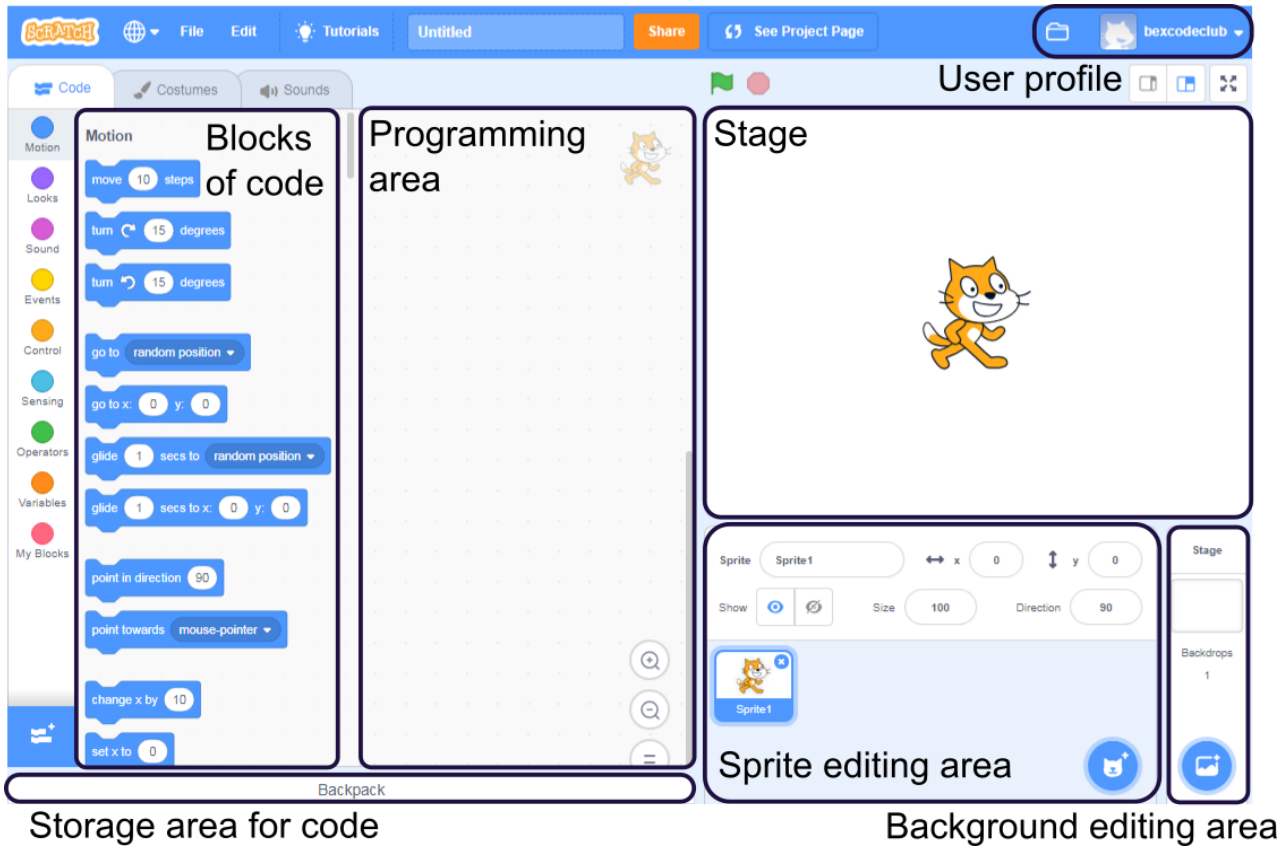
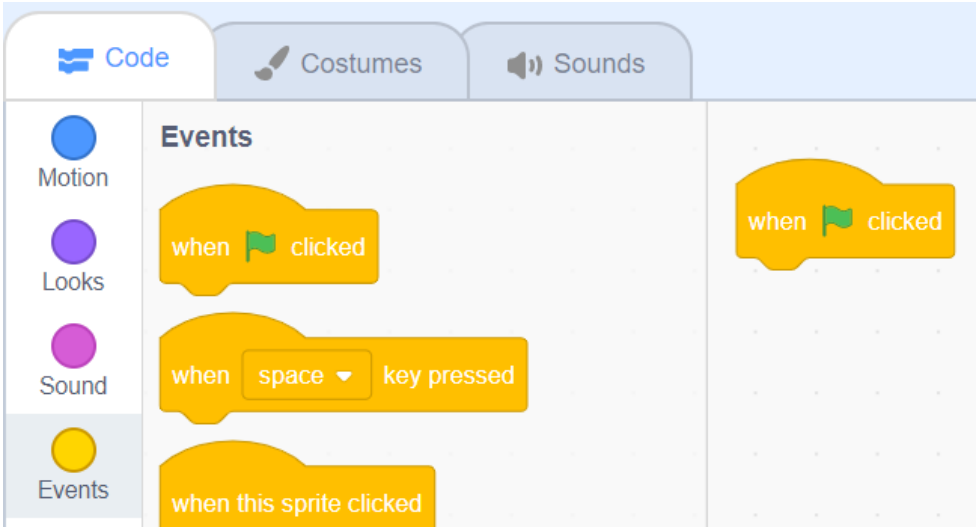
Drawing Shapes in Scratch

SESSION ONE – GETTING STARTED

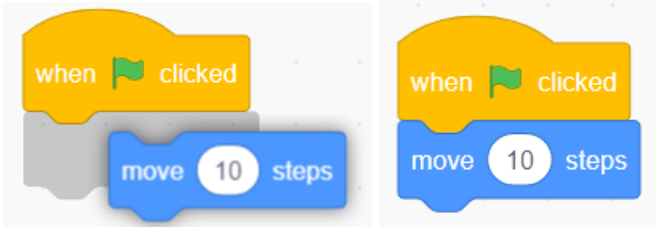
Join or log in to Scratch and click create to start a new project.



We need to use a block to signify when our program will start. These can be found in **Events**. From **Events**, find the 'when green flag clicked' block. Drag and drop the block into the programming area.



