Variables and Lists

Store and recall information to create customized animations, stories, and games

Cards in This Pack

- Reporter Blocks in Scratch
- Brightness Slider
- Interactive Storytelling
- Clone Piano: Using Local Variables
- Clone Piano: Using Global Variables
- Musical List
- Generate a Melody: Repeat through a List
- Generate a Sum: Repeat through a List

Perhaps you have used a variable to store a game score, but did you know a variable can hold numbers or text (also known as a "string")? See these cards for examples of non-score uses for variables and lists. See our in-editor tutorials or other coding cards for instructions on how to set up a basic score.

Reporter Blocks in Scratch

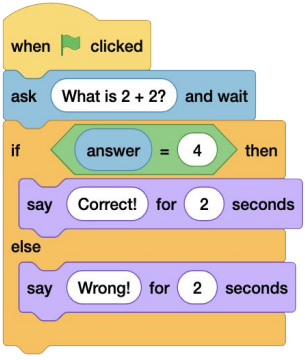
- Variables and lists hold information you can use in your program, but Scratch comes with some built-in reporter blocks that also store information.
- Unlike a stack block, reporter blocks go inside another block to serve as an input.
- You can click on a reporter block in the block palette or in the script area to see the piece of data it currently holds or the value it reports. Or check a box next to many of these reporter blocks to display them on the stage via a stage monitor.

Reporter Blocks in Scratch

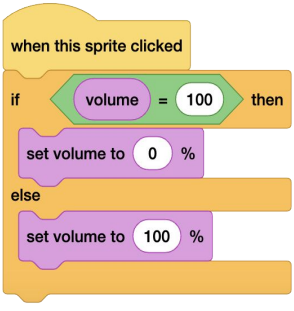
scratch.mit.edu

EXPERIMENT WITH SOME REPORTERS

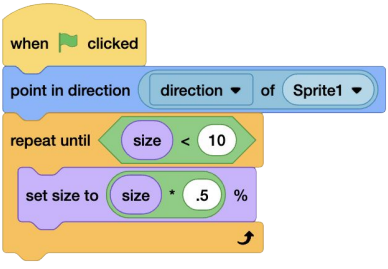
When you use the “ask” block to pose a question to a user, the answer they enter into the dialogue box is stored in a reporter block called “answer.”



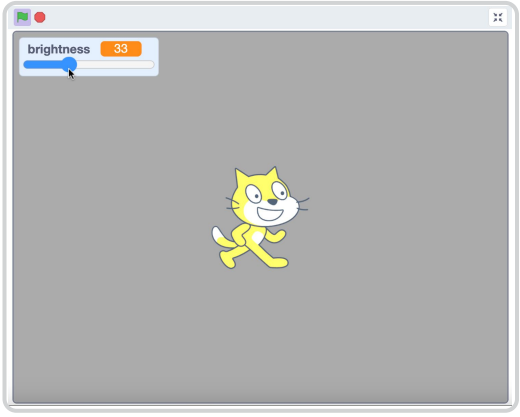
The “volume” reporter block stores the number representing the current volume of the sprite, clone, or stage.



You can use reporter blocks that store the position, direction, and size of sprites to perform calculations or mirror properties.



Brightness Slider



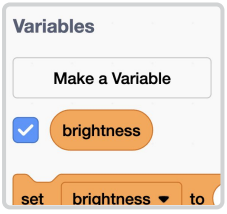
- Scratch comes with some built-in reporter blocks that store information, but what if you want to store and recall information for which there is no reporter block, like the sprite color or brightness?
- Let's create a project where a variable controls the brightness of the sprite.
- And let's put the power in the hands of users, by letting them control the value in the variable with a slider.

Brightness Slider

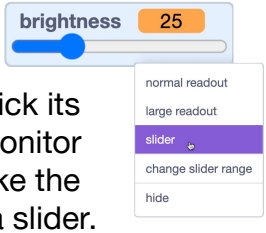
scratch.mit.edu

GET READY

Create a variable called “brightness.”

