

Sprite wants to remember ...

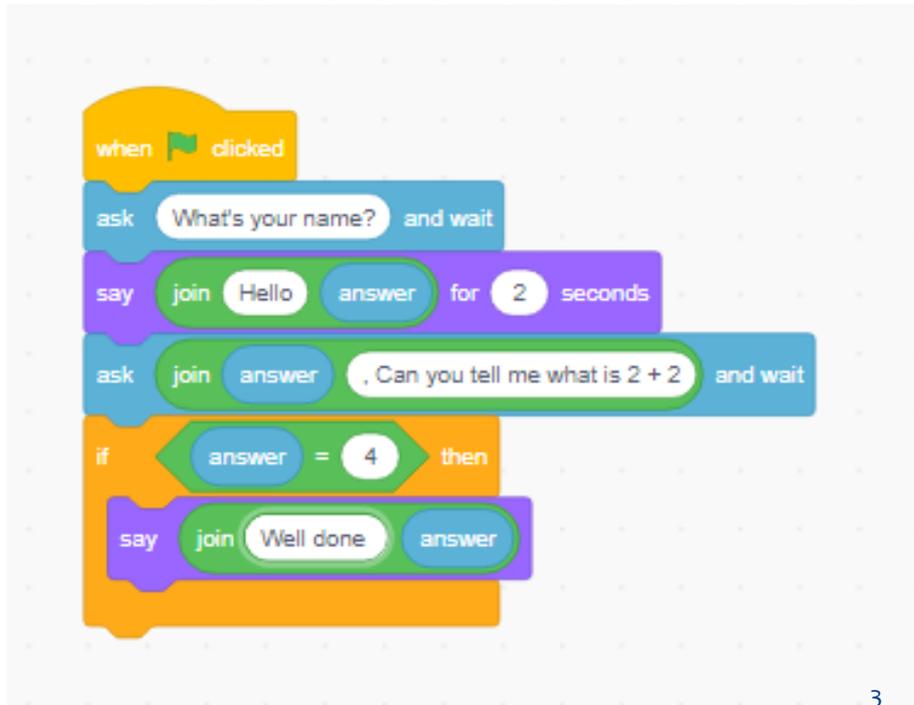
Vineet Srivastava

In this lesson, we will ...

- * Learn about variables:
 - * What variables are, how they work?
 - * How variables can be defined, set and changed?
 - * How we can see the value of variables?
 - * The use of sliders
- * Use this knowledge to make a simple PAINT activity.

Let's start with an example

- * In an earlier lesson, sprite asked us a question.



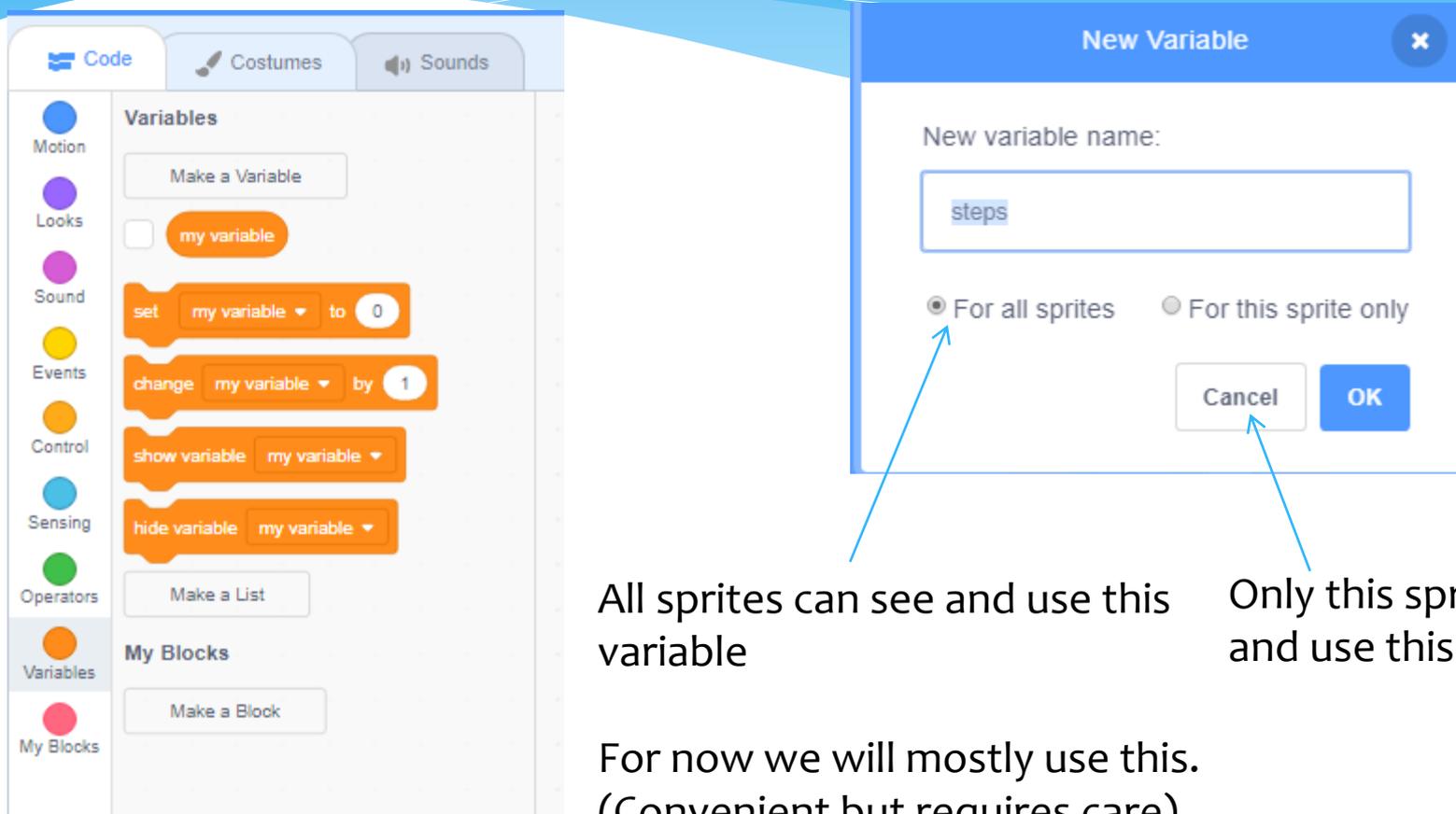
When I answered 4, I thought sprite will answer Well done Vineet, but it answered well done 4 😞 ... sprite totally forgot me ...

