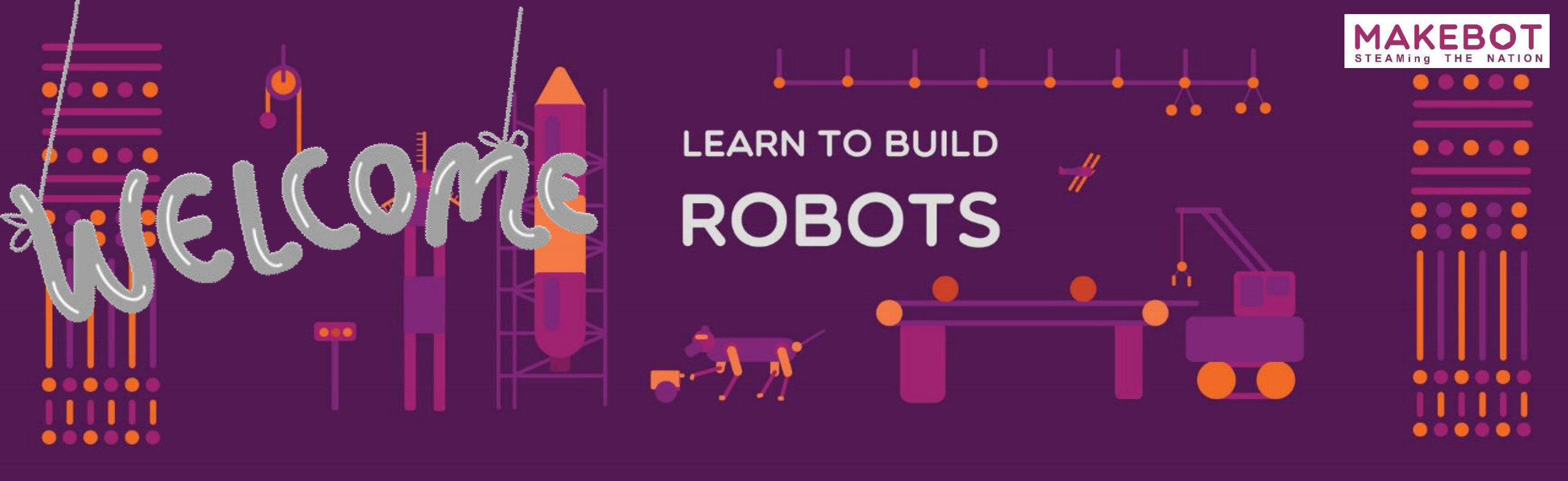




My first Software

For today's session:

1. mBlock software

2. My first software building



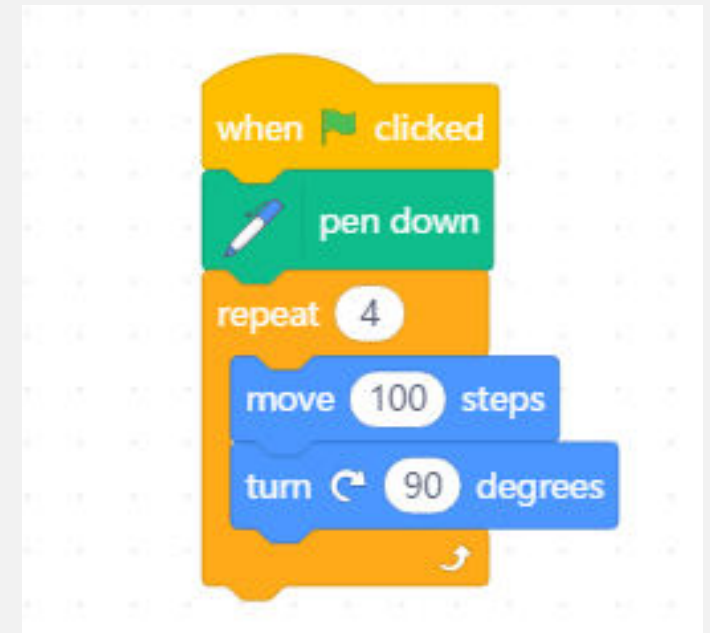
**Let's begin with
mBlock Programming**

mBlock- Scratch based software

- Based on MIT's scratch for STEAM education
- Drag and Drop type of coding
- makes programming as easy as building blocks
- Different Blocks with different functions
- You can create stories, games, animations, shapes, control your robot ,etc