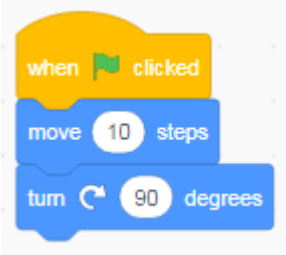
The next step is to find a block to make our sprite move. In **Motion**, find the 'move __ steps' block. Drag it into the programming area and under the 'when flag clicked' block until a grey shadow appears. Releasing the block will snap it to the 'when flag clicked' block.



Congratulations! You have now written your first program! Click the green flag above the stage area to test your program.

> Try changing the number in the motion block. What is the smallest number that you can still see the sprite move? What happens if you use a negative number?

To make our square, we next need the sprite to turn 90 degrees. Also from **Motion**, find the 'turn __ degrees' block. Snap it under your 'move __ steps' block, and change the number to 90. Now what happens when you click the green flag?



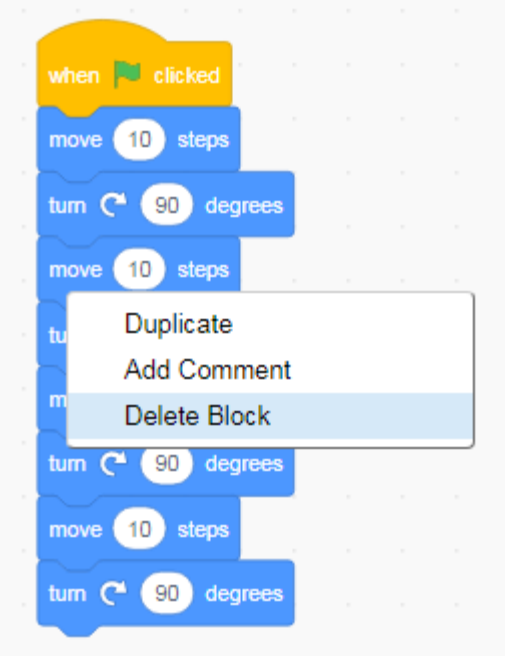
Continue adding blocks until you have enough instructions to make a square. What happens now when you click the green flag?

> Not working as expected? When our code doesn't seem to work, we need to identify the problem. This is called debugging. At what point does the code seem to stop working?
>
> Computer programs are often much faster than humans. If it looks like your sprite stops moving when it's programmed to start and stop in the same place, it is actually moving faster than your brain can see!

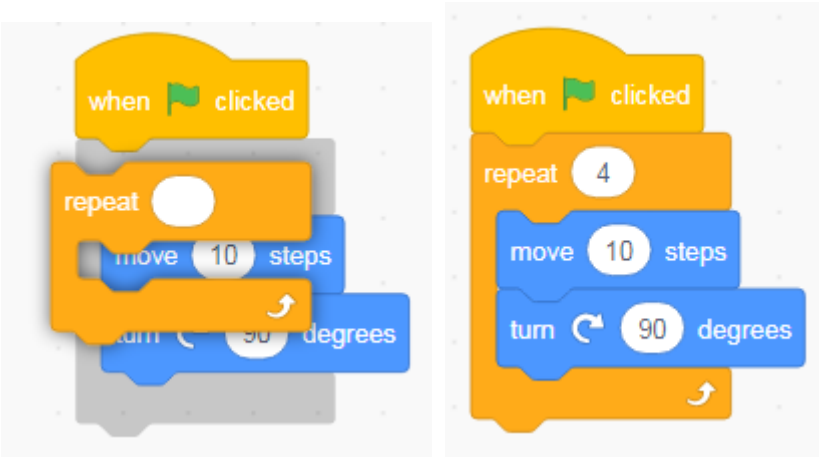
Using Repeats

Using four sets of move and turn blocks is fine when we are only making a square, but what if we wanted to make a shape with many more sides? It would quickly get messy and difficult to keep track of the number of blocks. Programmers have a way of simplifying this by using a repeat.

First, remove the bottom three pairs of block by dragging them back into the code area, or right clicking and selecting delete.

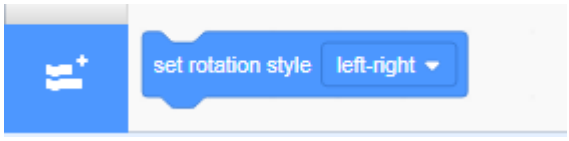


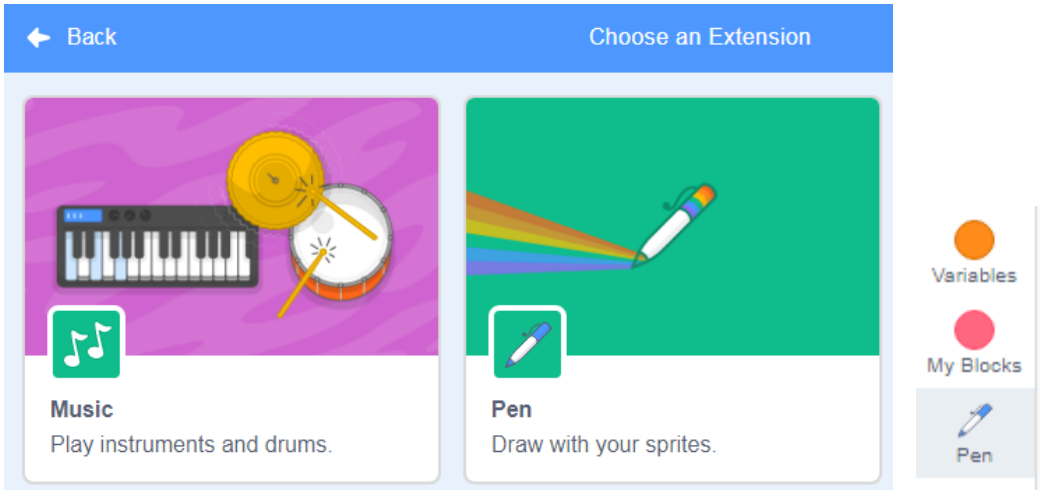
From **Control**, find the 'repeat __' block. Drag it into the programming area and release when the grey shadow surrounds the blocks of code you wish to repeat. Change the number of repeats to four.