How can I make sprite remember?

- * The answer is to use a variable.
 - * This is an extremely important programming concept.
- * Think of variables as small boxes in which the sprite can store bits of information. Variables are like little containers with labels. Analogy: A pencil box.
- * By storing this information, it can ‘remember’.
- * As the program runs, it can keep changing what is ‘inside’ these containers, and use this information for various tasks.



How to create a variable



The image shows the Scratch interface. On the left is the 'Variables' palette with a 'Make a Variable' button. On the right is the 'New Variable' dialog box. The dialog box has a title bar 'New Variable' with a close button. It contains a text input field with 'steps' entered. Below the input field are two radio buttons: 'For all sprites' (which is selected) and 'For this sprite only'. At the bottom of the dialog are 'Cancel' and 'OK' buttons. Blue arrows point from the text below to the 'For all sprites' radio button and the 'Cancel' button.

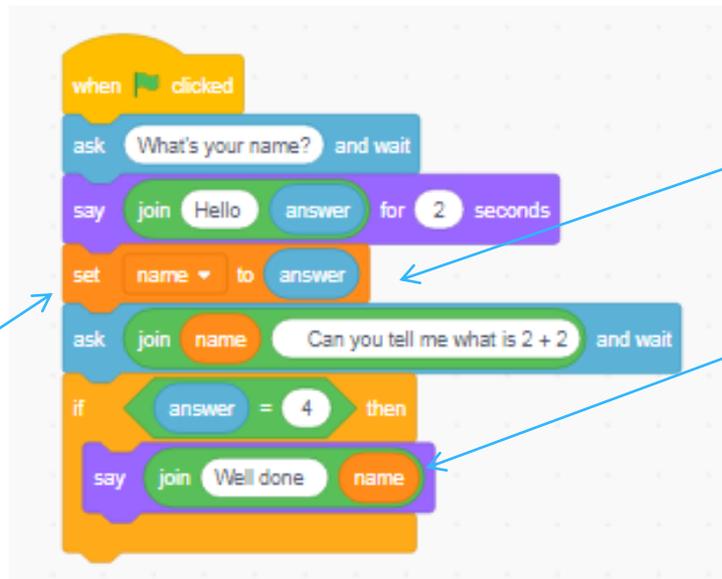
All sprites can see and use this variable

Only this sprite can see and use this variable

For now we will mostly use this.
(Convenient but requires care)

A variable called name

- * Let us now see the previous example with a small modification.



Sprite has an additional container

We will learn more about 'SET' statement in a coming slide.

Notice, here we replaced answer with variable 'name'

Now sprite can remember my name ... 😊

Variables that store numbers

- * Variables can store names (text/sentence/words) or even numbers.
- * However, a very large number of applications, we will be using variables to store numbers.
 - * For example, keeping score, keeping time, remembering how many steps something has to move, deciding if something has to be done and so on.
- * Hence, for now we will focus on only variables that store numbers.

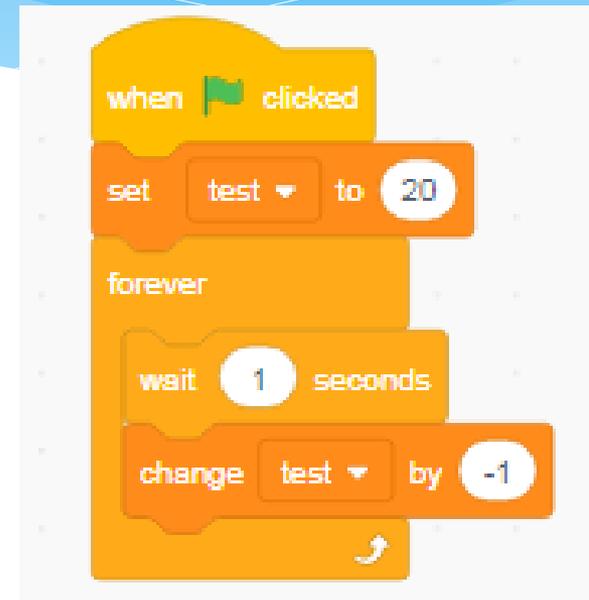
Set variable vs change variable

- * Set sets the value of the variable to a fixed known value.
- * CHANGE changes the value of that variable by some amount.
- * In general (though not always), we use SET in the beginning and change during the program.
- * This is much the same as what we did with the PEN SIZE and PEN COLOUR in the DRAW activity. (Link to the video)