- Download link:

<https://mblock.makeblock.com/en-us/download/>



My first software

Calculator

The image shows the mBlock software interface, which is used for creating and programming visual basic blocks (VBs) for micro:bit. The main workspace displays a calculator application with the following elements:

- Top Bar:** Contains a trash icon, a close icon, and a 'display' button.
- Main Stage:** Displays a grid of colorful 3D numbers (1-9 and 0) and mathematical operators (+, -, x, ÷, x², √, =).
- Left Sidebar:** Shows a list of sprites, including 'Lovely numbers2', which is currently selected. Below the list are buttons for 'Costumes' and 'Sounds'.
- Right Panel:** Contains tabs for 'Tutorials', 'Feedback', and 'Python', along with a search bar and a 'Blocks' tab.
- Bottom Taskbar:** Shows the Windows taskbar with various application icons and a search bar.

What did we see?

Observations:

- It's a calculator
- Using this calculator we can perform many operations like addition, subtraction, multiplication, division, square and square root.
- We can perform operation on any digit number.

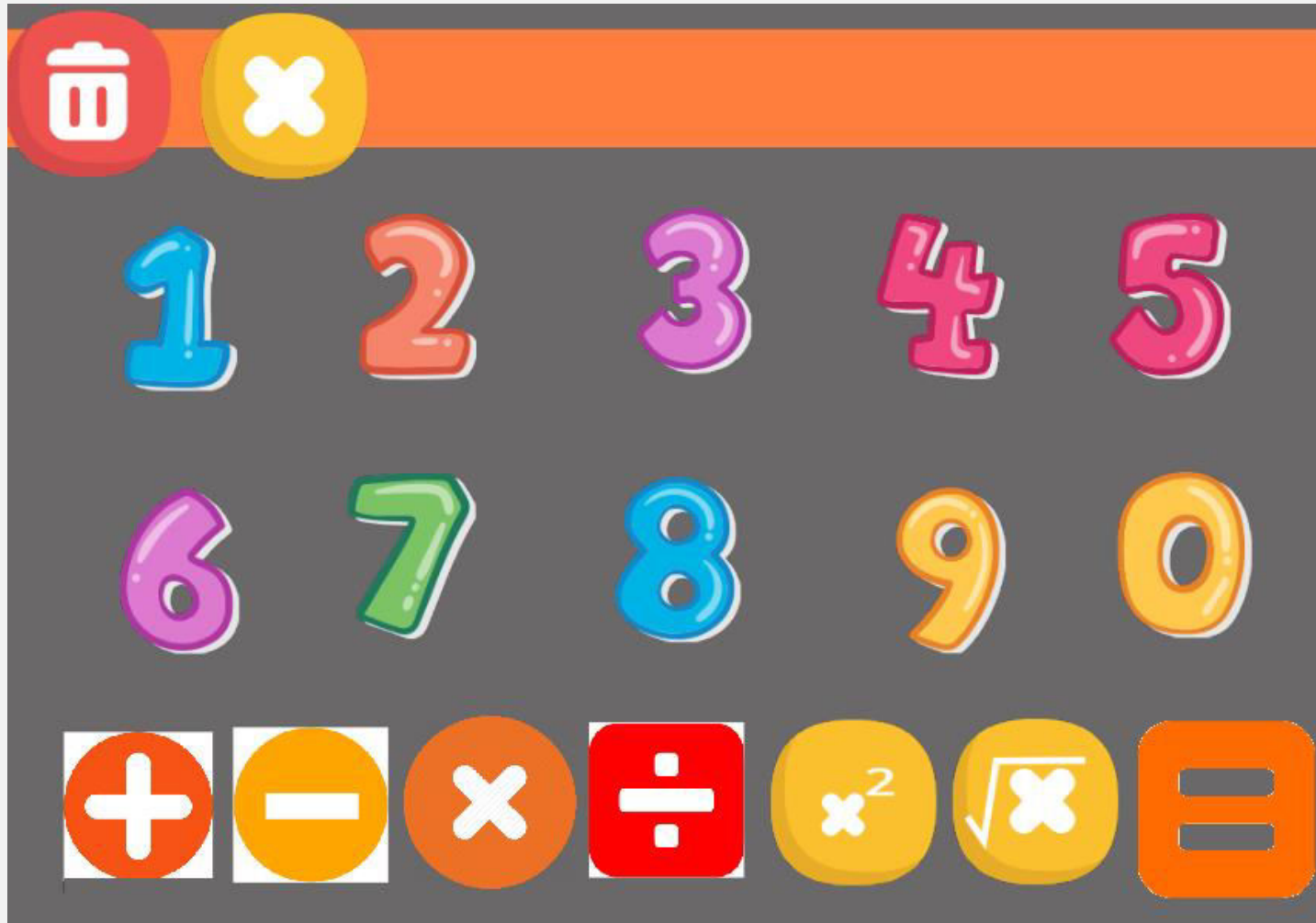
Let's make program in mBlock for “calculator”