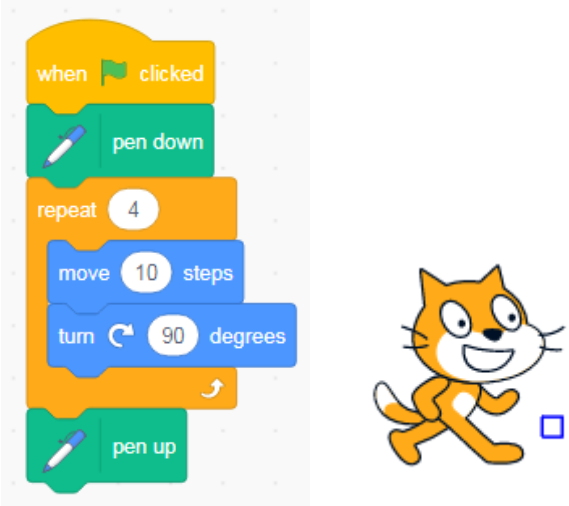
Drawing Shapes

Now that we can make our sprite move in a square, the next step is to get it to draw the shape in the stage area. Click the extensions button at the bottom of the code area and choose the pen extension. This will give us a new set of blocks

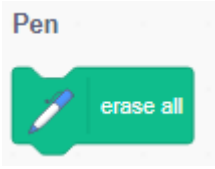




Add a ‘pen down’ and a ‘pen up’ block to your code and click the green flag. Have you drawn a square? You may need to move your sprite by clicking and dragging it to a different location on the stage to see it!



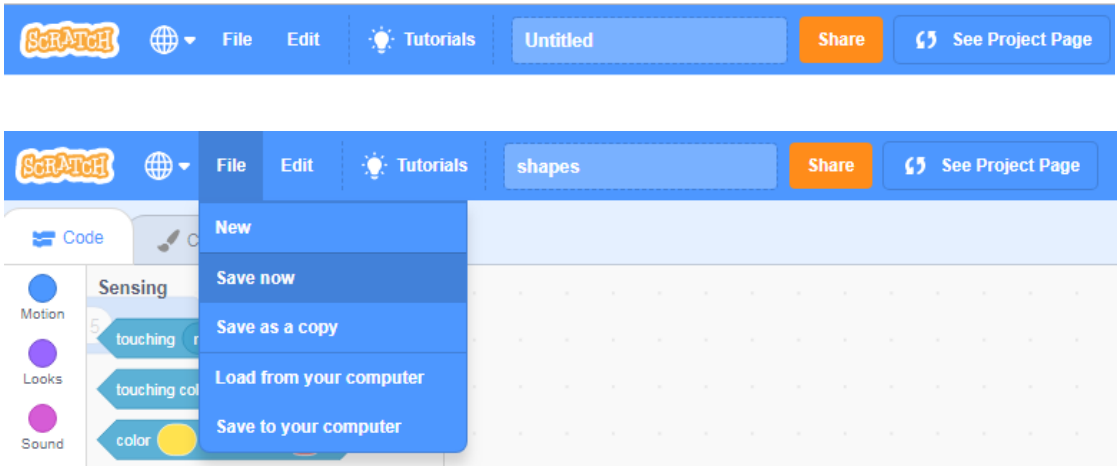
To clear the stage of your drawings, either double click the ‘erase all’ block in the code area, or add the block to your code.



Extra: Can you make your square bigger? Could you draw lots of differently sized squares inside each other?

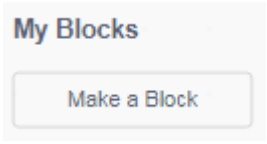
Saving Your Work

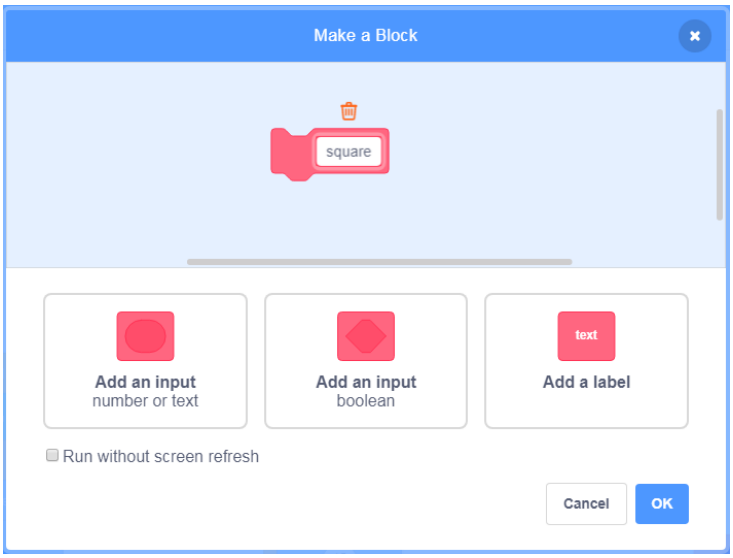
It is important that you save your work at regular intervals. Change the name of the file in the bar above the stage. Click file, ‘Save now’ to save the project.



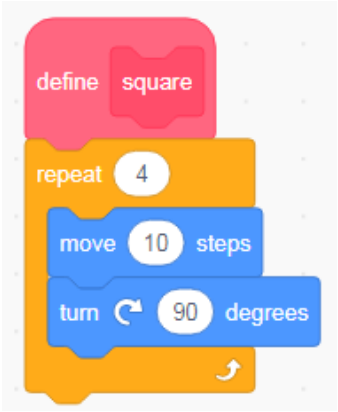
Extension – Using My Blocks

If we are likely to be repeating a set of commands or blocks multiple times in a program, we can create a new block to save ourselves time and to keep our work neat. Head into **My Blocks** and click make a block. Give your block a sensible name, like 'square'.