Examples: Set variable and change variable



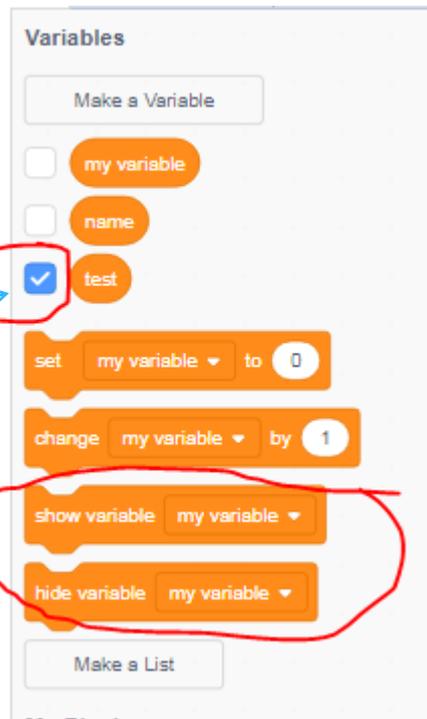
In this example, the variable 'test' is set to 0 when flag is clicked. Thereafter, every 1 second, its value is *changed* by 1. That is, it is *increased* by 1 every second.



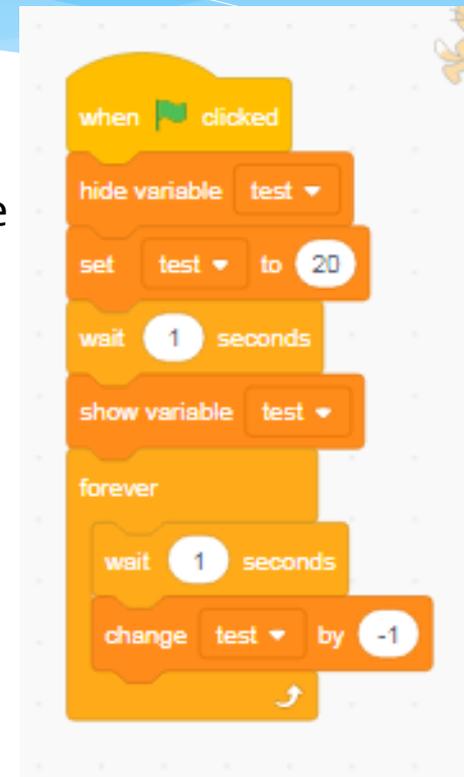
In this example, the variable 'test' is set to 20 when flag is clicked. Thereafter, every 1 second, its value is *changed* by -1. That is, it is *decreased* by 1 every second.

