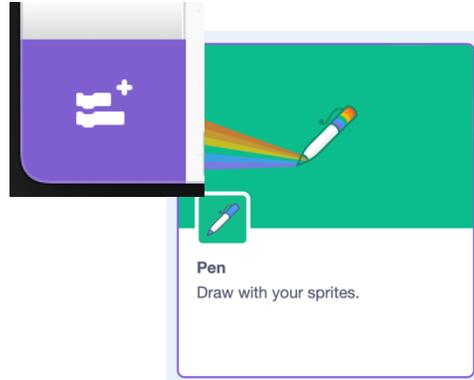


## Chasing Shapes

In Scratch, we can draw some interesting patterns by allowing animals to chase one another.

The first thing we should do is add the pen component to Scratch.



Let's leave our Scratch cat alone for a moment, and add a Scratch mouse to the stage. We're going to make our own blocks that we can swap out as we choose to make our mouse draw different shapes.

In order to make our own blocks, we first need to click on the My Blocks area and then click on Make A Block.

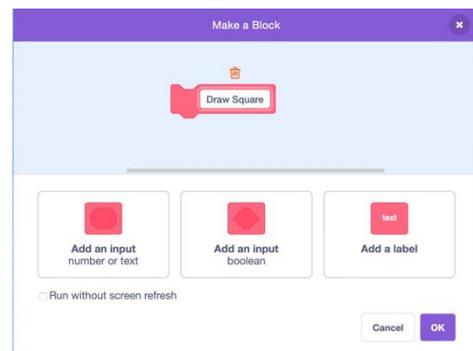


### My Blocks

Make a Block

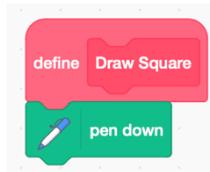
We will start simply by making a block that can draw a square.

When we click Ok, we will see a new pink block in scratch that we can add code to.



Under this block, we put in all the commands we need for drawing a square.

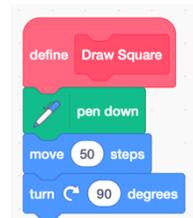
First we need to put the pen down



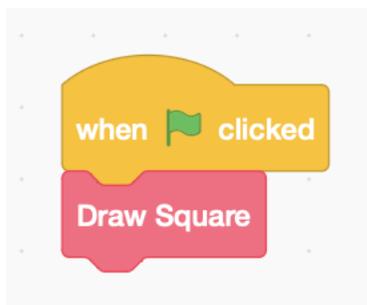
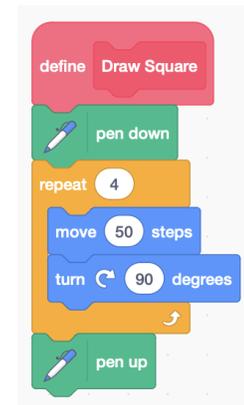
Then we need to move some amount of steps to draw a line



Then we rotate 90 degrees to make a corner



And a square has 4 sides, so we repeat this four times and when we're done we lift our pen from the stage.



Now that we have this, we can draw squares all we want by using our new custom block.

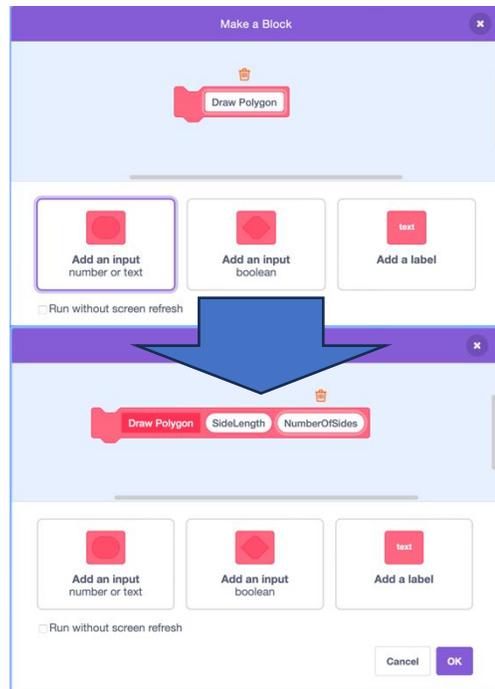
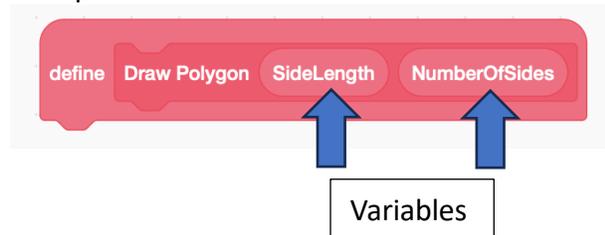
The problem is that our square is always going to be the same size and it's always going to be a square.