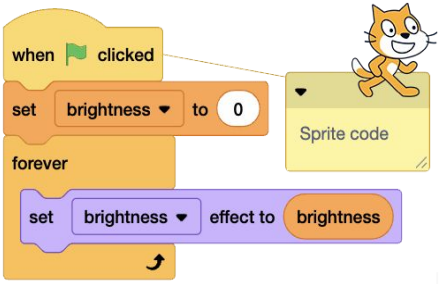


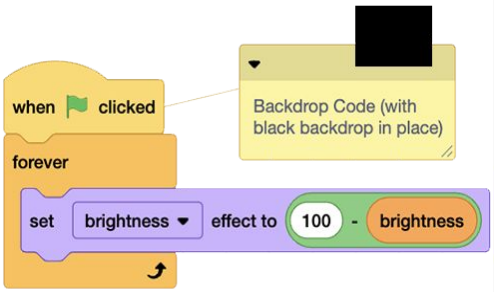
Right-click its stage monitor and make the readout a slider.



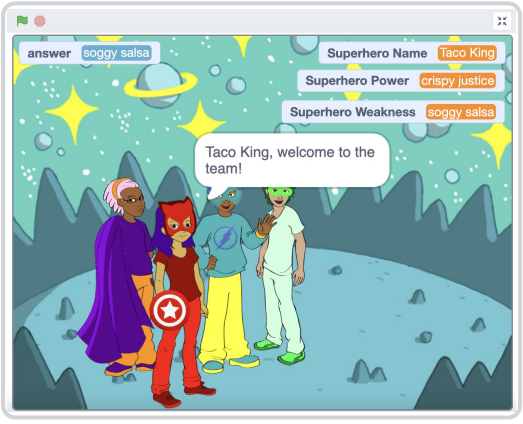
ADD CODE

1. Create a short script that sets the “brightness” variable to 0.
2. Then, have the program forever set the brightness effect to the number in the “brightness” variable. Test it out!
3. Try adding code to make another sprite or the backdrop have the opposite effect.





Interactive Storytelling



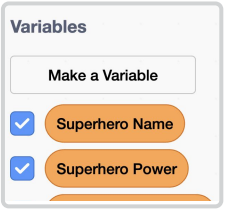
- You can pass information to a variable or list by clicking or moving a sprite, adjusting a slider, via code blocks, and more!
- You can also pass information from one reporter block, variable, or list to another. This could be helpful because variables and reporter blocks can only hold one piece of information at a time.
- Let's create a project that collects the user's answers to multiple questions, and repeats them back in the form of a story!

Interactive Storytelling

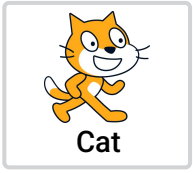
scratch.mit.edu

GET READY

Create multiple variables.



Choose any sprite.



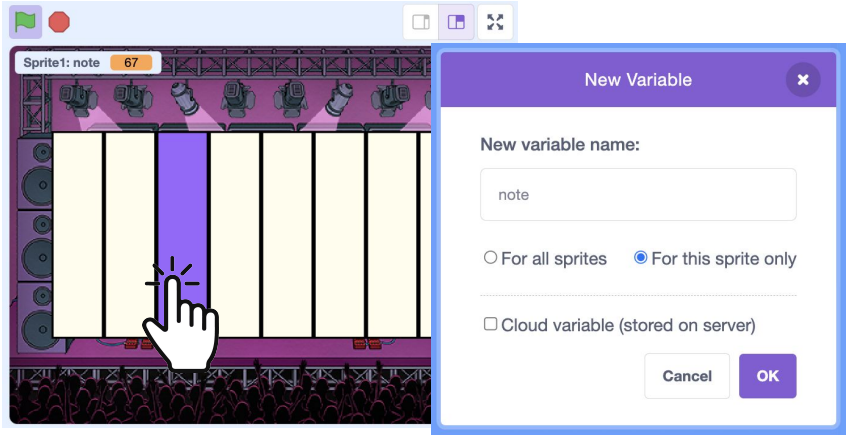
ADD CODE

1. Create individual variables for each answer you are collecting. The “answer” reporter block can only hold one piece of data at a time, so use the “set [variable] to” block to pass the “answer” into a variable for storage after each related question is asked.

2. Use a “say” block and a “join” block to weave each answer stored in a variable into a story sentence.

Alternative: Store and retrieve answers from a list!