How to display a variable

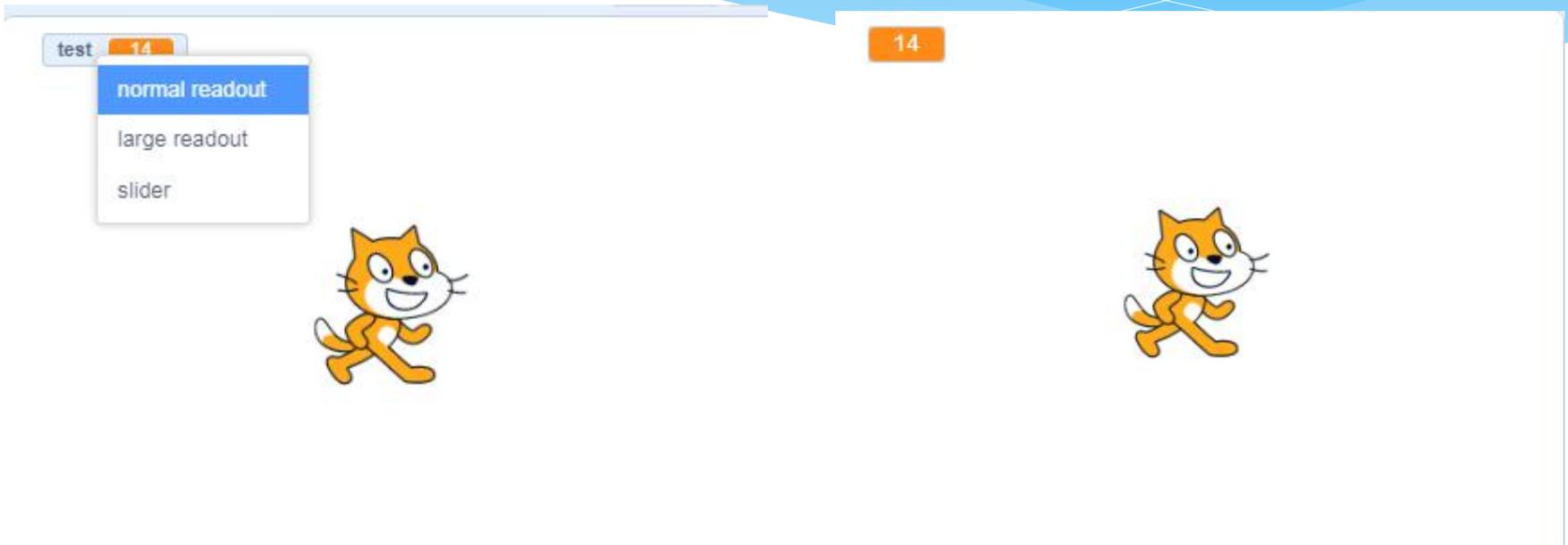
Click on this tick mark



We can use these blocks within a code



How to display a variable (Normal and large readout)



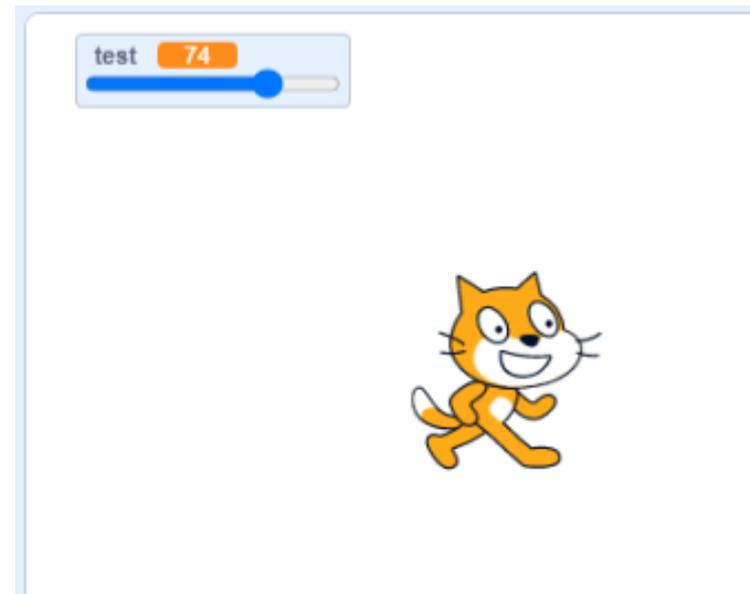
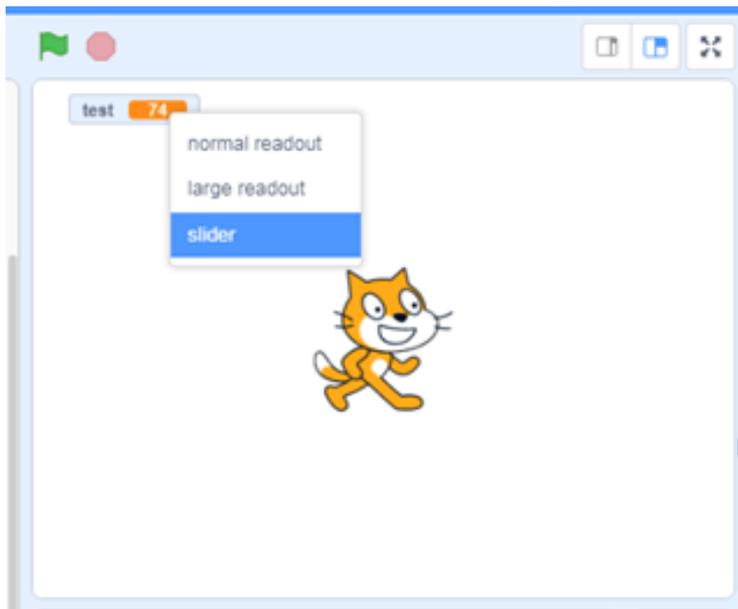
normal readout

large readout

You can move and place this anywhere on the stage.

Slider Readout (for variables that store numbers)

- * Moving the slider has the effect of 'SET' variable.



An example of slider readout

- * Moving the slider has the effect of 'SET' variable.

