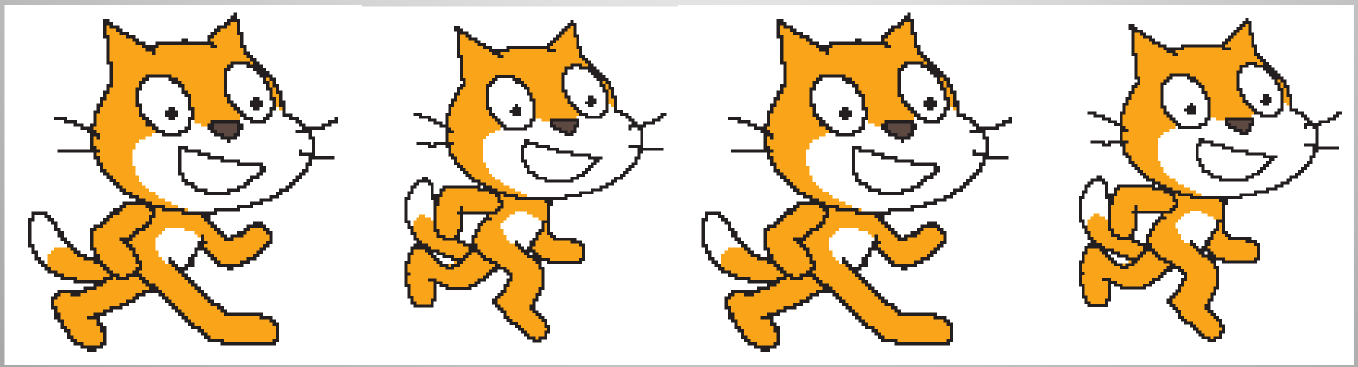
Introduction to Computer Programming - Project 2

Intro to Digital Technology

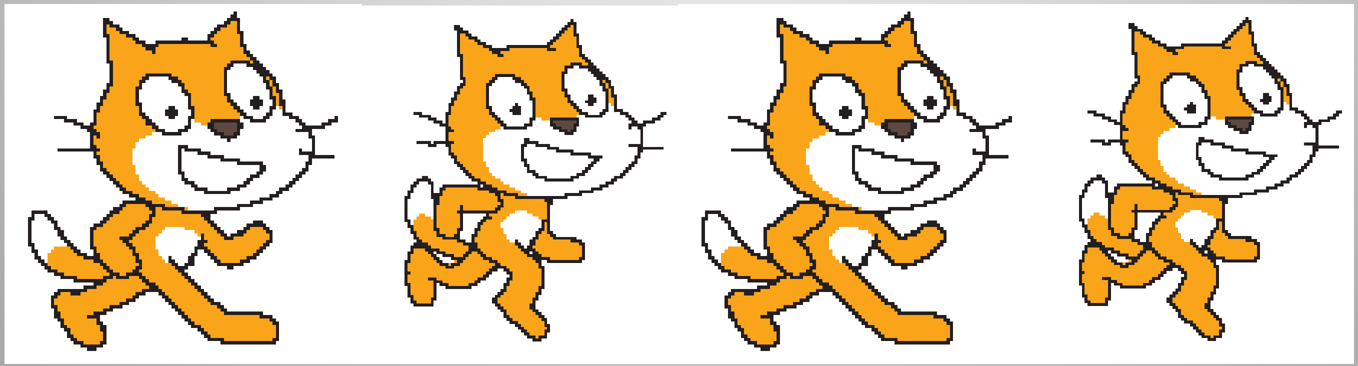
Costumes

A great feature of Scratch is that sprites can have multiple costumes. A sprite's costume is the image that shows when the sprite (object) is on the screen. Some have only one, but if it has more than one, you can also use programs to change the sprite's costume. Use the 'Costumes' tab to see a sprite's costumes.