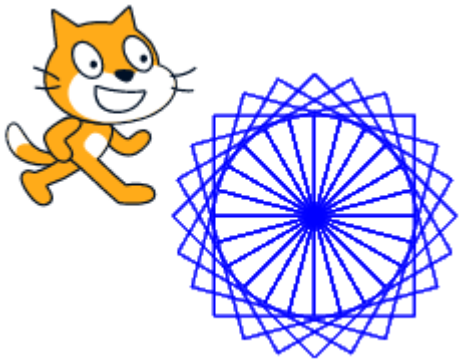


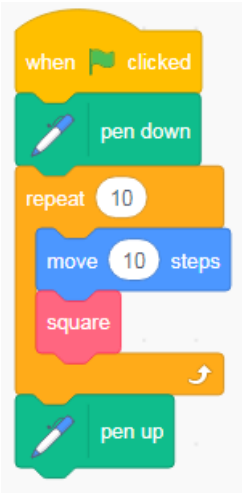
A 'define square' block will appear in your programming area. Add instructions to create a square to this block. Now, whenever you need to draw a square, you can use your custom 'square' block!

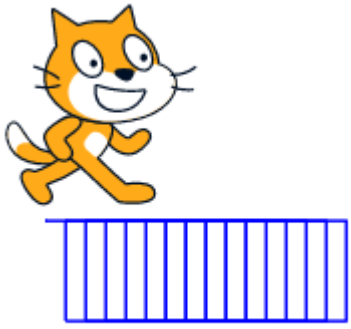


Extension – Creating Patterns

We can use Scratch to create different patterns. Try using your custom block along with extra turns, moves, and repeats to create new patterns.

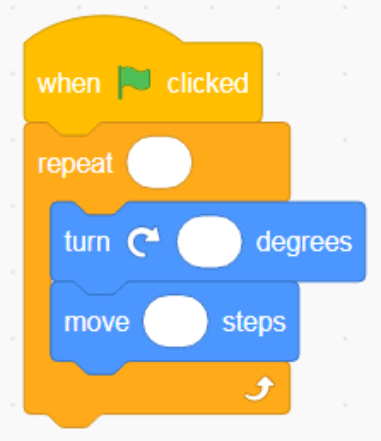






SESSION TWO – REGULAR POLYGONS

We can use Scratch to draw regular polygons by changing adjusting the angle that the sprite turns and how many times it turns. Open Scratch and create a new project. From **Events**, find the 'when green flag clicked' block and drag it into the programming area. Add a 'repeat' block from **Control**. Add a 'turn __ degrees' block and a 'take __ steps' block inside your 'repeat' block.



Edit the numbers in your block to create different shapes. Can you make a triangle? How about a hexagon?

> Not working as expected? The sprite will turn the number of degrees in the external angle rather than the internal angle. To calculate the external angle, take the internal angle away from 180 degrees. E.g. for a triangle, 180 - 60 = 120.

To draw the shape, add the pen extension, then add a 'pen down' and a 'pen up' block to your code.

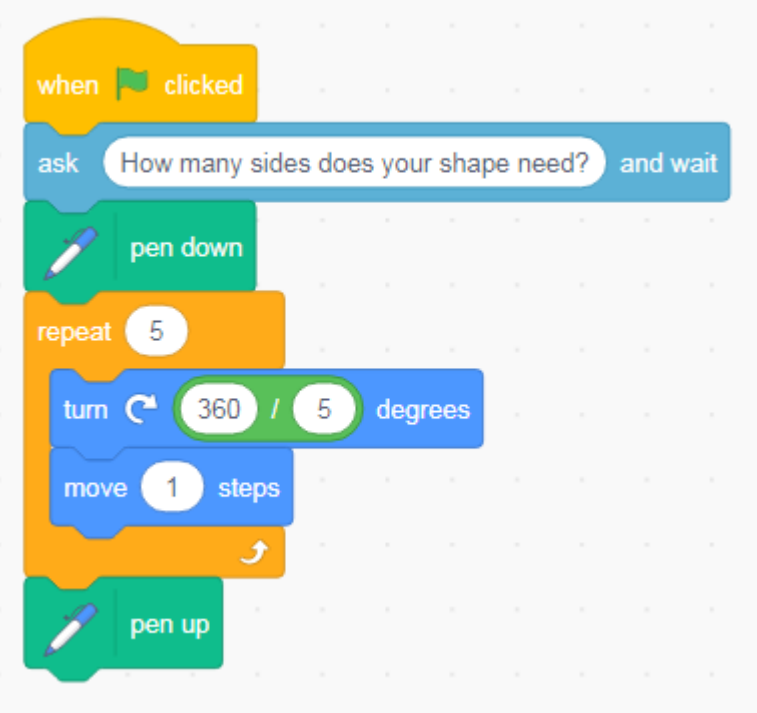
Using Operators

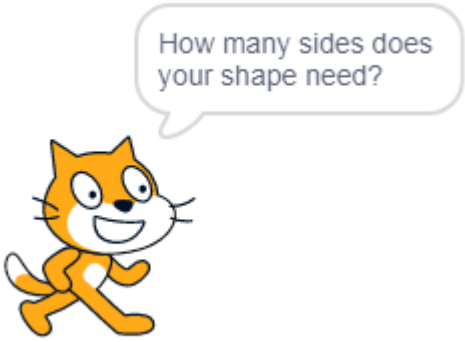
Rather than having to work out the angle we need to turn, we can let the computer do the hard work for us. From **Operators**, find the 'divide' block and add it to the white space in the 'turn' block. To calculate the angle needed, type 360 in the first white space and the number of sides of your shape in the second white space.

Taking User Inputs

As well as creating code to draw specific shapes, we can also write generic code that requires user input to specify the number of sides in the shape. To get user input, we will use the 'ask' block from **Sensing**.

We can type in a question that we want our Sprite to ask us.



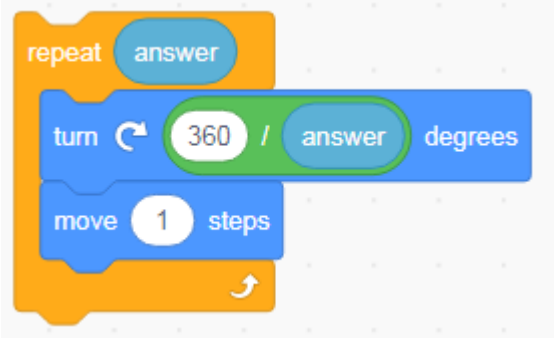


When we click on the green flag, a text box will appear in the stage area that allows a user to type their answer. The answer is then stored as a variable, which we can use later in our code.