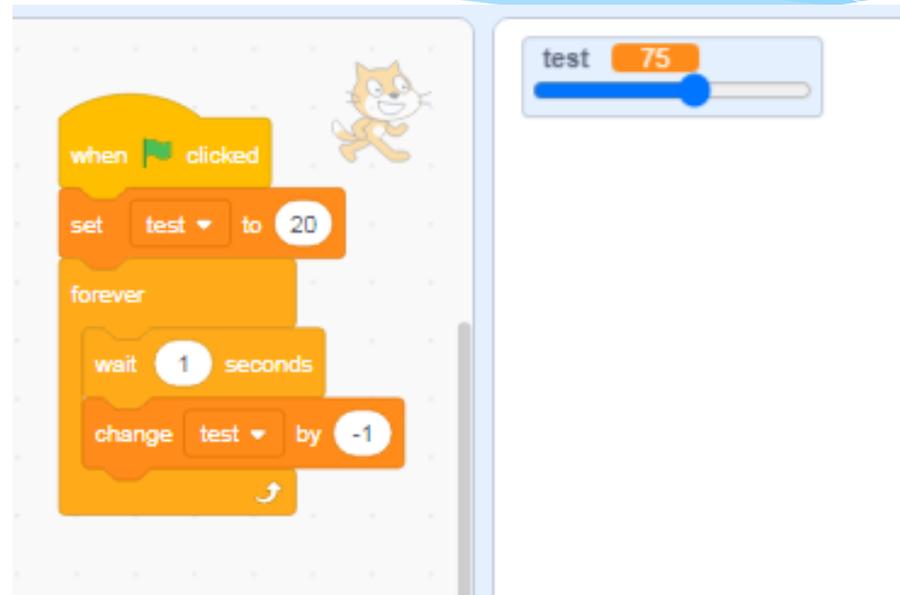
In this case,

Test is set to 20 when flag is clicked.

Thereafter every 1 second it decreases by 1.

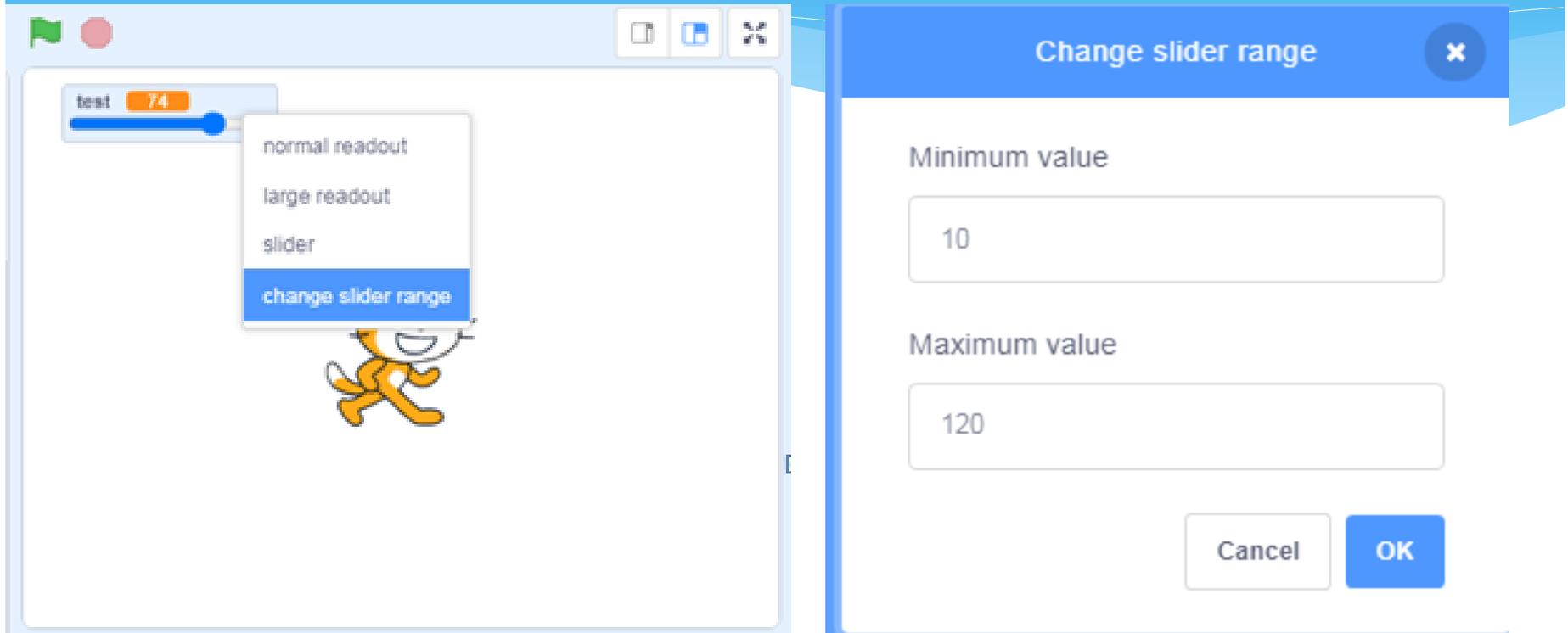
However, if we change the slider, test takes on a new value, and starts decrementing from that value.

In fact, in this case we can skip the initial SET also.