Steps to make a program of calculator in mBlock

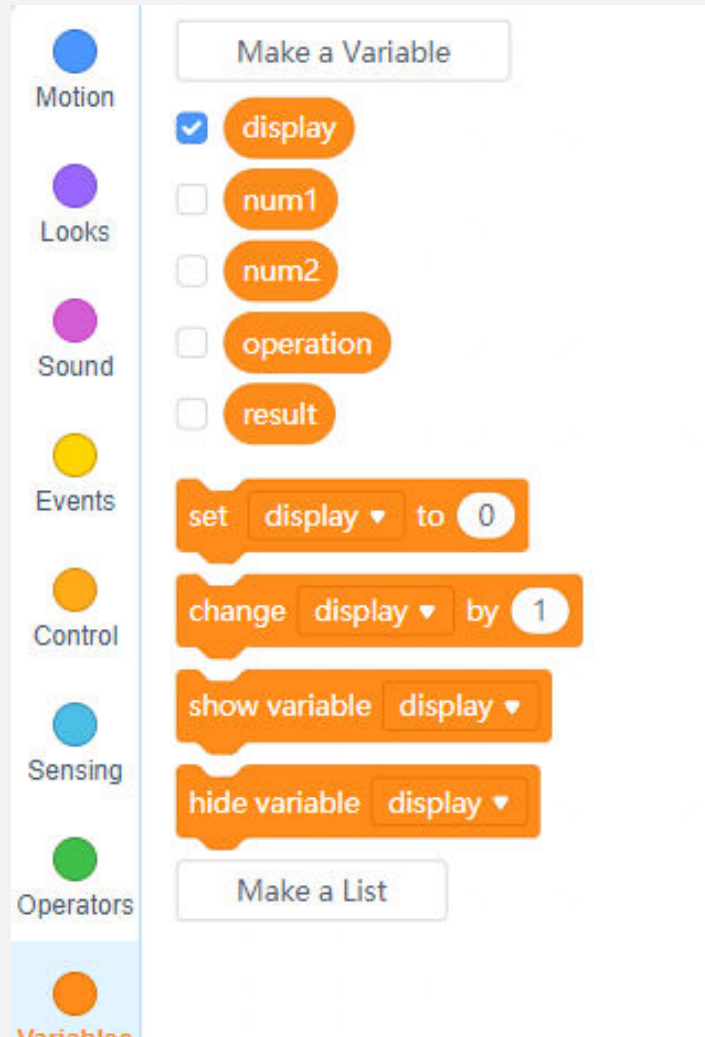
Step 1: Open mBlock Software. Go to Background tab and paint background as follows.



