Introducing Scratch

1 – Get to know Scratch

Scratch is a fun and easy way to get started with the important principles of computer programming. Creating projects in Scratch allows users to become familiar with common programming concepts such as variables and control structures. Unlike most programming languages though, with scratch there is no need to learn any complicated commands as it is all visual. Programs are created by dragging blocks into a code area in a logical and simple way.

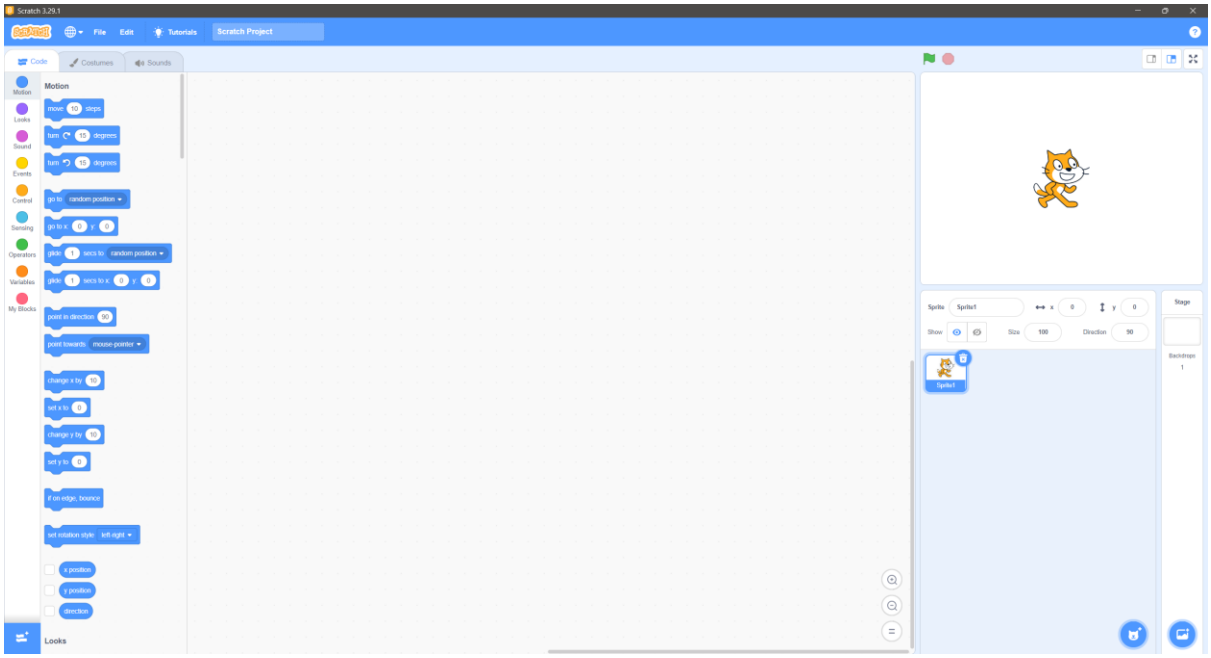
Not only is Scratch a great way to get started with basic programming, it's also free!

You can use Scratch through your web browser at https://scratch.mit.edu/. From there you can create and account and use the online Scratch editor. You can also download the desktop version of Scratch for Windows, Linux and Mac from http://scratch.mit.edu/download. The images shown in these exercises are from the desktop Windows version.

In the following exercises we'll be using Scratch to create a few simple programs. Let's start by getting familiar with Scratch.

Exercise 1. Setting up the Stage

When you open Scratch, there are three main sections to the screen.



On the right we have the **Stage** where all the action takes place, with your **Sprite List** below that.

On the left we have the **Blocks Palette**. This is where you find instructions for making things happen in your program.