TRY THIS YOURSELF TO GAIN FAMILIARITY

Variable Readout Range

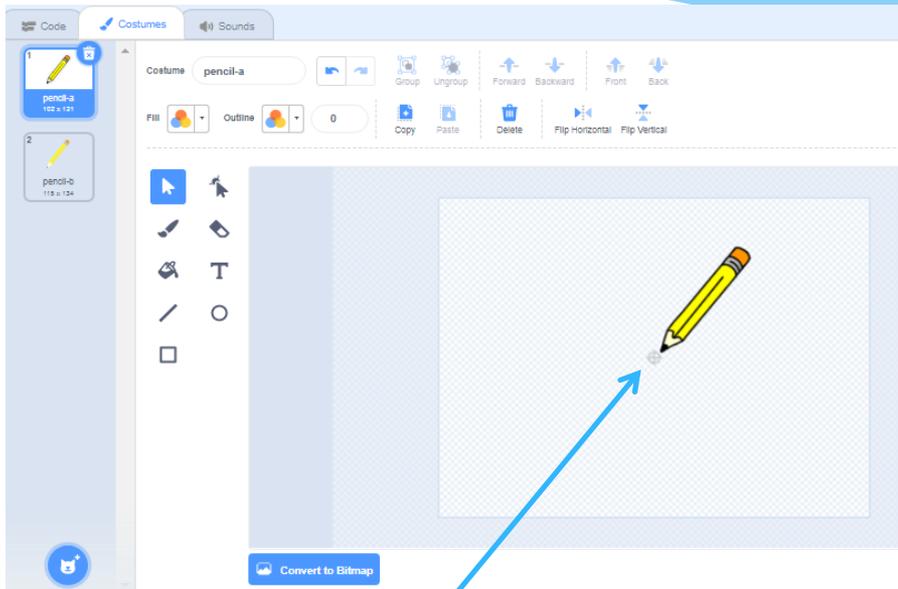


By doing this, we limit the range of values that the variable can take through the slider.

Using variables in a Paint Activity

- * Let us now make a paint box ...

Pencil Sprite ...



You can use any sprite for this, but generally speaking, a small, ball or dot shaped sprite, or a sprite with a sharp tip, which we can offset from centre, works better.

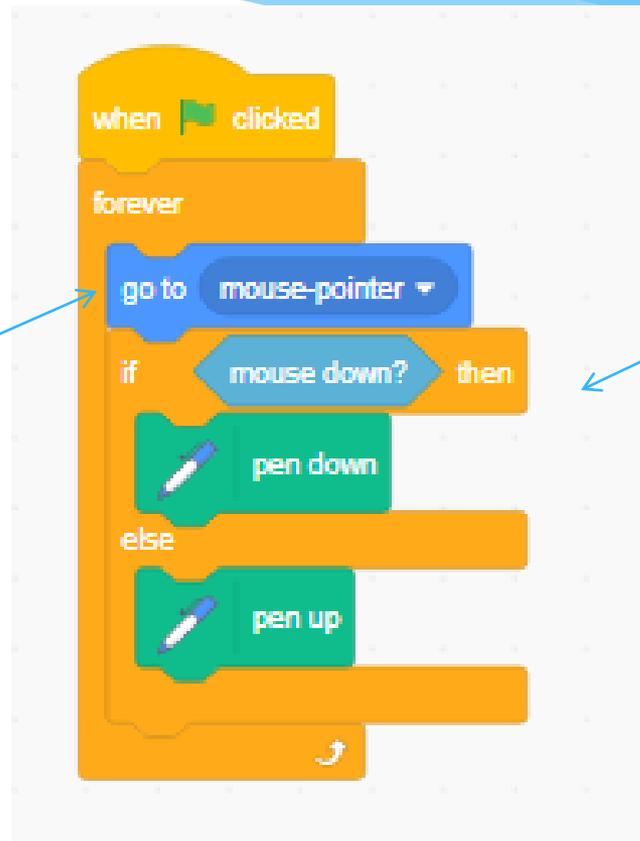
Notice, we have 'MOVED' the pencil sprite's costume a bit 'away' from the centre. This is not necessary, but creates an effect that the tip of the pencil is drawing.

Role of variables in the PAINT GAME

- * The 'PENCIL' sprite needs to remember how thick the pencil should write, what color it should write in and so on.
- * We can use 'variables' to set these.
- * Additionally we will use the PEN extension, as in the DRAW exercise (lesson 2), to draw.

Getting pencil sprite to draw

Always go to
MOUSE POINTER



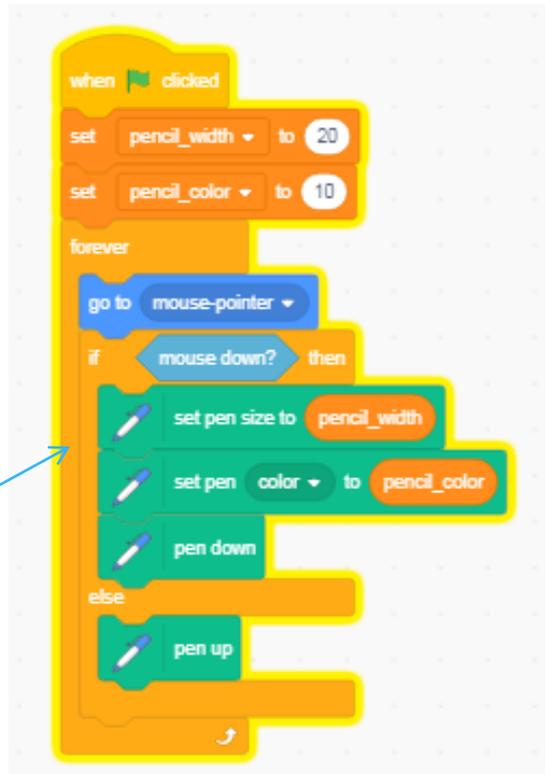
If MOUSE
CLICKED, DRAW

(Remember PEN
DOWN is like
putting the pen
to the paper).

