

Tiny Bubble Sort



The image shows a window titled "Tiny Bubble Sort" with a light blue background. In the top-left corner, there is a blue window control icon. In the top-right corner, there are green and red window control icons. The main content area has a light orange background. On the left side, there is a cartoon penguin with a white body, black wings, and a black beak, standing on a white ice floe. Above the penguin, there are three light blue bubbles of increasing size. In the center-right, there is a gray box titled "My List" containing a list of five numbers: 5, 9, 10, 22, and 41. Each number is displayed in a white box with a red border. Below the list, there is a plus sign and the text "length: 5". At the bottom of the window, there is a white ice floe and a blue body of water.

My List	
1	5
2	9
3	10
4	22
5	41

+ length: 5

Tiny Bubble Sort uses a *bubble sorting algorithm* to sort five pieces of data input by the user.

Create a background and sprite for decorative purposes only. In Data, create a variable, *n*, and a list, My List. The list will consist of exactly 5 items.

Write a green flag script to delete previous items from the list. Then ask the user to input, one at a time, 5 numbers. Each answer is added to the list.

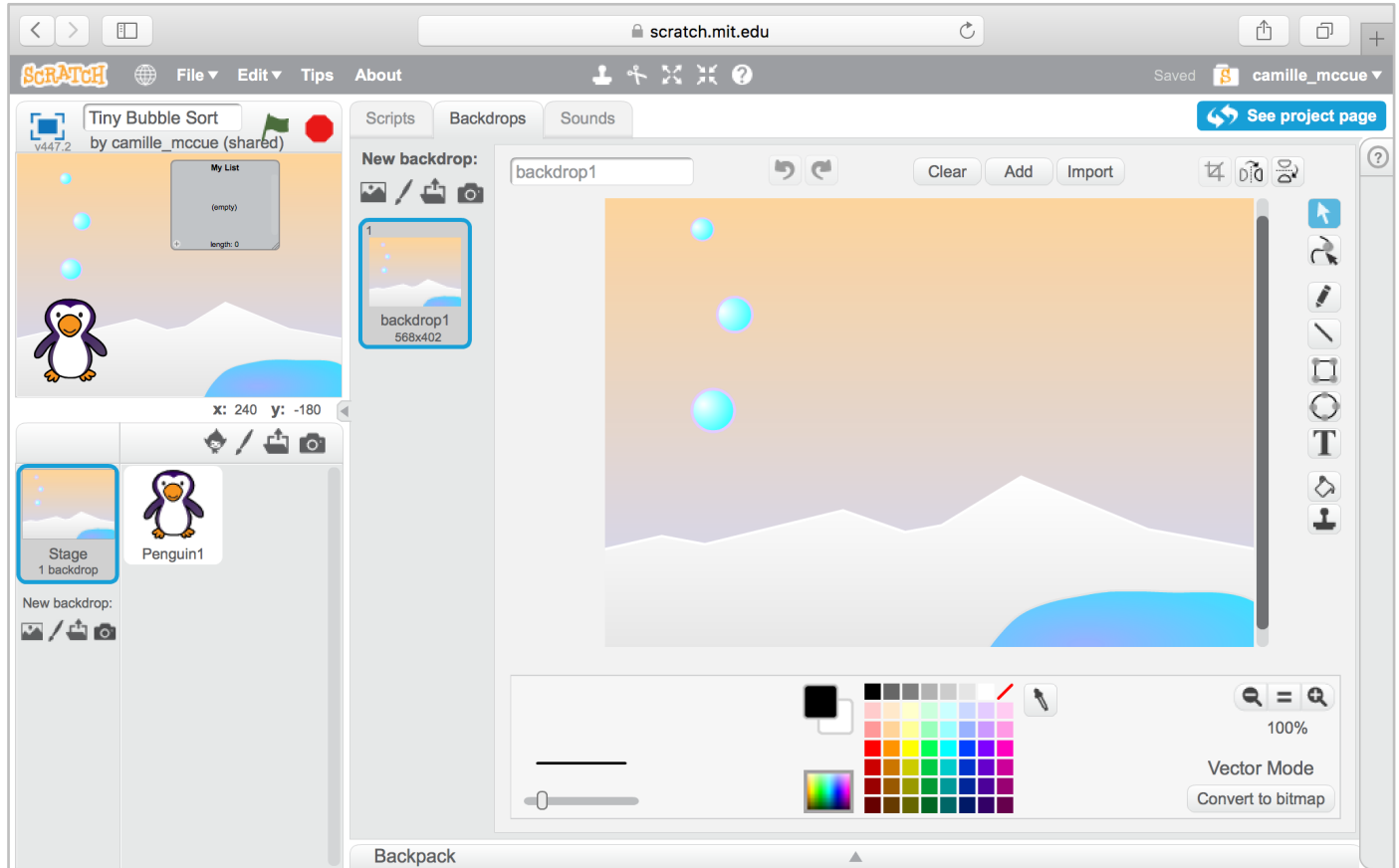
Write a sort script, executed by pressing the "s" key, that compares the first and second items, and orders them from small to large. Repeat this process, moving on to the second and third items; then the third and fourth items, etc. Finally, repeat this pairwise sorting a total of four times to completely order the list from smallest to largest.

