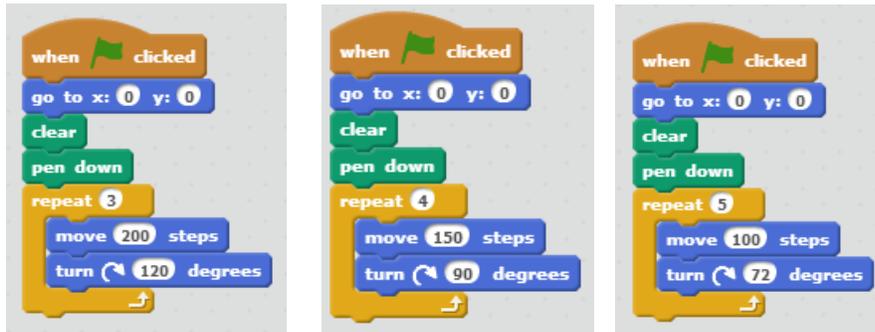
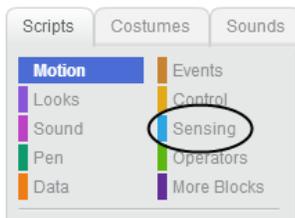


Drawing Any Regular Shape in Scratch

Now that you've drawn many different regular shapes in Scratch, you've likely noticed that the scripts to do each have a lot of similarities. Look at the scripts for drawing a triangle, square, and pentagon:



Everything before our  block is exactly the same! And everything in our  block is the same, except the number of steps, the number of degrees to turn, and the number of times to repeat. We can take advantage of this similarity to write a Script which allows Scratch Cat to draw *any* regular polygon we can imagine! To start, we need to know how many sides to draw. Open the Sensing palette by clicking the word "Sensing":

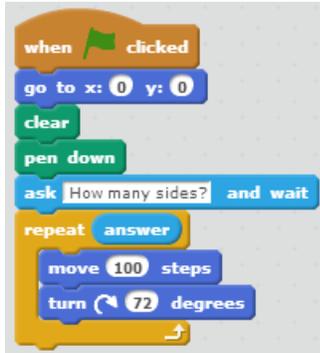


In this palette, we see this block: , which asks the user of our program a question, and then waits for an answer. Drag it to your script area, and add it to your script

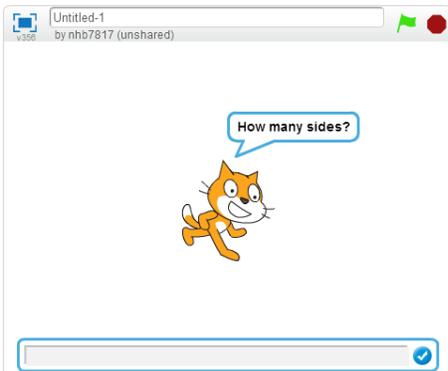
right before your  block. Change the question it asks to "How many sides?" Your script should now look like this:



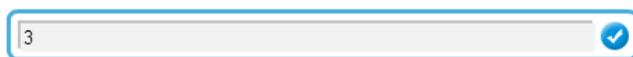
Now, we can find out what answer our program's user gave us with the `answer` block. It will always tell us the last answer the program received. We can use this in our `repeat` block, as it tells us how many sides need to be drawn:



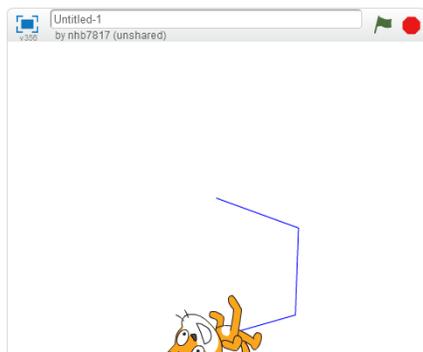
Now, when we run our program, Scratch Cat asks us how many sides we want to draw:



If we tell him 3:

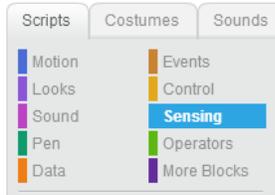


Then he'll only draw three sides:



Unfortunately, we're still using `move 100 steps` `turn 72 degrees`, which is for drawing a pentagon, not a triangle!

To draw the correct shape, we need to determine how many steps Scratch Cat should take, and how many degrees he should turn. Some blocks that may be helpful here are found in the Operators palette.



These are:

-  (adds the numbers in the two sockets)
-  (subtracts the numbers in the sockets)
-  (multiplies the numbers in the sockets)
-  (divides the numbers in the sockets)

You can either type a number into a socket, or drag in the  block. The operator blocks (, , , ) can also be dropped into a socket, like those in your  and  blocks. Use these blocks to finish your program!

© 2013 Nathan Bean

Released under the [Creative Commons By Attribution](https://creativecommons.org/licenses/by/4.0/) license