Adjust Pencil width and colour

Create a variable called pencil_width and pencil_color.
In the display, change these variables to 'slider' readout.

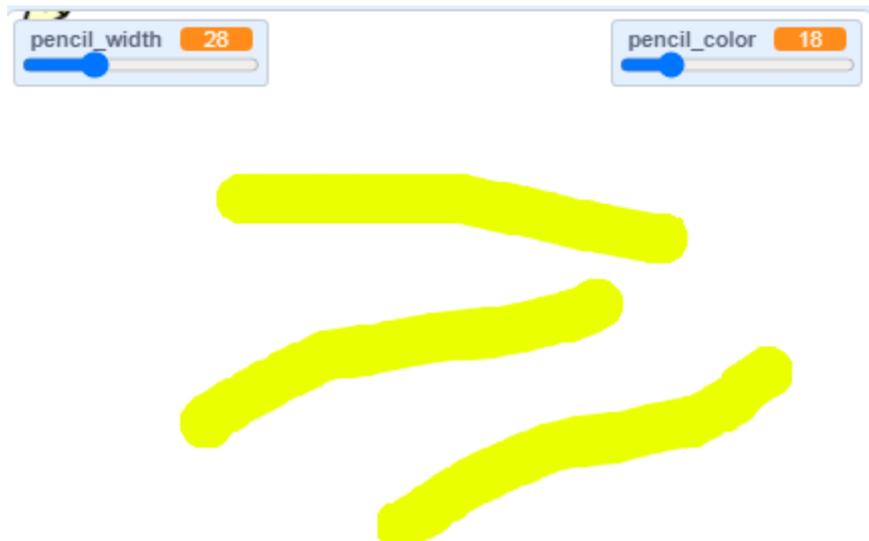
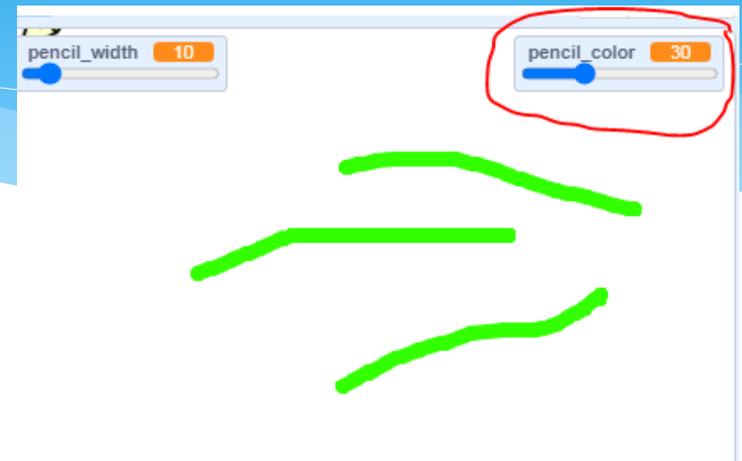
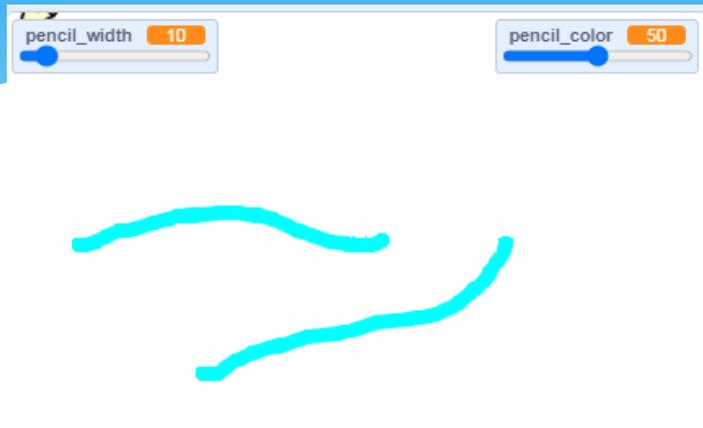
Set pen colour and size BEFORE putting the pen down



Notice, the PEN SIZE (in other words, the thickness of the PEN) will be controlled from the slider for 'pencil_width')

The PEN color (in other words, the colour of the PEN) will be controlled from the slider for 'pencil_colour')

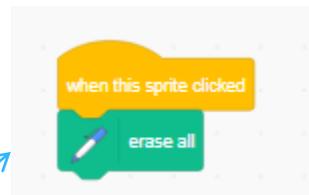
Now, see the effect of the Slider



Notice that by adjusting the slider the pen width and colour changes.

An erase button

- * An erase button which clears the stage.
- * Add a new sprite, give it this code.



Straightforward code



You are all set ...

- * Variables are the key currency of programming, learn them well. You will need them in whatever programming language you use.
- * For now, though, you are all set for your independent activity – 6.
- * Enjoy!

Extra Innings

Stamping

- * Pen Extension has a feature called 'STAMP'.
- * By using stamp, we can get sprite to make 'stamps' – like tattoos.