What do we do if we want to make other shapes? We can make a block that takes in numbers!

Let's make a second block called Draw Polygon, this time, before clicking ok, we're going to add some inputs.

We click the *Add an Input number or text* **twice**, and we will name our inputs SideLength and NumberOfSides.

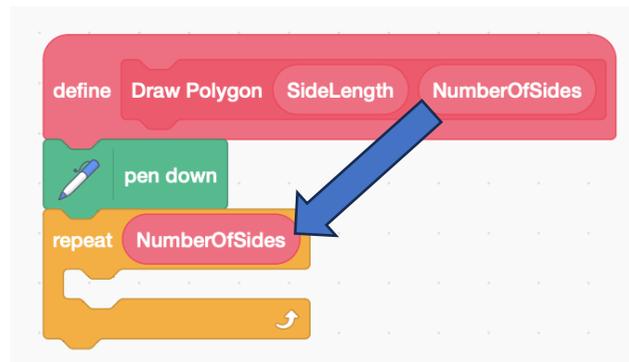
Now when we click Ok, we see that we get another **define** block, but it has two **variables** that we can manipulate.



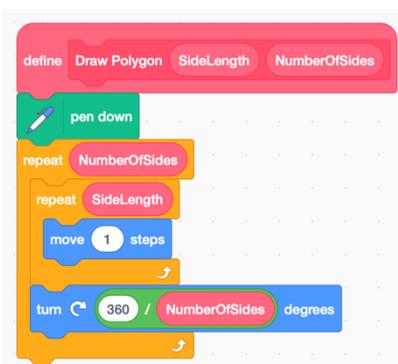
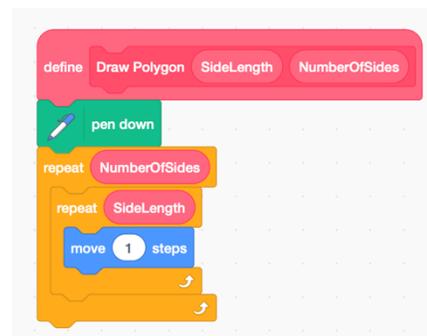
These variables are special in that they only exist for the define block. We can use them in the define block, but not really anywhere else.

First, we should remember to put our pen down. Then, just like how a square has 4 sides, and we repeated making sides and turning 4 times, if our shape has *NumberOfSides* sides, we should repeat our code *NumberOfSides* times.

We can get the *NumberOfSides* variable into the repeat block by dragging it from the define block.