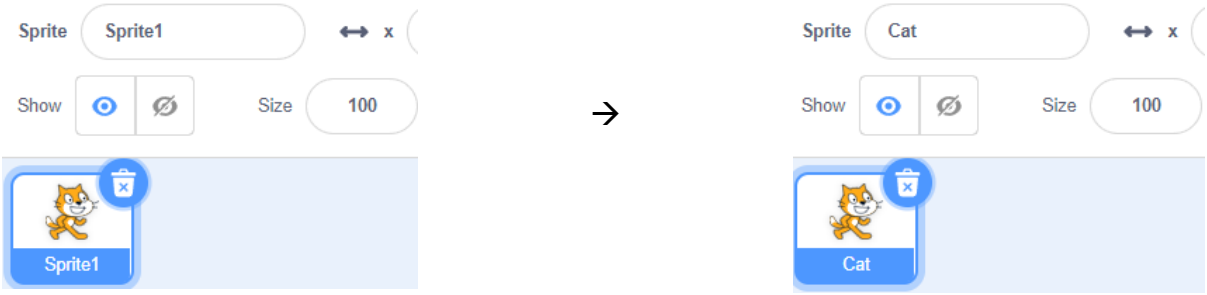
In the middle is the **Code Area**. This is where you piece programming instructions together.

Exercise 2. Working with Sprites

Characters and objects on your stage are known as **Sprites**. You can import sprites or draw your own sprites for use in your programs. Each time you start a new Scratch project you will begin with the Cat sprite. Let's use the Cat sprite to learn more about what Scratch can do.

Notice that the sprite has a name under it. When you have a lot of sprites in a program as you often do in Scratch, giving each one a good name makes it easier to program them. Instead of leaving the sprite with the name *Sprite1*, we'll give it a more suitable name.

Above the sprite area is a section with information about the sprite. This is where you can change the name of a sprite.

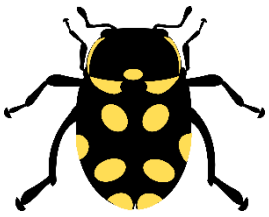


1. Change the sprite name to **Cat**.

Now let's try giving the cat some commands.

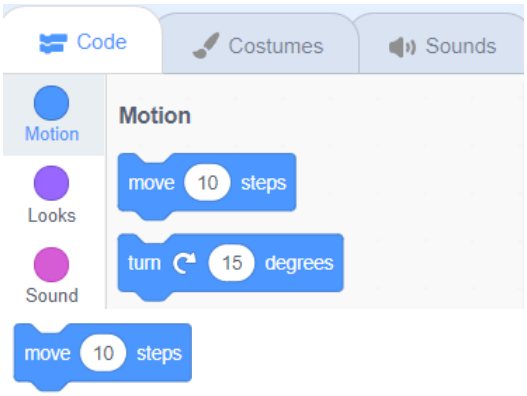
Logic Errors

A computer will do what you tell it to do, even if you accidentally tell it to do the wrong thing. Just like a person who is following your instructions might end up with the wrong result if you give them instructions that have a mistake. For example, if you are giving directions to someone and you tell them to turn right at a place where they should have turned left, then they're not going to get to where they are supposed to go. Or if you are meant to add 2 numbers together, and instead of a + sign, you accidentally put in a - sign, then you're not going to get the answer you were expecting. Giving instructions that don't make sense and won't give the results you're expecting is a **logic error**. It's easy to make these kinds of errors in computer programming, including programming with Scratch.



Tip	Problems (such as logic errors) that are stopping a computer program from working properly are often referred to as **bugs**. Finding and fixing these problems is known as **debugging**.

Exercise 3. Trying Out Code Blocks



On the side of the screen is the area where you get the code blocks to build your program. They are organised into categories, and each category has a colour to make it easier to find blocks, as well as see which category a code block has come from.

1. Make sure the **Motion** category with the dark blue code blocks is selected.
2. The first code block is the move 10 steps block. Click on it to see what it does to the Cat sprite. Each time you click it, the cat should move forward by a distance of 10 **pixels**. Pixels are the dots your screen is made up of and are used to measure a lot of distances in Scratch.
3. Blocks with a space in the middle like this one mean that you can change what is in the middle part. Click on the **10** inside the block and change it to negative ten (**-10**). Now when you click on the cat it will move backward.
4. Try it with different amounts.