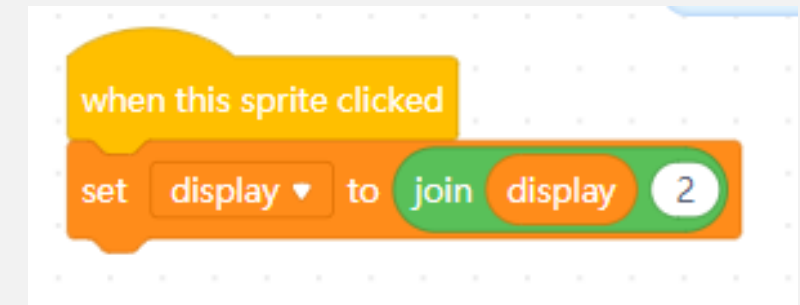
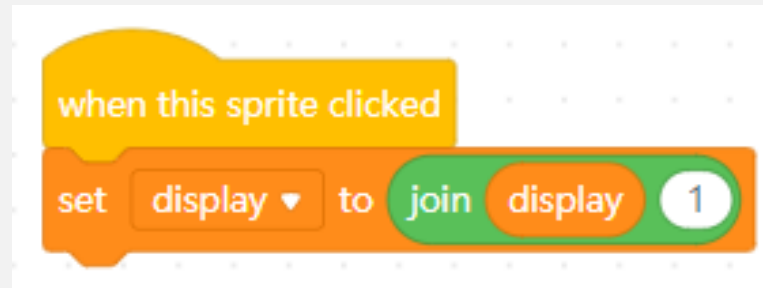
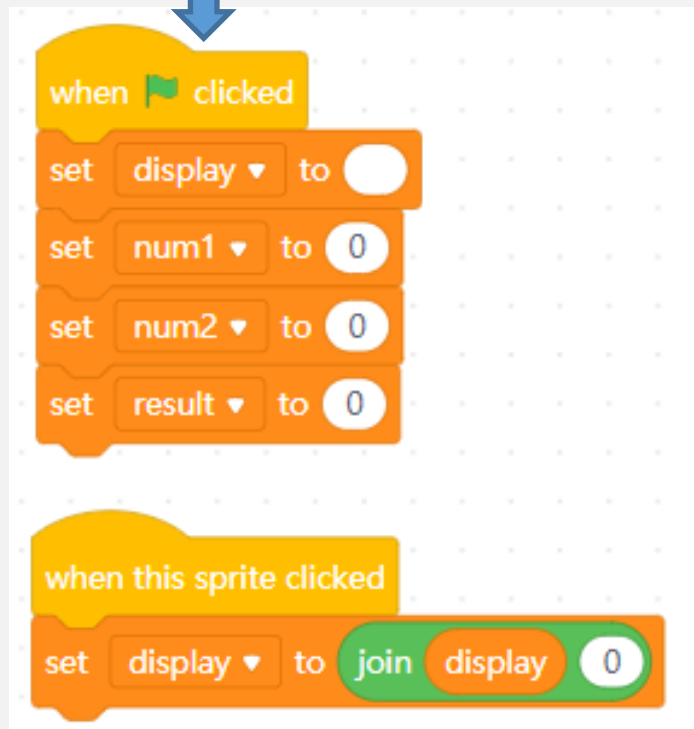
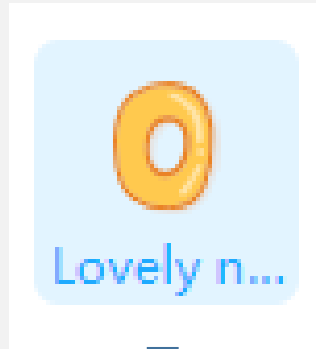
Step2: Now click on the sprite tab and add all the sprites required to make calculator like numbers, operators.(If not present in sprite library make one)