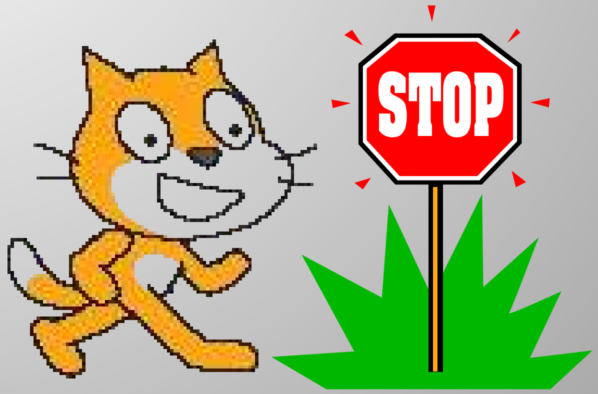
Costumes

If you check, you'll see that the cat sprite has 2 costumes, so let's change it. Get 4 'next costume' and 4 'say something for two seconds' blocks and lock them together alternating between each one (next costume, say, next costume, say, etc.) and run your program. You'll find out on the next slide why you had to also use the 'say' blocks.



Control

A program will run as fast as the computer can run it. Sometimes this is too fast, and you might need a delay between steps. Lock 4 more 'next costume' blocks in a separate program and run it. Do you see any changes? It's running so fast you don't see each change. That's why you put the 'say for 2 secs' blocks in there before – to make it pause between each change so you could see it. There's a better way…



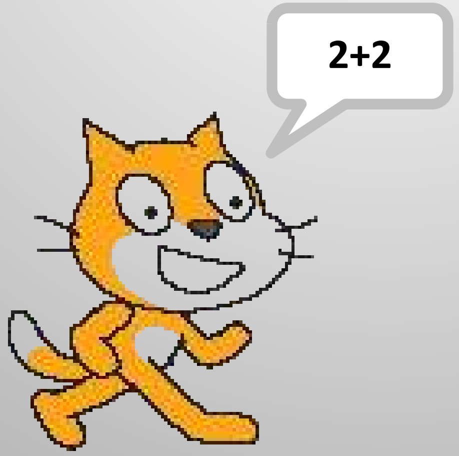
Control

To make your program pause, you need to use a block from another category – Control. These blocks control a lot of things about how and when your programs run. For now, we're only interested in the 'wait # secs' block, which makes the program stop for the specified time. Drag one of these into the program you just created in between each 'next costume' block. Run it and see what happens. Change the wait value and see the effect. You can use decimal values if you need fractions of seconds.

Strings & Numbers

Strings are combinations of letters, numbers, symbols, etc. They’re treated like characters (even the numbers), and the computer cannot do math with them. In Scratch, if you enter something into a rectangle input box, it will be treated as a string.

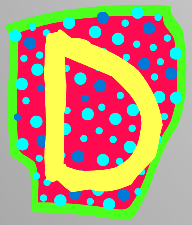
Make your sprite say 2+2 and see what it says.

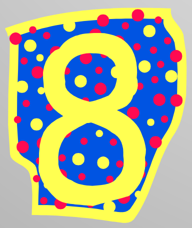


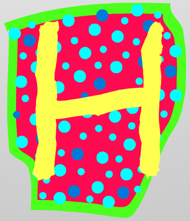
Strings & Numbers

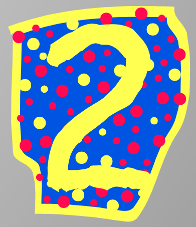
Numbers include only numbers and valid numeric symbols (+, -, etc.). The computer CAN do math with numbers. In Scratch, numbers are entered into rounded input boxes. In fact, Scratch prevents you from entering non-numeric values into these boxes.

Find a block with one of these rounded input boxes and try to enter a letter into it.