From **Sensing**, find the 'answer' block. Add two of these blocks to your code in the positions where we need to know the number of sides.

> A variable is a way for the computer to store small chunks of information like a name or a number. In Scratch, the default variable type is a number.

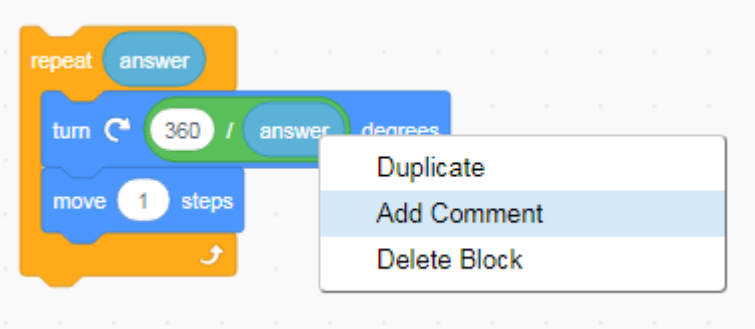
> Did everything work as expected? Can you think of any potential problems with asking for a number? What would happen if the user typed a word rather than a number?

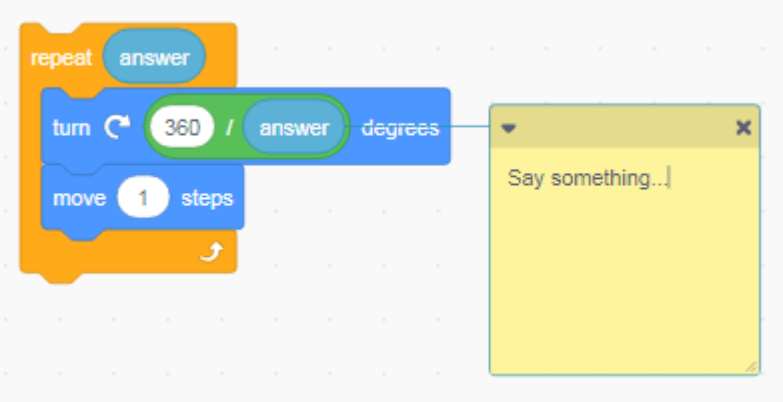


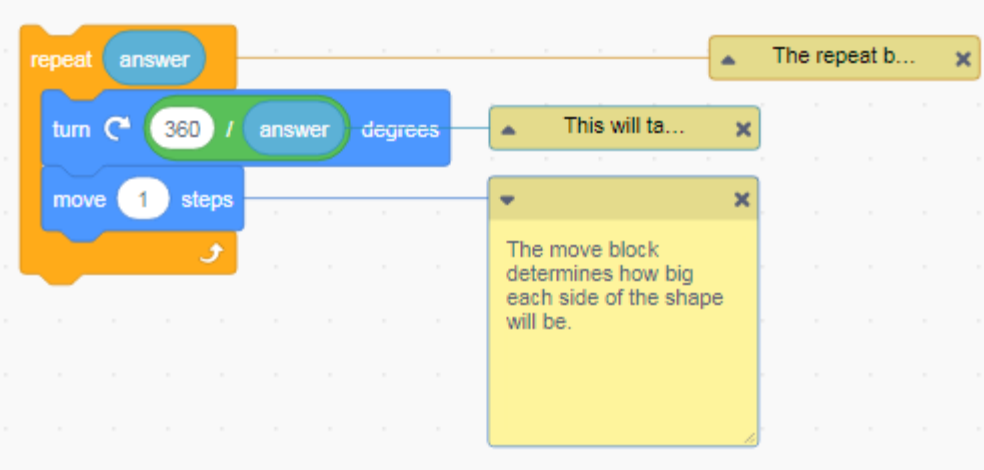
Extension – Commenting on your code

We can add comments to our blocks of code to demonstrate our learning and to explain our work to others. Right click on the block or blocks to which you want to add a comment. Use the small arrow to minimise the comment to keep your workspace neat.

Tip: Asking students to add comments to their code makes it easier to check their understanding and mark their work