Clone Piano: Using Local Variables



A local variable (“For this sprite only”) is individual to a single sprite or a single clone. (Versus a global variable that applies to all sprites in the project and all their clones.)

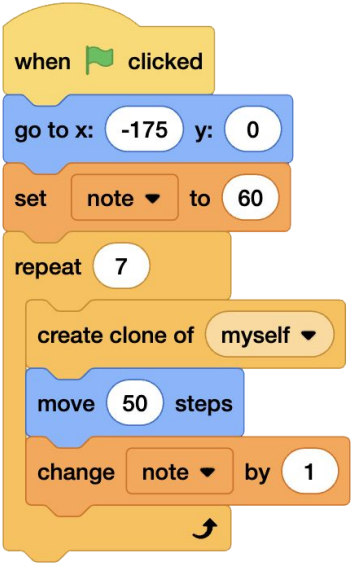
The value stored in each clone’s local variable is the value that was present at the moment the clone was created.

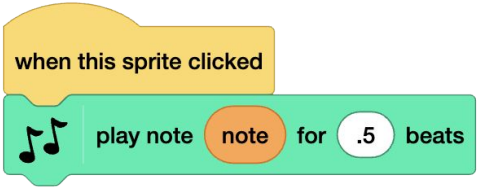
Local variables show the sprite name followed by the variable name in the stage monitor.

Clone Piano: Using Local Variables

scratch.mit.edu

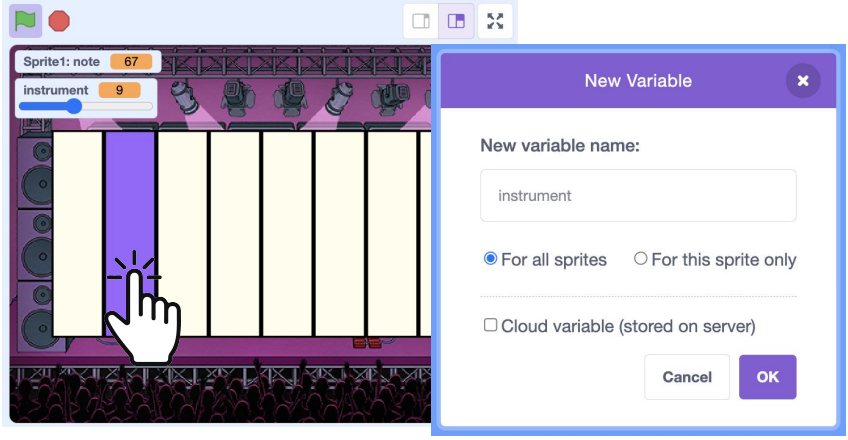
1. Create a piano key sprite by using the Rectangle drawing tool in the Paint Editor.
2. Assemble a script that creates clones of the piano key sprite in a row.
3. Create a local variable (“For this sprite only”) to store the note for each clone and the original sprite.
4. Set the initial note, and then change the note after each clone is created. Use the “note” variable in the “play note” block. Test and debug!





Optional: Adjust your program so it changes the color of the piano key or shows a different costume for the piano key as the note is played, so you can hear and see when a piano key is pressed.

Clone Piano: Using Global Variables



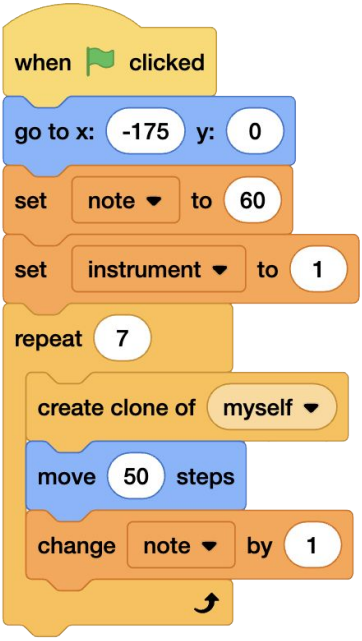
A global variable (“For all sprites”) applies to all sprites in the project and all their clones. (Versus a local variable that is individual to a single sprite or a single clone.)

The value stored can be changed, and the variable is updated for all sprites in a project.