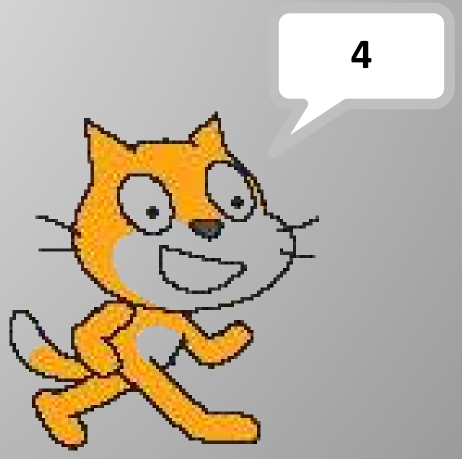




Calculations

To do mathematical calculations in Scratch, you have to use another category of blocks - Operators. These allow you to do basic math, compare values, and perform other types of calculations. The operator blocks can be dropped into any input box (rectangle or rounded) on any block, and Scratch will first calculate the result then use it in whatever action the block does.

Drop an add (+) operator into a ‘say…’ block, put a ‘2’ in each of the input boxes, then see what the cat says.



Calculations

You can ‘nest’ operators (put one inside the other) to create complex calculations. Programming languages follow the order of operations (PEMDAS) when doing math. Scratch does this by the way you nest your operators. As you nest them, it puts them on higher layers. The layers are computed top to bottom.

Start with an + operator and put a multiply (*) one in the second box. Put 2s in all boxes, put the whole thing in a ‘say...’ and see what you get. Reverse it and try again.

Random Numbers

One thing practically all programming languages can do is generate a random number, an unpredictable value from a specified range. This is useful for games, simulations, etc. In Scratch, this is done using an operator called ‘pick random x to y’, where x and y specify the range of numbers from which to choose (they default to 1 and 10).

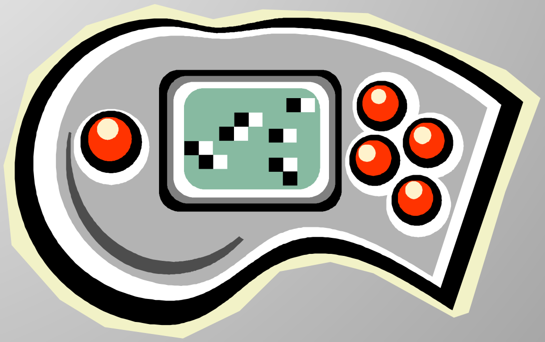
Drag a ‘say’ block into the scripts area, then drop a ‘pick random’ block into its input area. Run it several times and see what you get.

Variables

A computer has a memory and can remember stuff. To get a computer to remember stuff, you use *variables*. A variable is a name you specify for a location in the computer's memory where it can store a piece of information. Many languages are picky about what you can name a variable – no spaces or special characters, can't start with a number, etc. Luckily for us, Scratch isn't that picky about what you call one.

Variables

To use a variable in Scratch, you have to create it first in the Variables category. You can then use it to hold on to any information you want. For example, suppose you're writing a game and you want to a way to keep score. Click the 'make a variable' button and create a variable called 'score'. We want this variable to be visible by all sprites (a 'global' variable), so make sure 'For all sprites' is checked. Click OK.



Variables

Once you create a variable, it shows up in the Variables category. There are also blocks there to set the variable to a value and to change a variable's value. When you want to use the variable in any other way, you'll need to drag and drop it in the input area of other blocks.

Notice that it is also displayed on the stage. If you don't want it displayed, uncheck the checkbox next to the variable name in the Variables category.