To start the action: Press the green flag.

To sort the list: Press the "s" key.

Tiny Bubble Sort

Stage – Backdrop

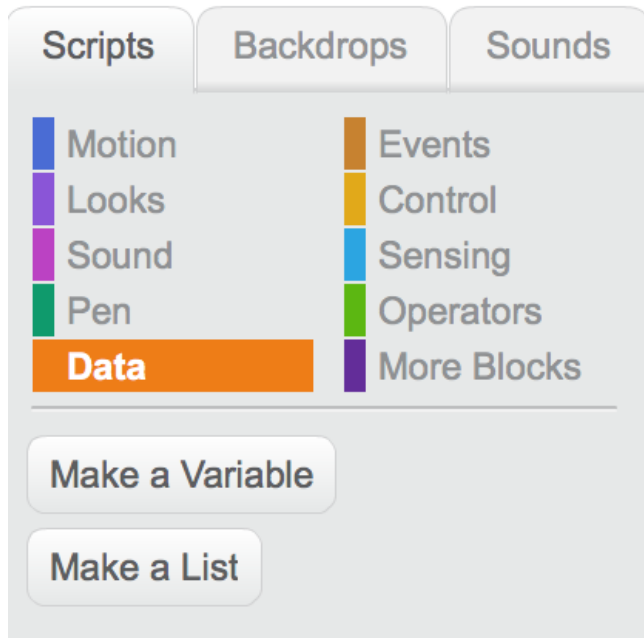


Penguin Sprite – Costume



Tiny Bubble Sort

Penguin Sprite – Variable and List

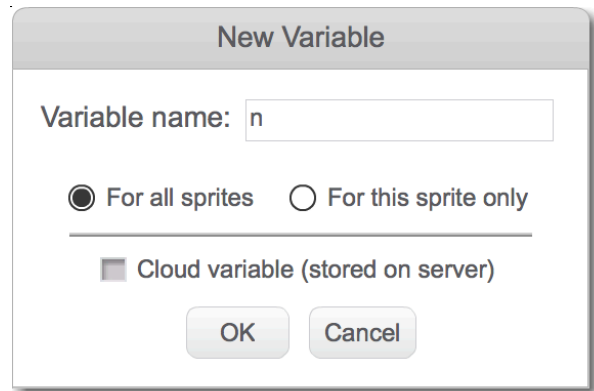


The image shows the Scratch Scripts menu. The 'Data' category is highlighted in orange. Below the menu are two buttons: 'Make a Variable' and 'Make a List'.

- Scripts
- Backdrops
- Sounds
- Motion
- Looks
- Sound
- Pen
- Data**
- Events
- Control
- Sensing
- Operators
- More Blocks

Make a Variable

Make a List



The 'New Variable' dialog box is shown. The variable name is 'n'. The radio button for 'For all sprites' is selected. The 'Cloud variable (stored on server)' checkbox is unchecked. The 'OK' and 'Cancel' buttons are visible.

New Variable

Variable name:

For all sprites For this sprite only

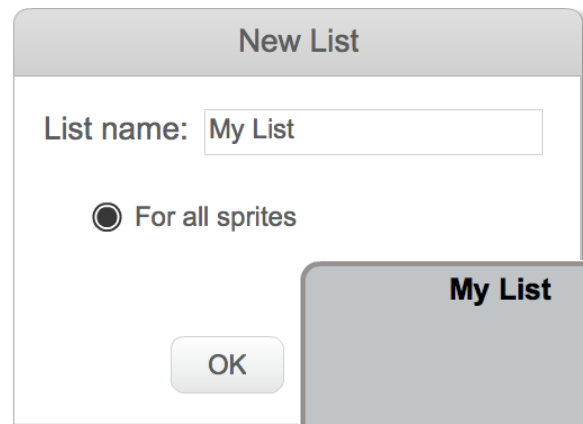
Cloud variable (stored on server)

OK Cancel



The variable monitor for 'n' is shown with the value 5.

n 5



The 'New List' dialog box is shown. The list name is 'My List'. The radio button for 'For all sprites' is selected. The 'OK' button is visible.

New List

List name:

For all sprites

OK



The list monitor for 'My List' is shown. It is currently empty. The length is 0.