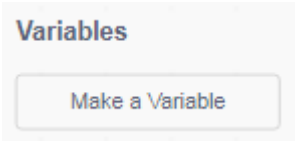


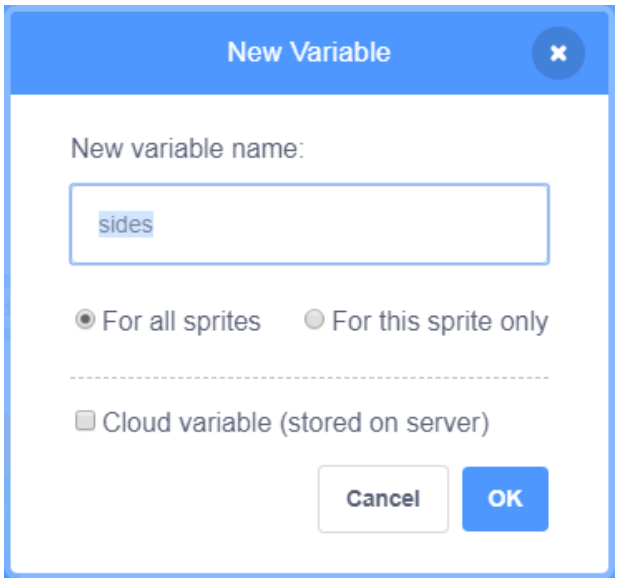




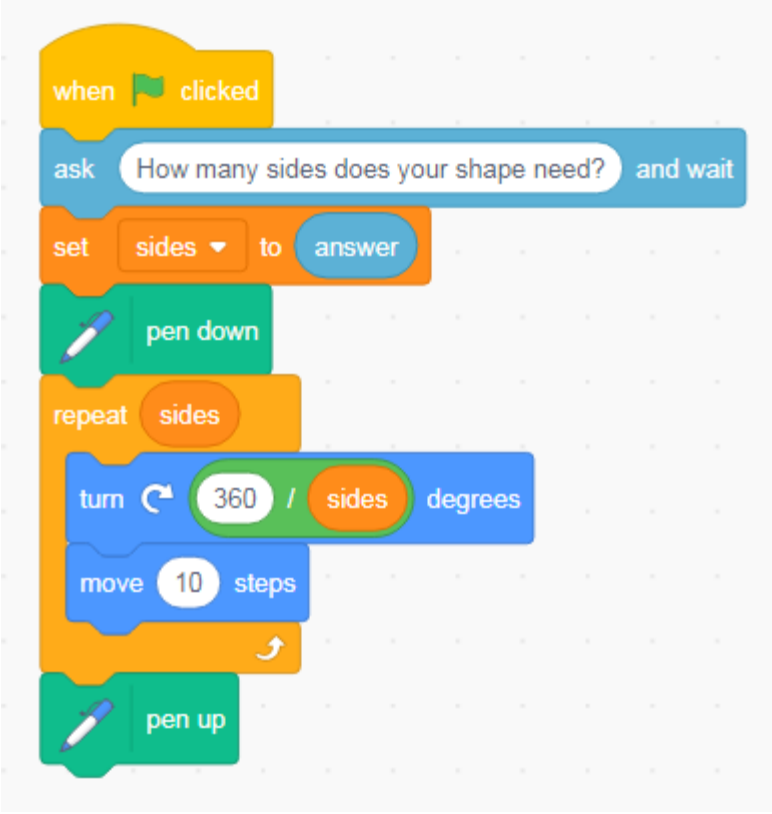
Extension – Creating Patterns with User Input

In Scratch, we can ask a user more than one question, but it can only store one answer at a time. To get around this, we can store each new answer as a variable. Head to **Variables** and click 'Make a variable'. Give your new variable a sensible and recogniseable name, like 'sides'.

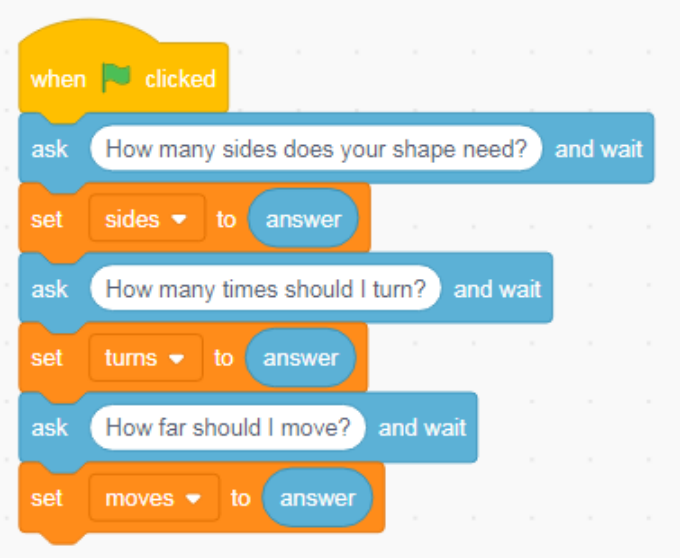


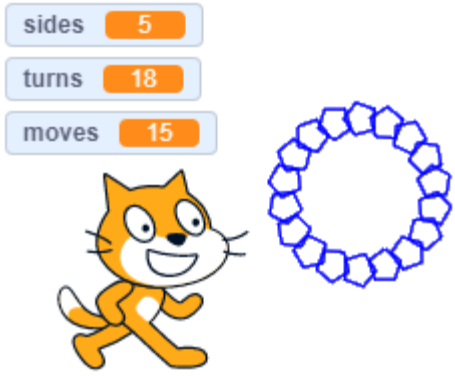


We can now set our variable, sides, as the answer to our first question and replace the 'answer' blocks in our code with the new 'sides' block.



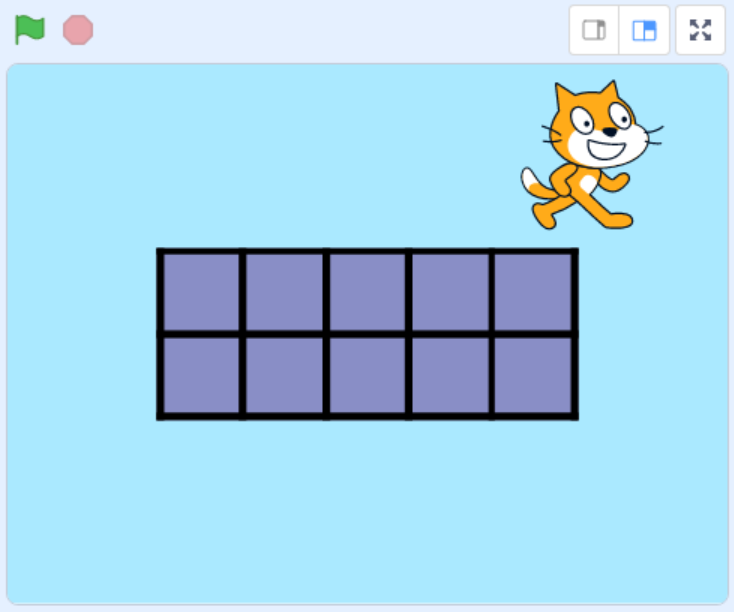
We can now add additional questions and use the answers to customise pattern creation with additional moves, turns, and repeats.