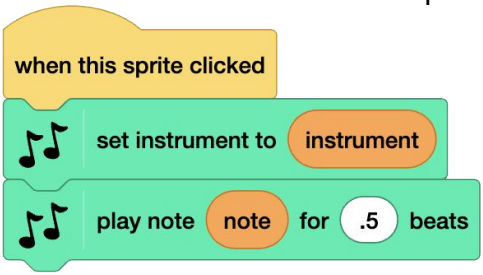
Global variables just show the variable name in the stage monitor.

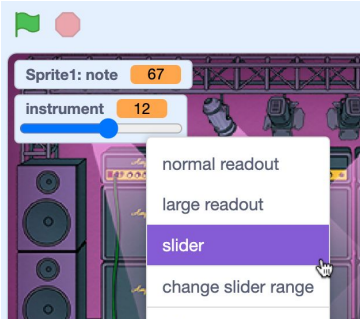
Clone Piano: Using Global Variables

scratch.mit.edu

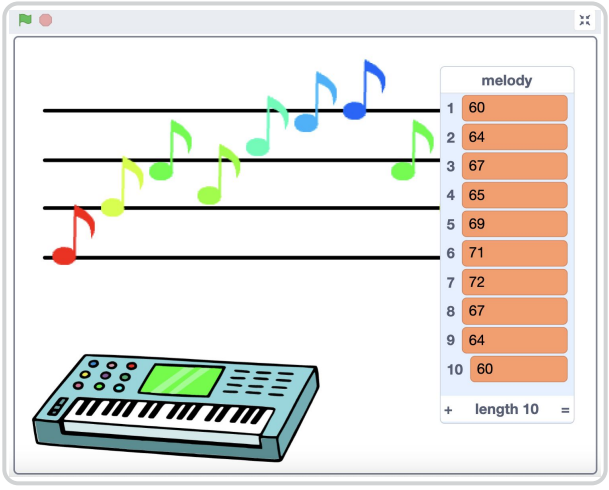


1. Create a global variable (“For all sprites”) to store the instrument for each clone and the original sprite.
2. Set the initial instrument, and then use the “instrument” variable in the “set instrument” block.
3. Right click on the “instrument” stage monitor to change it to a slider. Then, right click again to set the range to 1-21 (the number of instruments available). Now, use the slider to change the instrument globally, for all piano keys. Test and debug!





Musical List



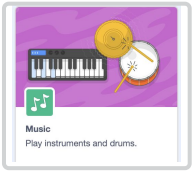
- You can use a predefined list to determine animation.
- Try creating a melody project, storing song notes in a list that create a musical score that can be played.
- As a bonus, you can use the Pen extension to stamp notes on a scale and produce a visual representation of your musical score.

Musical List

scratch.mit.edu

GET READY

Add the Music extension.



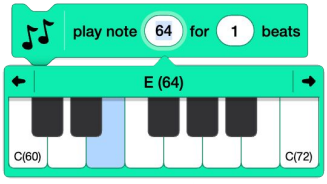
Choose any sprite.

Keyboard

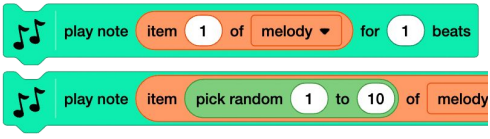
ADD CODE

1. Create a list. Add song notes to the list via the stage monitor (add rows manually and type note numbers in) or by using the “add to [list]” block.

   You can find note numbers by clicking on the input of the “play note” music block.

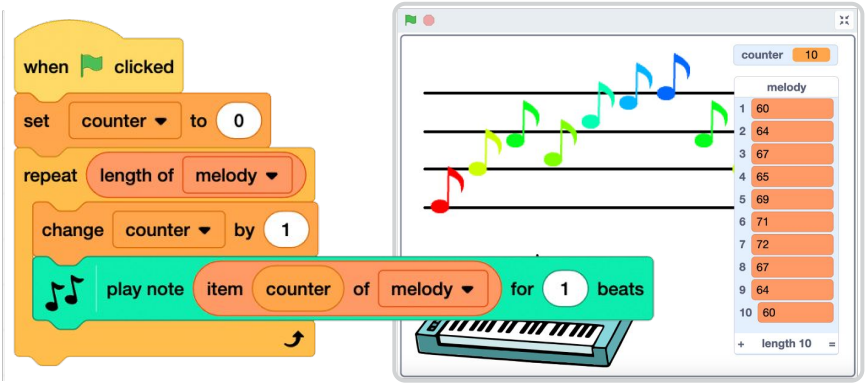


2. Write a script to play each note on the list by item number, or let the program pick the note to play randomly.



See the next card to learn how to create a “counter” variable to automate moving/repeating through the list in order.