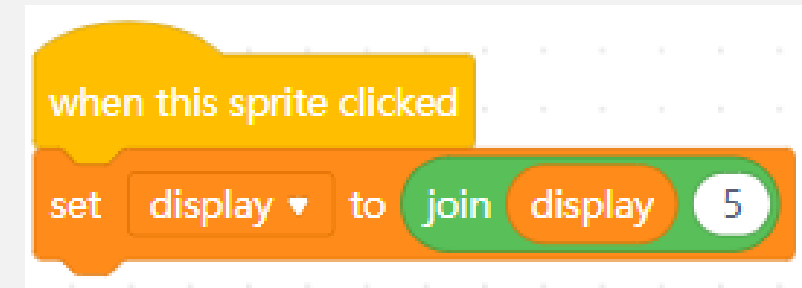
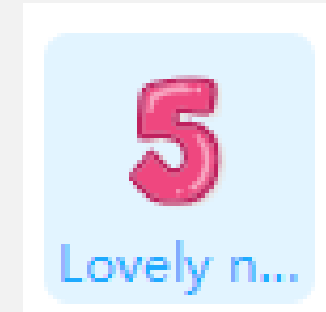
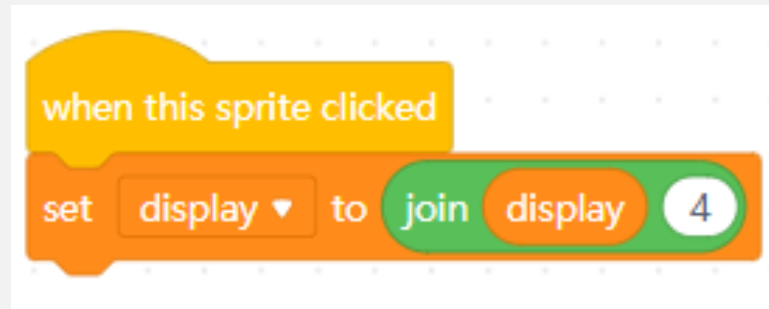
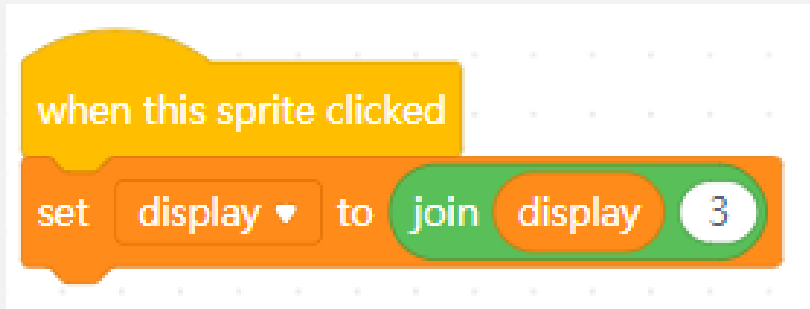
Step 3: Make variables to store numbers, operation type, result and display



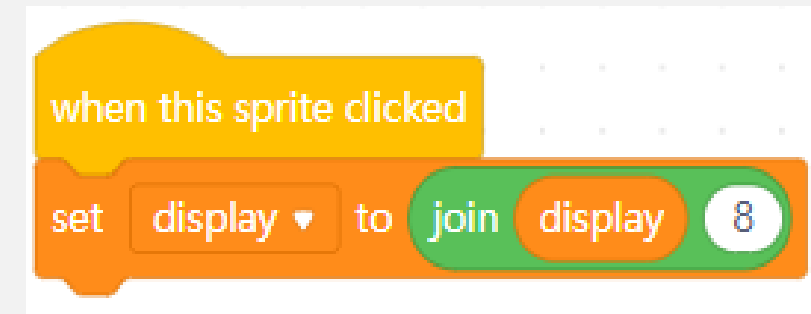
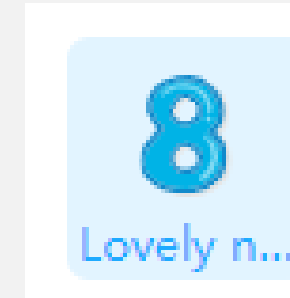
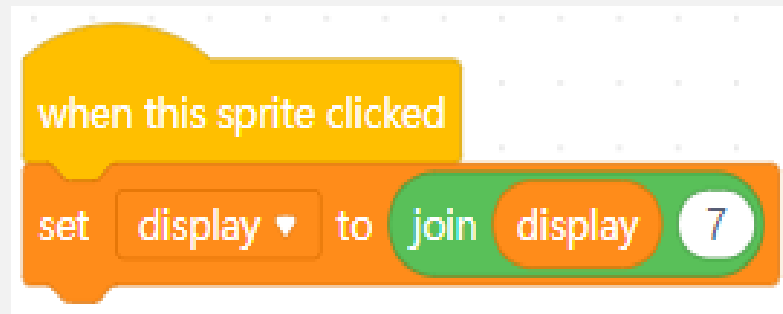
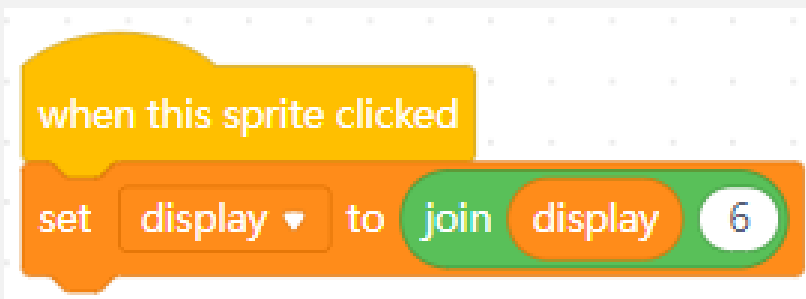
Step 4: Write program for each sprite



