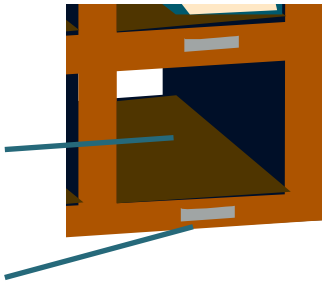
Loops & Variables

Computer Science Principles

VARIABLES

What is a Variable?

- A variable is a named space in memory.
- Think of a mailroom with a large wall of slots for the mail as your memory.
  - This is very simplified of course.
- Each of these slots would be assigned to a variable (by its name) and would hold the values assigned.



Adding Variables

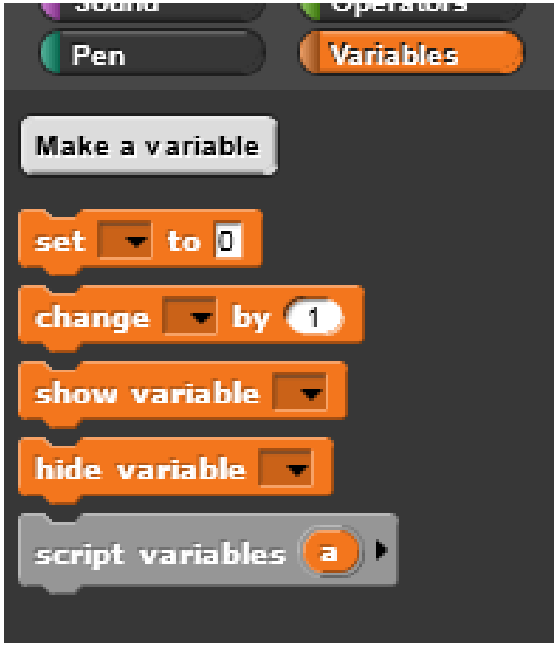
- You can add variables to your program to increase its flexibility.
- The variable allows you to change a value as the script runs.
- To add a variable, select the Variables tab, then click on the Make a Variable button.

Creating a Variable

- When you click on the Make a Variable button, the Variable Name window will open.
- Note you can create the variable for the active sprite only (called a local variable) or for all sprites (called a global variable).

Variable Blocks

- You have multiple variable blocks.
  - Checking the checkbox beside a variable name will display the value of the variable on the stage. (only visible when there is a variable.)
  - set [*variableName*] to ()
    - Sets the value
  - change [*variableName*] by ()
    - Changes the value
  - show variable [*variableName*]
    - Displays the value on the stage.
  - hide variable [*variableName*]
    - Displays the value on the stage.
  - script variables (a)
    - Creates local variables

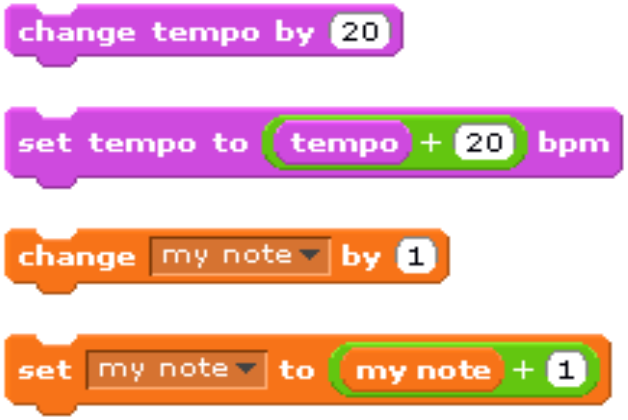


Example of Variable Use

- Create a variable called mynote that will be the value of what note is played.
  - Now the note will change as the loop runs (from using the *repeat ()* block).
  - Note that we had to give the variable a starting value.
    - This is called initializing the variable.

Set vs. Change

- Note that using a set [variableName] to () block will set the value of the variable – NOT update it.
- To update or change a value, use the change [variableName] by () block.



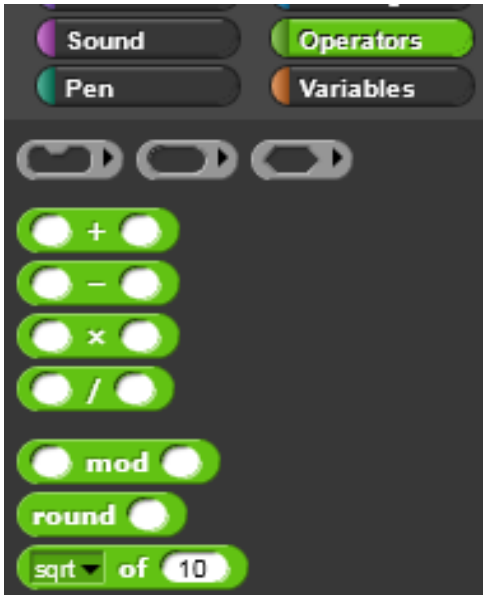
LOOPING

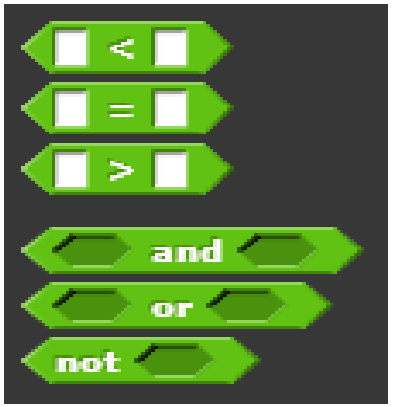
Looping

- There are times when we want certain blocks to repeat more than one time.
- There are blocks that allow us to do just that.
  - *Warp*
    - Does not show the interim steps – only the final product
  - *forever*
    - Will continue to loop until the program closes
    - This is basically an infinite loop as it goes on forever..
  - *repeat ()*
    - Will continue to loop the specified number of times.
  - *repeat until < >*
    - Will continue to loop until the condition is met (*true*)

Helping Blocks

- There are blocks that you will want to use with your variables and loops.
- These blocks are in the Operator's palette.