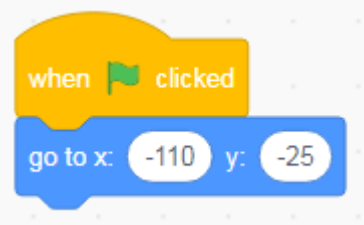


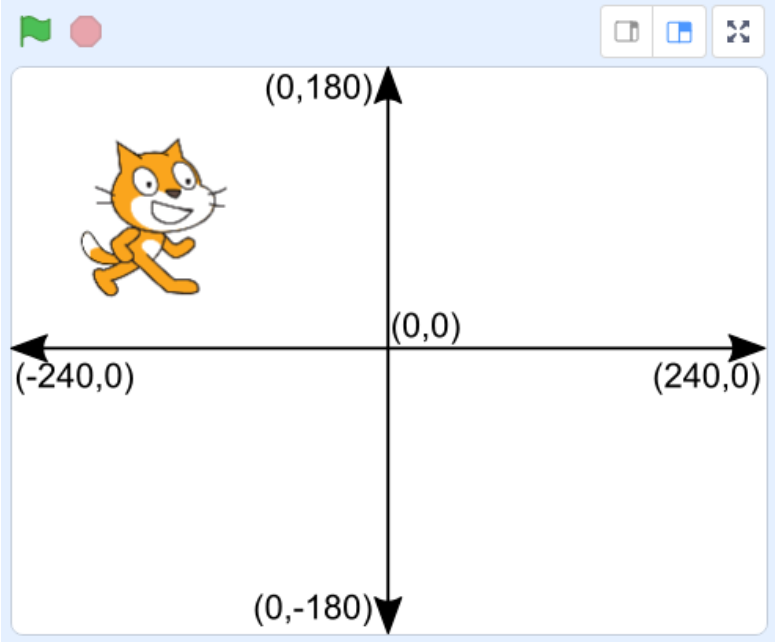
SESSION THREE – MAKING A TENS FRAME

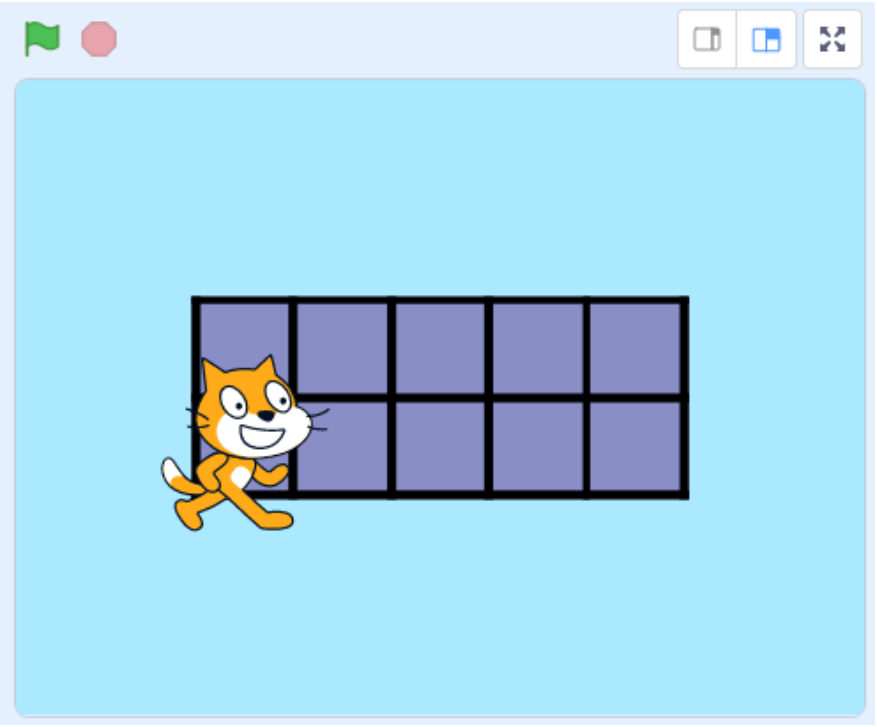
Open this project in Scratch. As you can see, we already have a tens frame drawn in the stage area. What we need to do is to write a program to fill the tens frame.



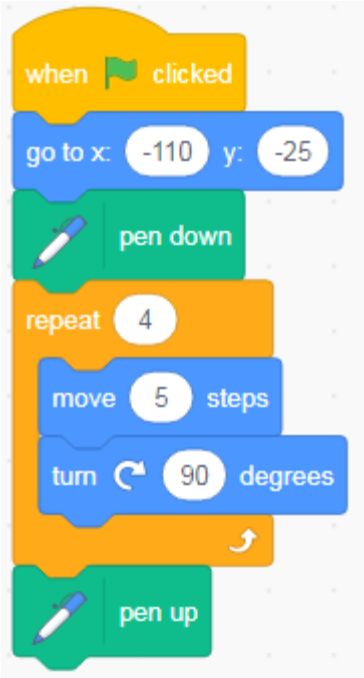
The first thing we want to do is to move your sprite into the first square of the tens frame. The stage area in Scratch is set up in a grid with x coordinates ranging from -240 to 240, and y coordinates ranging from -180 to 180. Use a 'when flag clicked' block and a 'go to x:__ y: __' block from **Motion** to position the sprite at grid point x:-110, y:-25.