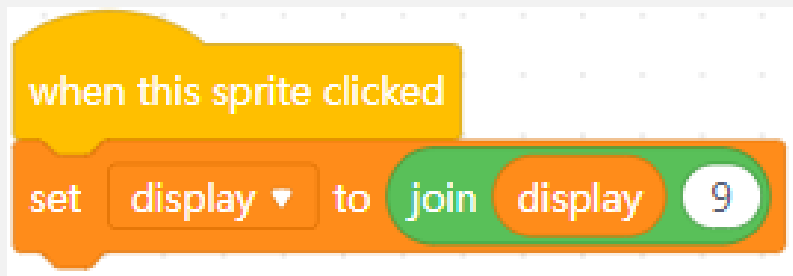
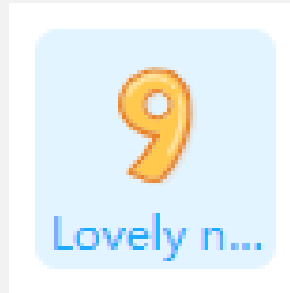
Step 5: Write program for each sprite



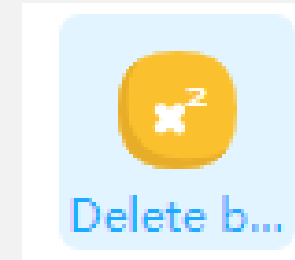
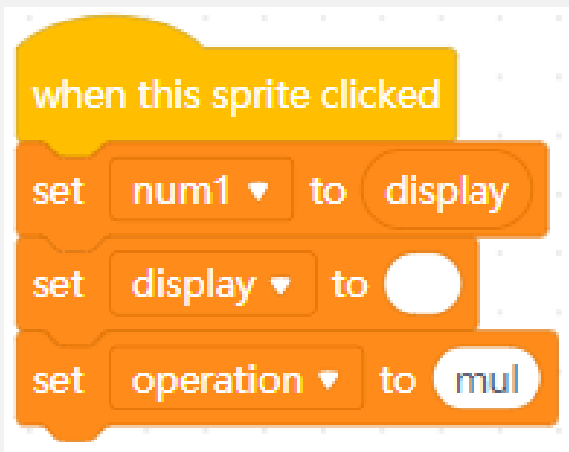
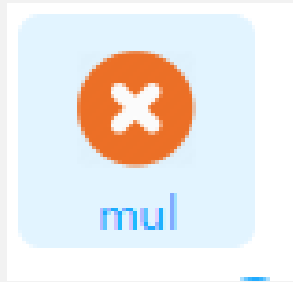
Step 6: Write program for each sprite