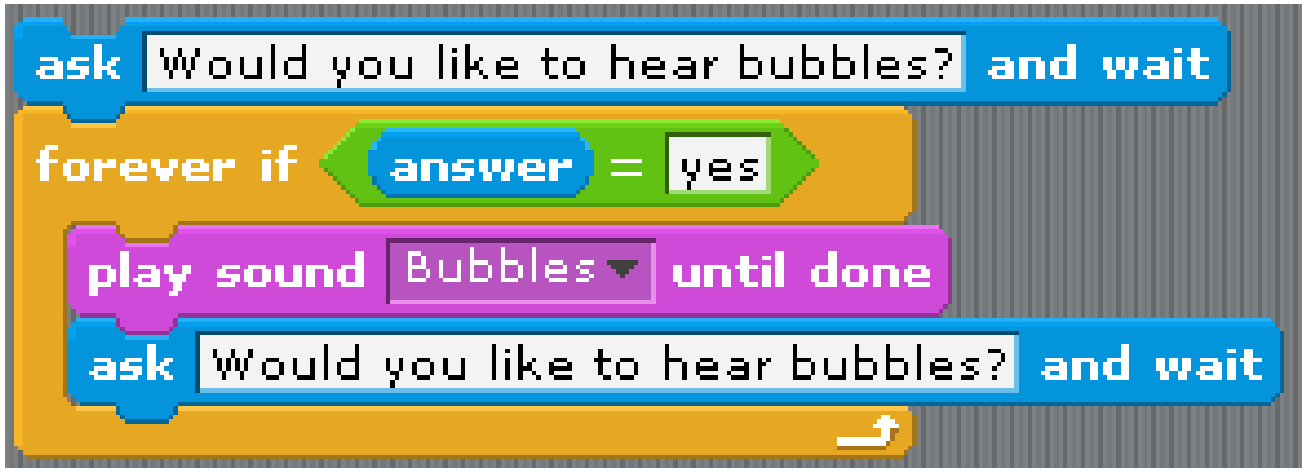
Looping Blocks

- Will continue to play the Bubbles sound.

- Will play the Bubbles sound three times

Looping Example

- Will ask the question, then wait for the answer.
- If the answer is "yes" it will play the Bubbles sound.
- Then ask the question again and wait for the answer.
- Playing and asking the question will continue to loop until the answer is something other than "yes"

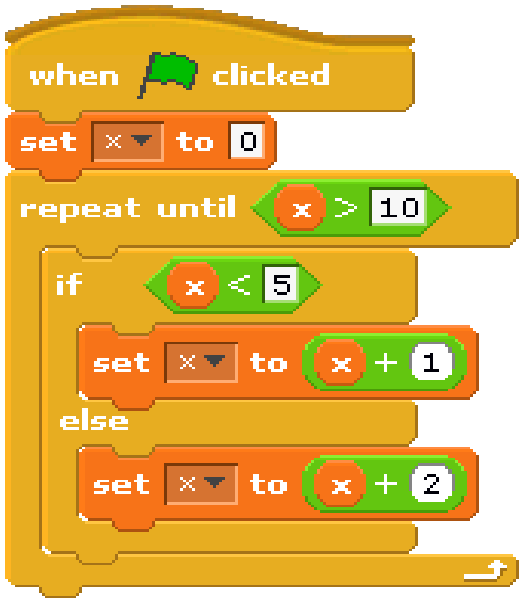


Looping Example

- Let's look at the "repeat until" block a bit closer.
- Just like REPEAT, it will do everything inside the C-shaped block a certain number of times.
- However before it starts the loop each time, it checks to see if the condition (x > 5) is true.
- When this is condition is true, it will not repeat again.

	x
Before the loop	0
Top of loop	0
Bottom of loop	1
Top of loop	1
Bottom of loop	2
Top of loop	2
Bottom of loop	3
Top of loop	3
Bottom of loop	4
Top of loop	4
Bottom of loop	5
Top of loop	5
Bottom of loop	6

repeat until <>



Value of x

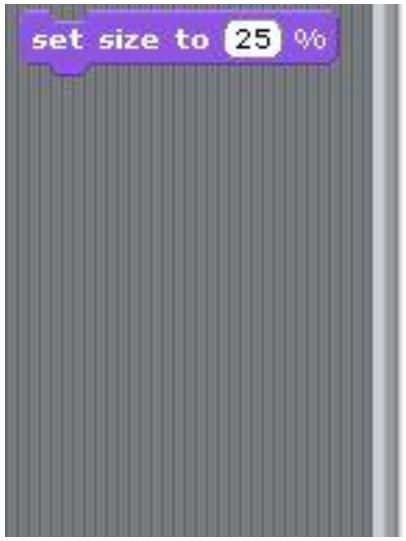
DRAWING BLOCKS

Important Blocks for Drawing

- There are several blocks you will use to draw.
  - *move () steps*
    - Will move your sprite which will draw for you.
  - *turn () degrees*
    - Will turn your sprite to face that direction
  - *clear*
    - Will clear your stage
  - *pen down*
    - Will tell the sprite to start drawing
  - *pen up*
    - Will tell the sprite to stop drawing

Changing Sprite Size

- In order to see your drawing, you might want to change the size of your sprite.
- In the Looks area, you will set the *set size to () %* block.



Where is my Sprite?

- You might also need to know where your sprite is located by the x and y positions as well as the direction your sprite is facing on the stage.
- Look in the Motion area, you will see the several blocks you can use.
- By checking these blocks, the information will be displayed on the stage.