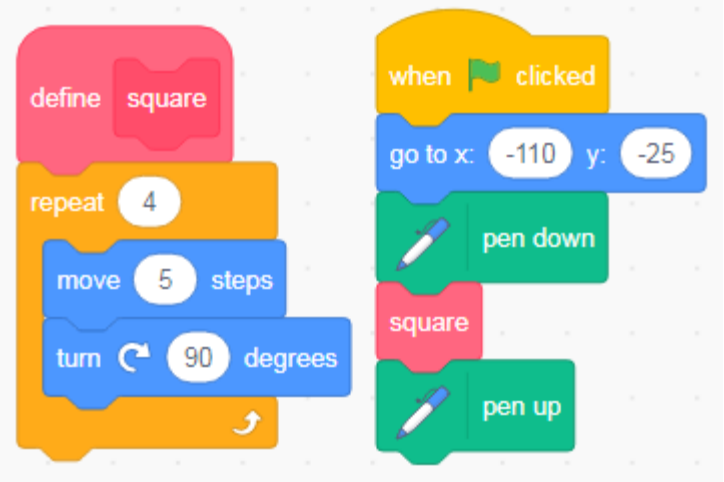




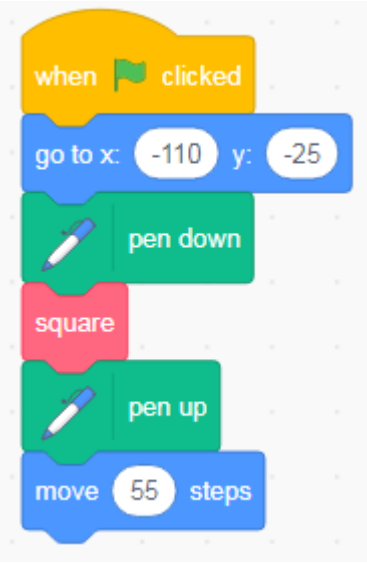


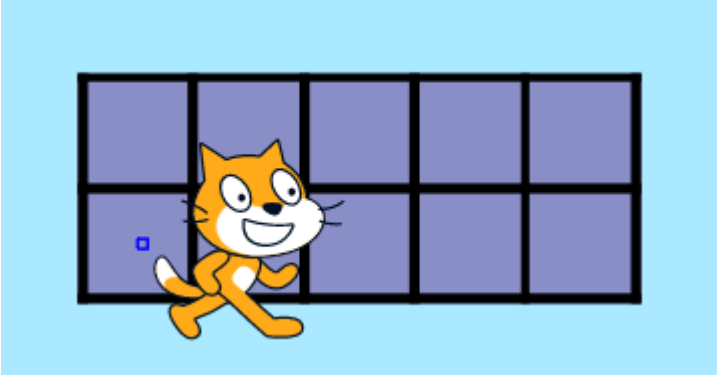
In the centre of the first square, we want to draw a little shape. Add code to draw a square, either using a repeat, or using **My Blocks** to create a new block to move in a square.





The centre of the next square is approximately 55 steps away. Add another ‘move __ steps’ to move into the next square.





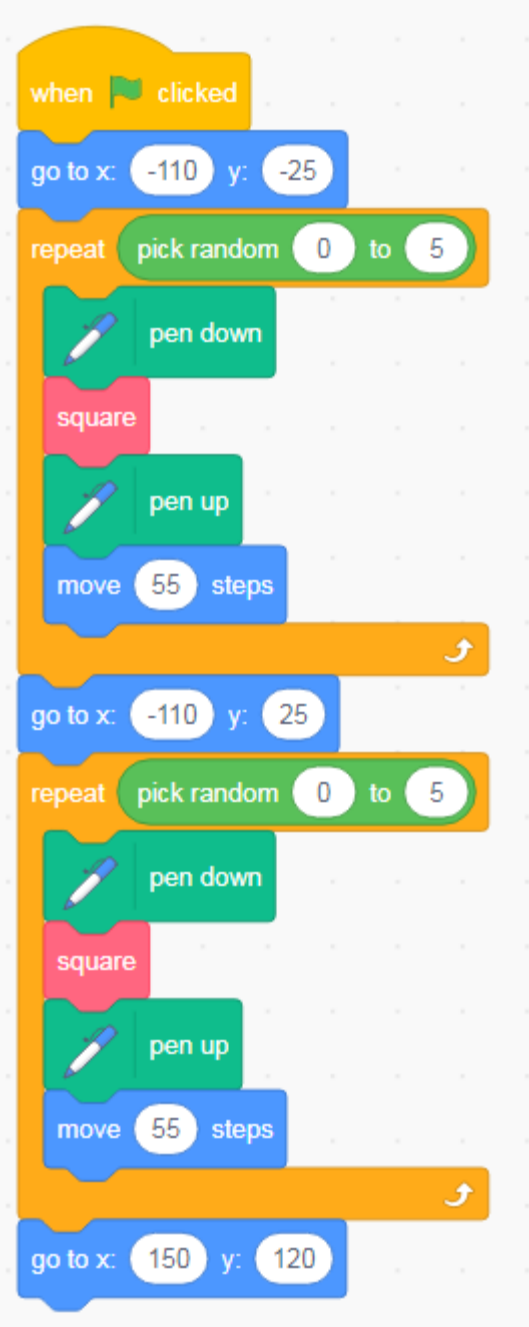
We want to repeat this set of code until we’ve drawn a square in each space on the bottom row of the tens frame, after which we’ll move to the upper line. For this, we can use a repeat. Add a repeat block to your code.

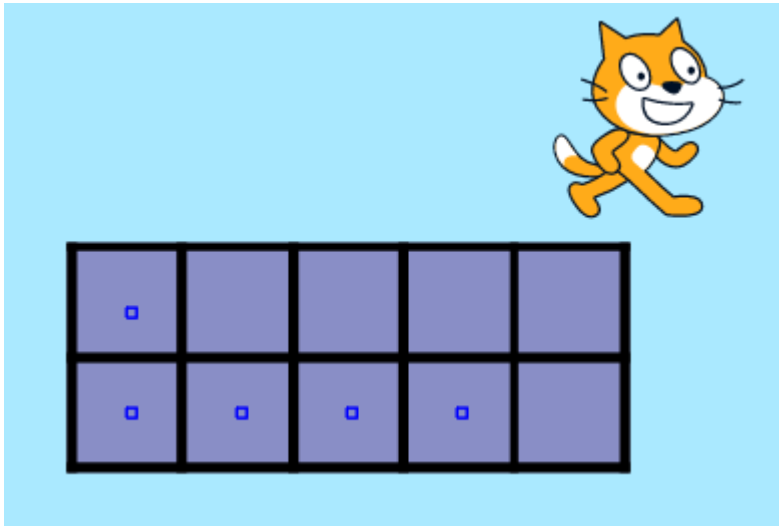
Random Numbers

We do not necessarily want to fill in the same number of squares each time we use the tens frame. Instead, we can use a random number generator to pick how many squares our sprite fills in. From **Operations**, find the ‘pick random __ to __’ block and drag it inside the white space in the repeat. We have five squares in a line, so set the block to pick a number between 0 and 5.

Computers find random numbers very difficult to create, they need order and instruction after all! Instead, many instances of random numbers are created using a pseudorandom number generator. This is an algorithm that creates a list of numbers that looks random to us humans!

Our next step is to move to the square in the top left corner of the tens frame. This is position, x:-110, y:25. Once there, we can repeat our draw square, move forward code! Finally, it may be a good idea to move the sprite away from the tens frame so that your work is visible.





Tip: An 'erase all' block from **Pen** at the start of your code could be useful if you want to create different tens frames one after the other.