My List

(empty)

length: 0

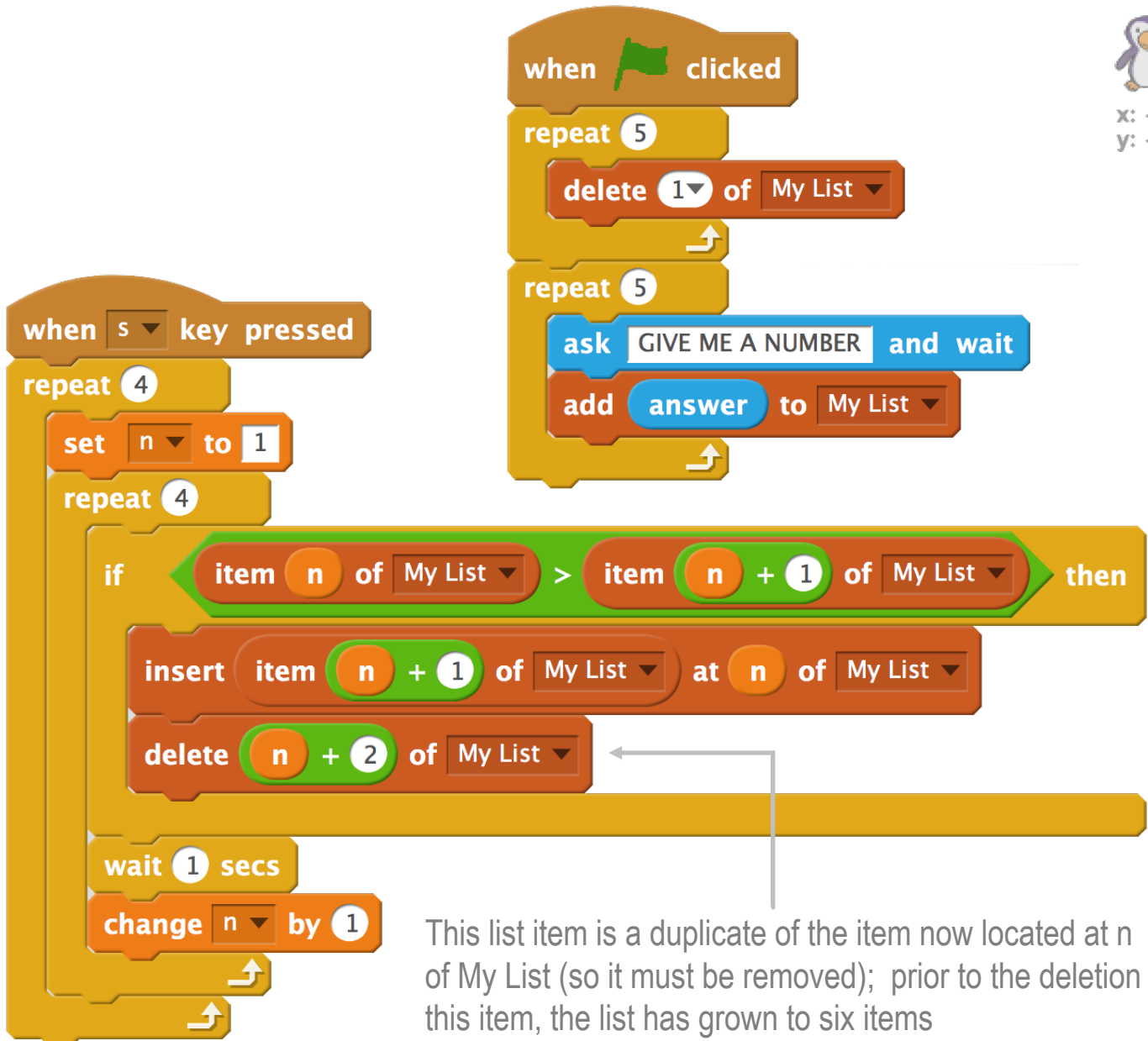
Penguin Sprite – Scripts

The screenshot shows the Scratch IDE interface. The project title is "Tiny Bubble Sort" by camille_mccue (shared). The stage shows a penguin sprite with a speech bubble that says "GIVE ME A NUMBER". A variable named "My List" is visible, containing the values 22 and 10. The Scripts area is open, showing the following code blocks:

- when green flag clicked
- repeat 5
- delete 1 of My List
- repeat 5
- ask GIVE ME A NUMBER and wait
- add answer to My List
- when s key pressed
- repeat 4
- set n to 1
- repeat 4
- if item n of My List > item n + 1 of My List then
- insert item n + 1 of My List at n of My List
- delete n + 2 of My List
- wait 1 secs
- change n by 1

Tiny Bubble Sort

Penguin Sprite – Script Closeups



This list item is a duplicate of the item now located at n of My List (so it must be removed); prior to the deletion of this item, the list has grown to six items

Extension: Can you think of a way to, through abstraction, make this bubble sorting algorithm more general so that a user can vary the length of My List on every use of the app?

Algorithmic efficiency is the property of an algorithm which measures the amount of computational resources used by the algorithm. Better efficiency means faster processing time – something that may matter greatly when the data set is large. What is the algorithmic efficiency of a bubble sort? How does this compare with other sort methods – is it a “fast” or “slow” method?