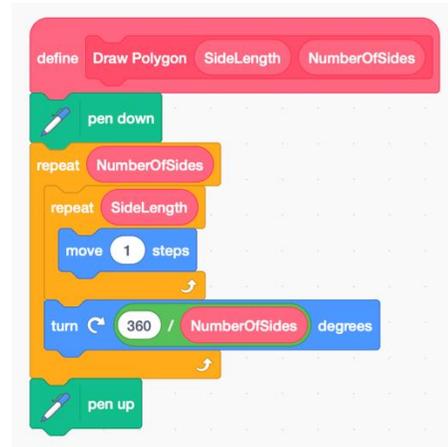
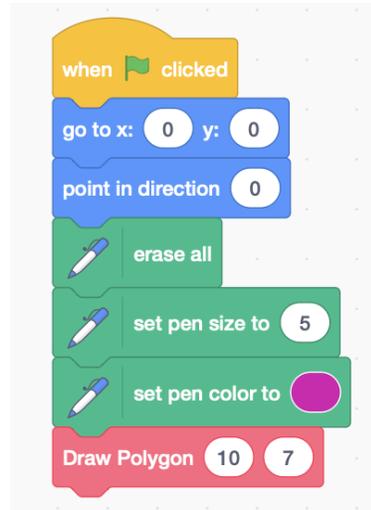
Now, in order to simulate our mouse walking, we are going to move 1 step at a time and repeat this for how long we want each side to be.



After we finish drawing a side, we need to rotate so we can draw the next side. The amount that we need to turn involves a little bit of math,

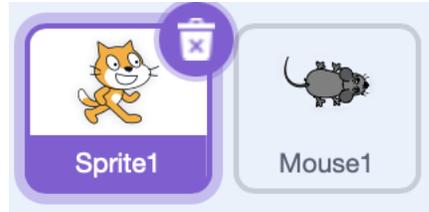
Of course, to finish off we should lift up our pen again.

We can use this block just like we use any other block. Test out your next block by changing the numbers in the block!



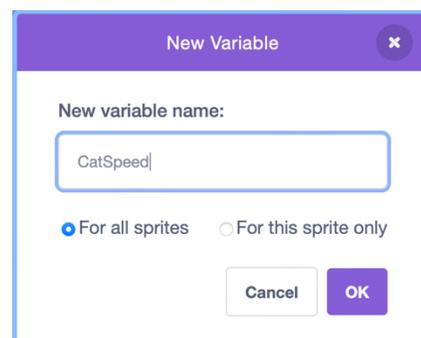
Now, we have a cat to deal with too. It turns out, that animals chasing each other can sometimes make some pretty cool looking patterns.

Let's go back to our Cat Sprite.

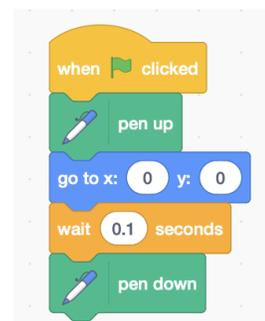


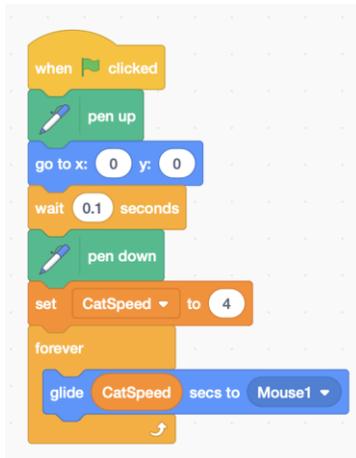
We want the cat to chase the mouse, and we want it to draw its path.

First, we should determine how fast our cat will move. Let's create a variable called CatSpeed.



Next, we want to make sure that our cat's pen is up, and we want to make sure it starts in the same place as the mouse, then we will put our pen back down. We also want to give the mouse a fighting chance, so we give it a bit of a head start.





Next, we will set the Cat's speed. Then, we make sure that the Cat follows the Mouse around forever.

We can even make more animals that all follow each other around using the same code.