A variable called stamp_mode

- * We create a variable called the ‘stamp mode’.
- * Remember, this variable is ‘visible’ to all sprites.
- * In what follows, we will see how two sprites will access the same variable.

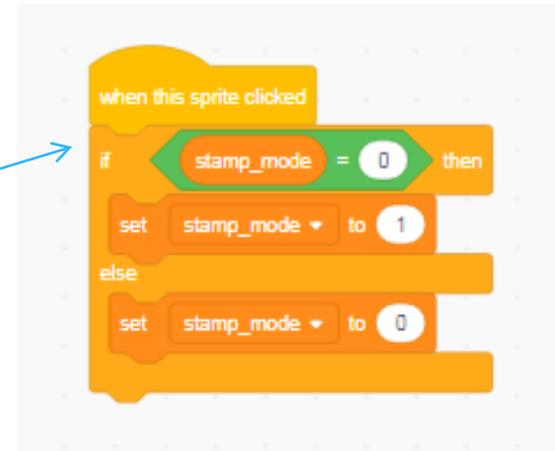
A button that turns on stamping

WARNING: A BIT ADVANCED

- * Now we add another Sprite called the 'STAMP MODE'. If this sprite is clicked, it 'toggles' the STAMP MODE.
- * That is, if the sprite was stamping, it returns to drawing.
- * If it was drawing, it starts to stamp.



Notice how the stamp_mode changes to 1 (if it was 0) or changes to 0 (if it was 1).

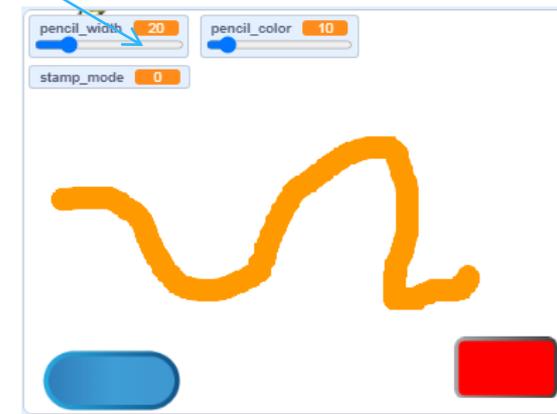
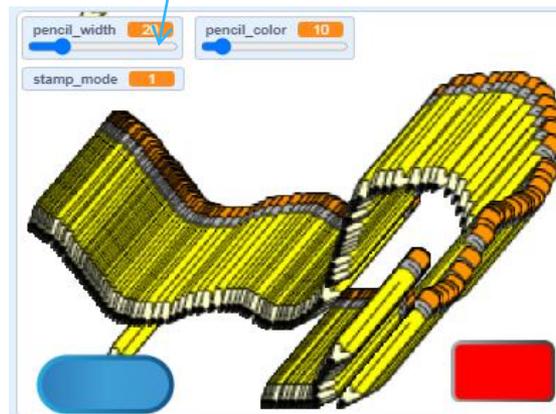


Modifying the DRAW code for STAMPING

WARNING: A BIT ADVANCED

```
when clicked
  erase all
  set pencil_width to 20
  set pencil_color to 10
  forever
    go to mouse-pointer
    if mouse down? then
      set pen size to pencil_width
      set pen color to pencil_color
      if stamp_mode = 1
        stamp
      else
        pen down
    else
      pen up
```

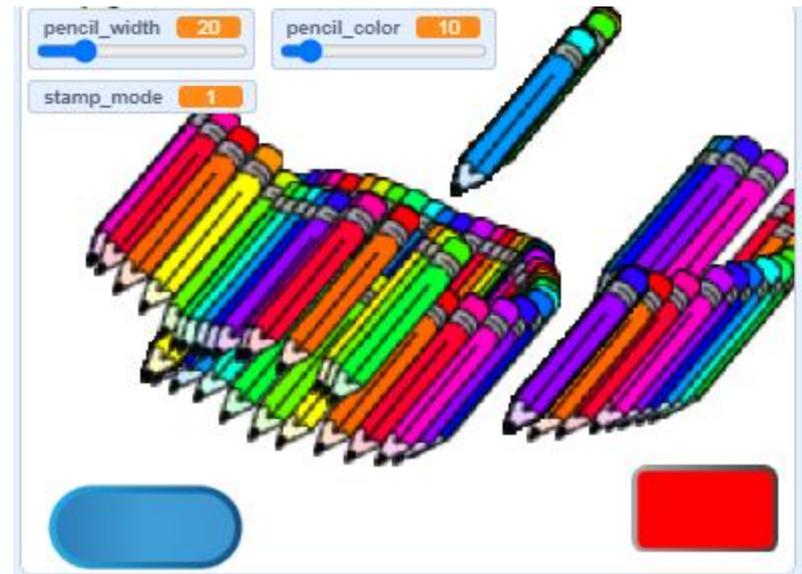
If Stamp_Mode is chosen, we change the drawing to 'STAMPING'



Color Effect

Use Colour effect with stamp to double the fun:

```
if stamp_mode = 1 then
  stamp
  change color effect by 25
else
  pen down
```



Play with this 😊