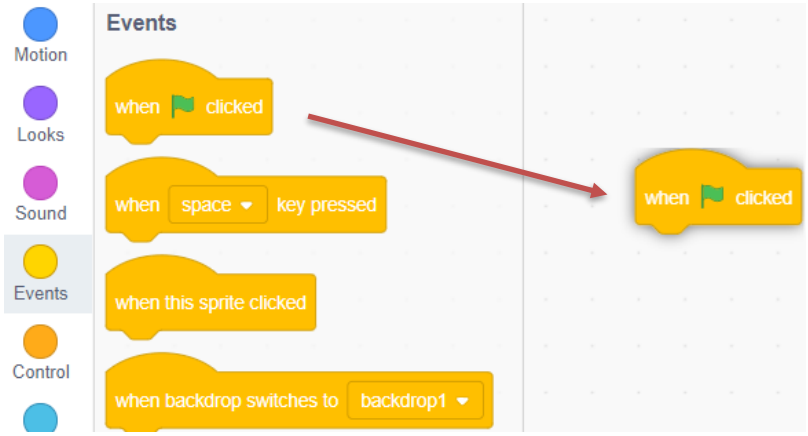
Tip You can test any block in Scratch by clicking on it. This even works on groups of blocks as we will see later.

Exercise 4. Using Code Blocks

The large empty space in the middle of the screen is where we place code blocks that we want to use in our program. We'll try some code blocks that will make the cat move.

1. Select the **Events** (yellow) category in the code blocks and find the **when**  **clicked** block.

2. Drag this block in to the code area.



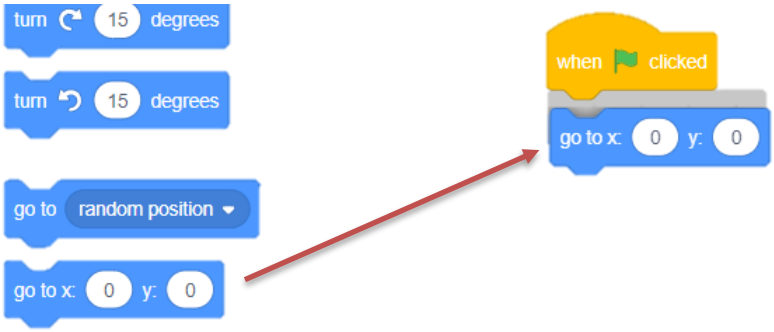
Note It doesn't matter where you place it in the code area. When you have a lot of blocks in the code area though, it is helpful to arrange them neatly, so it is easier to work on your program.

Above the stage there is a **go** button and a **stop** button. . Clicking the go button will start a program and clicking the stop button will end a program. The code block we have just placed allows us to tell Scratch what things should happen as soon as the program has started (when someone clicks on the button).

3. Select the dark blue **Motion** category and find the block. Change both of the numbers to **0** of they aren't already.