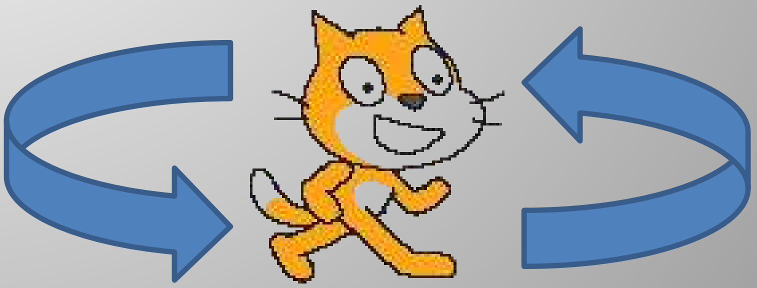
Variables

Clear out your scripts area, then drag a block to set ‘score’ to 1. Drag 3 ‘change’ blocks and attach them to the ‘set’ block. Make sure the change blocks are changing the score variable, and change the values to something between 1 and 100. Insert a ‘say’ block after each variable block to say the value of the score variable. You’ll have to drag the variable into the say block for it to work.

Y?

More Controls

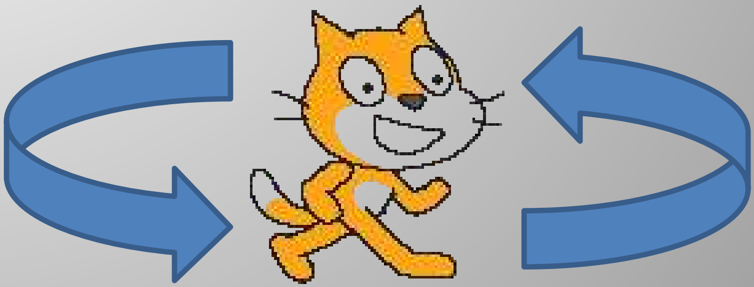
What if you wanted a group of blocks to run more than once? Maybe run forever? In a program, running a set of commands multiple times is called a loop. Some of the control blocks can do just that. Look in the Control category and find the 'forever' and 'repeat 10' blocks. Notice that they're shaped sorta like a 'C'. This allows you to put other blocks inside the 'C', and it will expand to hold them as you insert the blocks.



More Controls

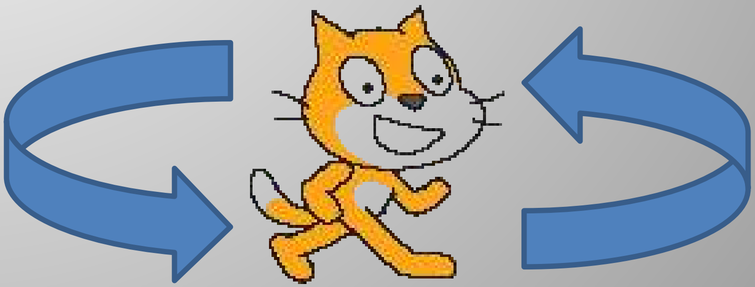
The ‘forever’ block will run the blocks inside it over and over until you click on it again to stop. This is known as an endless loop. In most cases, it’s not a good thing. Notice there’s no tab at the bottom of it. You can’t attach blocks below it, because once you’re in the loop, you can’t get out.

The ‘repeat 10’ block will run them the number of times specified in the input box – it defaults to 10, but you can change it.



More Controls

Clear your scripts area and drag a block to set the score to zero. Attach a ‘repeat 10’ block below it (not around it) and change the repeat value to any number 50-200. Attach a block <u>inside</u> the repeat block to change the score by 1. Attach a block to the bottom of the repeat block to say the score. Run it and see what happens.



Programming Project 2

Now it’s your turn once again:

1. Prior to creating the script, create 2 variables, one to store a delay value, one for ‘loop control’. Then create a script to do the remaining steps.
2. Set the loop control to a random number between 5 and 20.
3. Position the sprite at X:-125, Y:0.
4. Add a repeat loop block and insert the loop control variable into the loop count.
5. Put the following in the repeat loop:
    1. Say a message of your choice (just use the plain ‘say’ block).
    2. Switch to the next costume.
    3. Set the delay value to a random number between 1 and 5.
    4. Divide the delay value by 10 (hint – set it to itself divided by 10).
    5. Insert a wait block and use the delay variable for the value.
    6. Move 10 steps.

Save this in your Intro to Programming folder using this file name: Project 2