Generate a Melody: Repeat through a List



- While there is no “next item of list” block, you can create a script that loops through the items of a list in order.

  The ability to automate moving or repeating through a list can speed up your coding process and make editing scripts quicker.

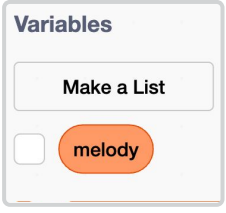
- This can be useful if you want to add items of a list together, speak or say items in a list, etc.

Generate a Melody

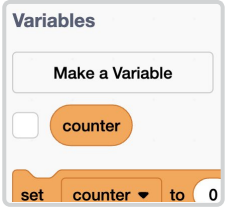
scratch.mit.edu

GET READY

Create a list.



Create a variable.

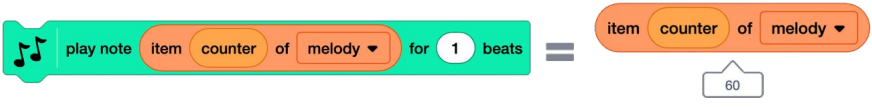


ADD CODE

Step through the code on the card front to see what it does:

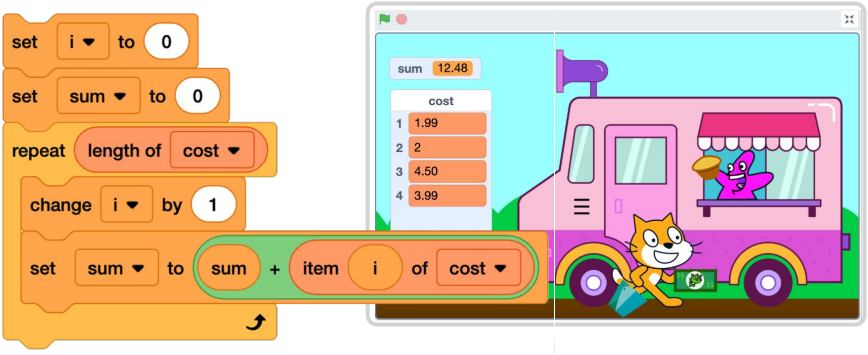
1. Changes the “counter” variable (that stores a number to represent an item number on the list) by one.

2. Plays the note number associated with that item number (the number entered on that line of the list). Note: This is why it is important to first set “counter” to zero first each time the program runs.



3. Repeat as many times as there are rows in the list/for the length of the list.

Generate a Sum: Repeat through a List



- While there is no “next item of list” block, you can create a script that loops through the items of a list in order.

  The ability to automate moving or repeating through a list can speed up your coding process and make editing scripts quicker.

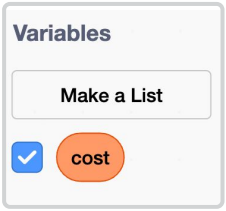
- This can be useful if you want to add items of a list together, speak or say items in a list, etc.

Generate a Sum

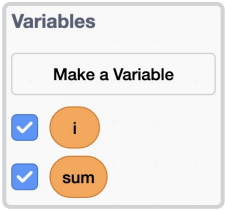
scratch.mit.edu

GET READY

Create a list.



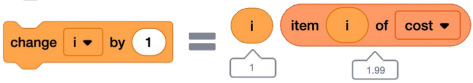
Create two variables.



ADD CODE

Step through the code on the card front to see what it does:

1. Changes the “i” variable (that stores a number to represent an item number on the list) by one.



2. Adds the amount associated with that item number (the number entered on that line of the list) to the value already stored in the “sum” variable, creating a new “sum” value. Note: This is this why it is important to set “sum” and “i” to zero first each time the program runs.

3. Repeat for the length of the list.