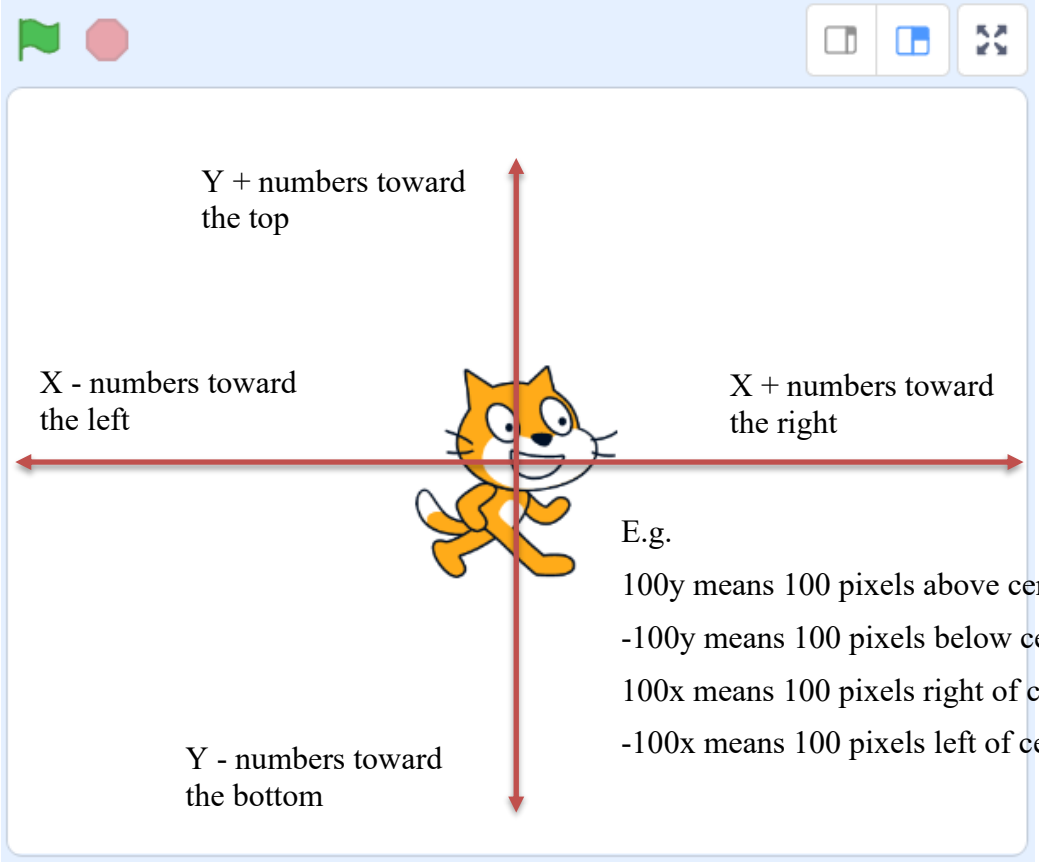
Notice that many of the blocks have a small tab on the bottom of them. These blocks can be attached to other blocks a bit like jigsaw puzzle pieces or Lego pieces.

4. Drag the **go to** block so that it attaches to the bottom of the block we already placed in the code area.



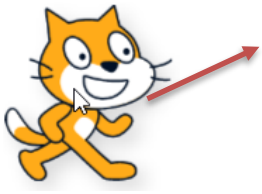
This will tell scratch that the second block will run right after the first block. In this case that means that when the go button is clicked, then this sprite will move to a certain position.

Coordinates



Coordinates are used to refer to positions on the stage, a bit like coordinates on a map. The first number or the X coordinate refers to how far to the left of the stage or the right of the stage. 0 is in the middle, positive numbers are to the right and negative numbers are to the left. The second number or the Y coordinate, refers to how far up or down on the stage. 0 is in the middle, positive numbers are towards the top and negative numbers are towards the bottom.

With that last block we added the X coordinate (sideways) and the Y coordinate (up and down) are both 0 so that means that as soon as the Go button is clicked to start the program, the cat will be positioned right in the centre of the stage area.



5. Drag the cat to a different part of the stage.
6. Click the button above the stage. Clicking the button and starting the program will move the cat to the centre of the stage.
7. Change the numbers and click the button again. Where does the cat go to with the new buttons?
8. Try the following:
   - Put negative (-) number in both the x and y coordinates and click the button to see where the cat moves to when you use negative numbers.
   - See what the highest number is that you can put into each coordinate (and the lowest number with a – sign). This will give you an idea of how big the stage area is.
9. Change both the x and y numbers back to 0 and click the button again.

Exercise 5. Responding to Keyboard Buttons

Now we will program the cat to move when the arrow buttons on the keyboard are pressed.

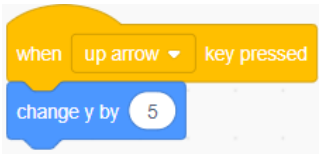
1. Go to the yellow Events category.
2. Find the **when space key pressed** code block.