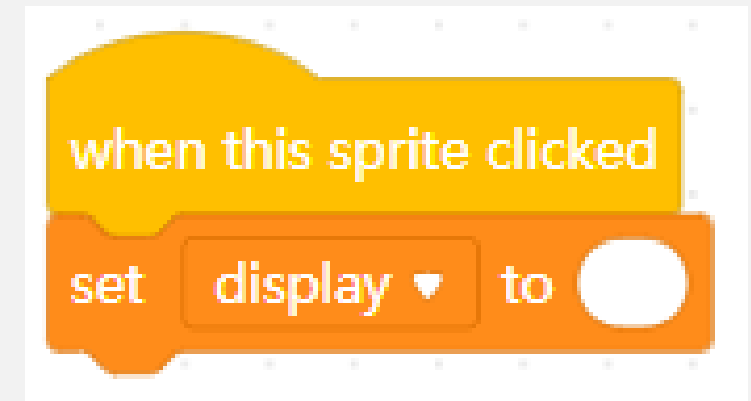
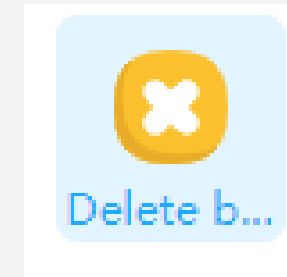
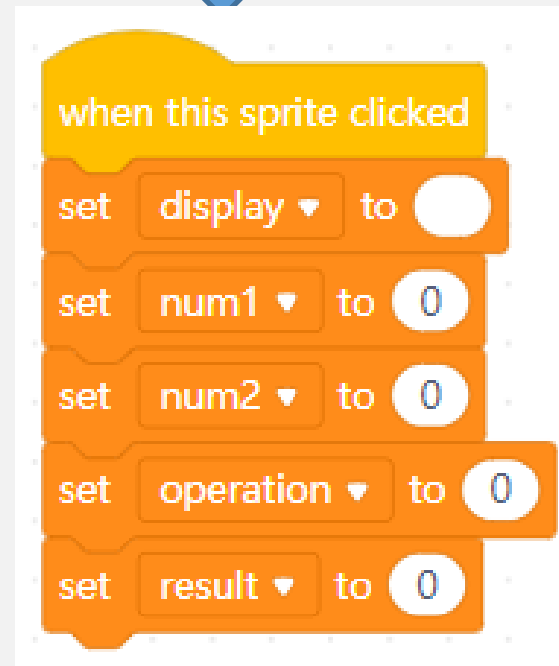
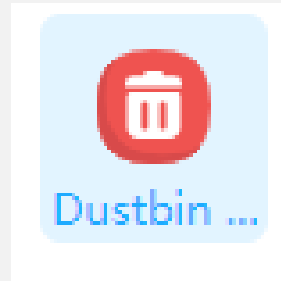
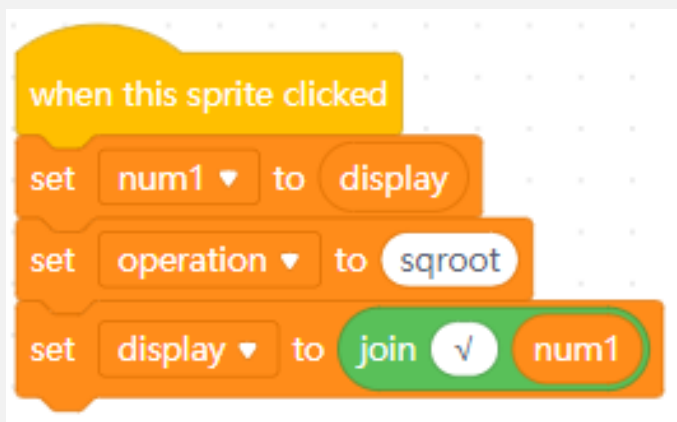
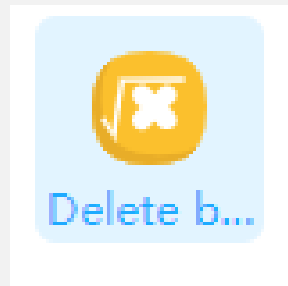
Step 5: Write program for each sprite



Step 5: Write program for each sprite



Step 5: Write program for each sprite



Step 5: Write program for each sprite



```

when this sprite clicked
set num2 to display
if operation = sqr then
set display to join num1 join * join num1 join = num1 * num1
else
if operation = sqroot then
set display to join √ join num1 join = sqrt of num1
else
if operation = div and not num2 = 0 then
set display to join num1 join ÷ join num2 join = num1 / num2
else
if operation = mul then
set display to join num1 join x join num2 join = num1 * num2
else
if operation = add then
set display to join num1 join + join num2 join = num1 + num2
else
if operation = sub then
set display to join num1 join - join num2 join = num1 - num2
if operation = div and num2 = 0 then
set display to Can't divide by 0

```

