 4, …

4.1.11.1.2.   LIST If you set argument CYCLES to LIST, the loop will go for every item in the list file list.csv

4.1.11.1.3.   INPUTDATA if you set argument CYCLES to INPUTDATA, the user will choose how many times the loop will cycle.

4.1.12.   END_LOOP() - Every loop must be completed by the END_LOOP command.

```
LOOP(15)
MOUSE_CLICK(left)
END_LOOP()
INPUT(Set the number of times to loop)
LOOP(INPUTDATA)
MOUSE_CLICK(right)
END_LOOP()
LOOP(LIST)
MOUSE_CLICK(middle)
END_LOOP()
```

Code 4 – example of function LOOP(). In this simple example, there are 3 loops. The first one will make the mouse click left 15 times. The second loop will make the mouse click right as many times as the user inputs. the third loop will make the mouse click the middle for as many lines as there are in the list.

4.1.13. KEYBOARD_WRITE(TEXT) - This command will input a pre-written text. This text can be given in the code or taken from the list.

    4.1.13.1. **TEXT** This argument is the text the automation will write. It can be a per-written text or a list value. To make it a list value set this argument to LSTVAL#X where X is the column of the list you want to take information from. Another possible value can be INPUTDATA where the keyboard will write the user input taken from the INPUT command (see INPUT command for more info)

```
list.csv content:
row11,row12,row13
row21,row22,row23
row31,row23,row33
```

```
script.csv:
INPUT(Now open notepad and press ENTER to continue)
SLEEP(5)
LOOP(4)
KEYBOARD_WRITE(Hello World)
KEYBOARD_CLICK(enter)
END_LOOP()
LOOP(LIST)
KEYBOARD_WRITE(From List:)
KEYBOARD_CLICK(enter)
KEYBOARD_WRITE(LSTVAL#0)
KEYBOARD_CLICK(enter)
KEYBOARD_WRITE(LSTVAL#1)
KEYBOARD_CLICK(enter)
KEYBOARD_WRITE(LSTVAL#2)
KEYBOARD_CLICK(enter)
END_LOOP()
```

```
Output:
Hello World
Hello World
Hello World
Hello World
From List:
row11
row12
row13
From List:
row21
row22
```

```
row23
From List:
row31
row23
row33
From List:
row31
row23
row33
```

Code 5 – example of function KEYBOARD_WRITE(TEXT). Notice! When using the KEYBOARD_WRITE() function, make sure that there is a place to receive the text. For example notepad or input box in a web page.

    4.2.   list.csv

        4.2.1.   You can insert any number of rows and columns you need.

5.   Output

    5.1.   -N/A-

6.   Document version history

    6.1.   `v1.0` Created on 2024-01-08

    6.2.   `v1.1` Created on 2024-10-28

        6.2.1.   New document format

    6.3.   `v2.0` Created on 2025-01-12

        6.3.1.   New document format