3. Drag four of them into the code area.

4. Change one of them so that instead of saying space, it says up arrow.

5. Change the others so that there is one for each arrow button.

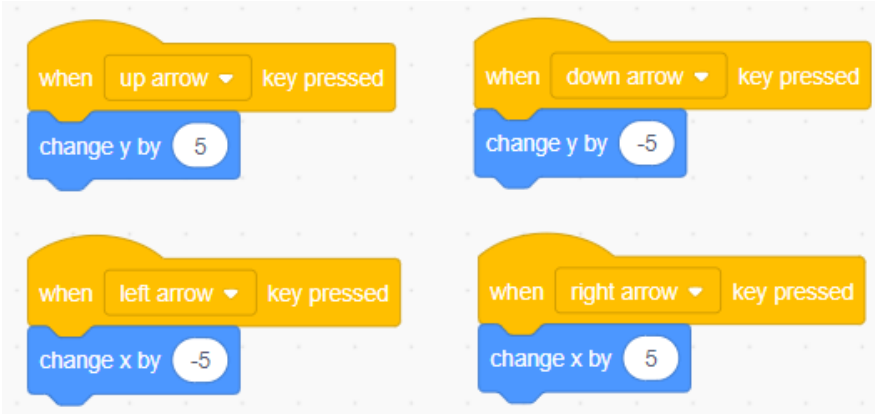
Each one of these will tell Scratch to do something when a button is pressed. We'll add a code block to each one that will move the cat in a certain direction when the key is pressed.



6. From the **Motion** category, find the **change y by** block.
7. Add it to the **when up arrow key pressed** block and change the number to **5**.
8. Test it by pressing the ↑ key on your keyboard.

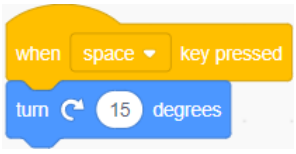
Remember that **Y** coordinates are for up and down position. Positive numbers are toward the top. This block adds 5 to the Y position which moves the cat upward each time the ↑ key is pressed.

9. Add a **change y by** block on to the when **down arrow key pressed** block. Change the number to **-5**. Since it is a negative number, it will take away from the Y position, moving the cat downward.
10. Add change coordinate blocks to the other key press blocks as shown below.



11. Test the code blocks by trying each of the arrow keys on your keyboard to move the cat around.
12. Click the  button above the stage to restart the program and move the cat back to the middle of the stage.

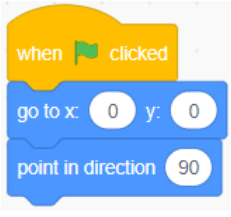
We'll add one more key command. This time, we'll add one to make the cat turn when the spacebar is pressed.



13. Add 2 new code blocks like the following.
14. Press the **spacebar** to make the cat turn.

Tip Try holding down the spacebar or arrow keys to make the movements continuous.

15. Add a new block to the starting code from the **Movement** section so that when the program is restarted, the cat will face to the right as well as move to the middle of the stage.

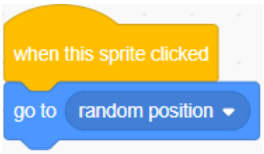


Exercise 6. Responding to the Mouse

The last thing we'll add to this program is some code blocks that will make the cat move to a random position on the stage when you click on him. Hey – maybe he doesn't like being clicked!

1. From the **Events** category add a **when this sprite clicked** block.

From the **Movement** category, add a **go to random position** block so that it looks like the example shown.

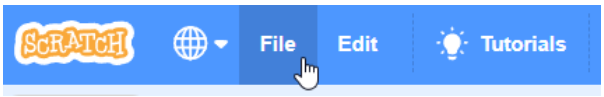


2. Click the cat to test the newly added code blocks.

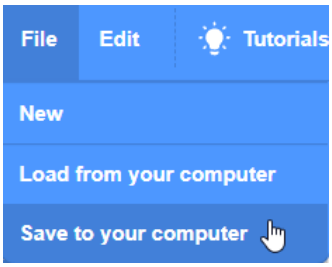
Exercise 7. Saving the Program

Saving your programs regularly in scratch is just as important as it is in other programs.

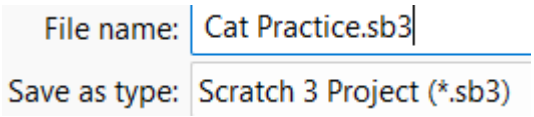
1. Click the **File** menu at the top of the screen.



2. Select **Save to your computer**.



3. Save the project in a suitable location with a suitable filename (such as Cat Practice).



Finished Code

The completed code for your project should look like the following. Remember that your code blocks can be positioned wherever you like in the code area but arranging them neatly can make it a lot easier to modify your code blocks and see what they are doing.