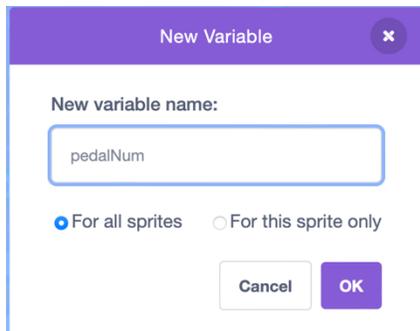
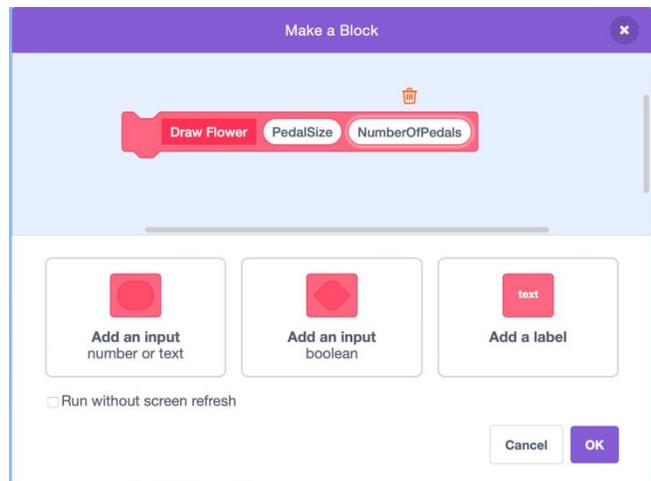
For instance, we can make a variable called DogSpeed and then use the same code to make the dog, chase the cat that is chasing the mouse.

The more animals we add the more complex our patterns will get.

### Drawing a flower

We can also build a block that allows us to draw a flower. This is quite a bit more complex than just drawing shapes, but follows the same principle.

First, we create a new block called Draw Flower that has two inputs: PedalSize and NumberOfPedals.



We also need a variable called pedalNum that keeps track of how many pedals on a flower we've drawn.

We start with the same skeleton of a block as before: We put our pen down, we will set our pedalNum to 1 because we will start by making our first pedal, and we will want to repeat making pedals the specified number of times before eventually lifting our pen up again.

```

define DrawFlower PedalSize NumberOfPedals
  pen down
  set pedalNum to 1
  repeat NumberOfPedals
  pen up
  
```

```

define DrawFlower PedalSize NumberOfPedals
  pen down
  set pedalNum to 1
  repeat NumberOfPedals
    repeat 360 / NumberOfPedals
      repeat PedalSize
        move 1 steps
        turn 1 degrees
    pen up
  
```

To draw half a pedal, we slowly move 1 step at a time while simultaneously changing our direction by 1 degree, and we repeat this  $\frac{360}{\text{NumberOfPedals}}$  times. The number 360 allows our pedals to look round (because circles have 360 degrees in them)

We have to draw the other half of our pedal. First, we have to turn around, then we repeat our steps while facing the opposite direction.

```

define DrawFlower PedalSize NumberOfPedals
  pen down
  set pedalNum to 1
  repeat NumberOfPedals
    repeat 360 / NumberOfPedals
      repeat PedalSize
        move 1 steps
        turn 1 degrees
    turn 180 - 360 / NumberOfPedals degrees
  pen up
  
```

Before we continue, try testing your new block to see how one pedal is drawn. If we change the number of pedals or the size, how does the pedal change?

```
when clicked
go to x: 0 y: 0
point in direction 0
erase all
set pen size to 5
set pen color to purple
DrawFlower 2 5
```

Now back to our block definition. Once we've completed one pedal, we have to point in a new direction so we can draw the next pedal. We also have to make sure in increase our pedalNum variable by 1 since we've completed a pedal.

This is what our completed flower block definition should look like.

Use these blocks along with other chasing animals to draw fancy patterns.

**Challenge:** Can you get the mouse to run away from the cat that is chasing it? What kind of pattern does this produce?

```
define DrawFlower PedalSize NumberOfPedals
pen down
set pedalNum to 1
repeat NumberOfPedals
  repeat 360 / NumberOfPedals
    repeat PedalSize
      move 1 steps
      turn 1 degrees
    turn 180 - 360 / NumberOfPedals degrees
  repeat 360 / NumberOfPedals
    repeat PedalSize
      move 1 steps
      turn 1 degrees
  point in direction 0
  turn 360 * pedalNum / NumberOfPedals degrees
  change pedalNum by 1
pen up
```