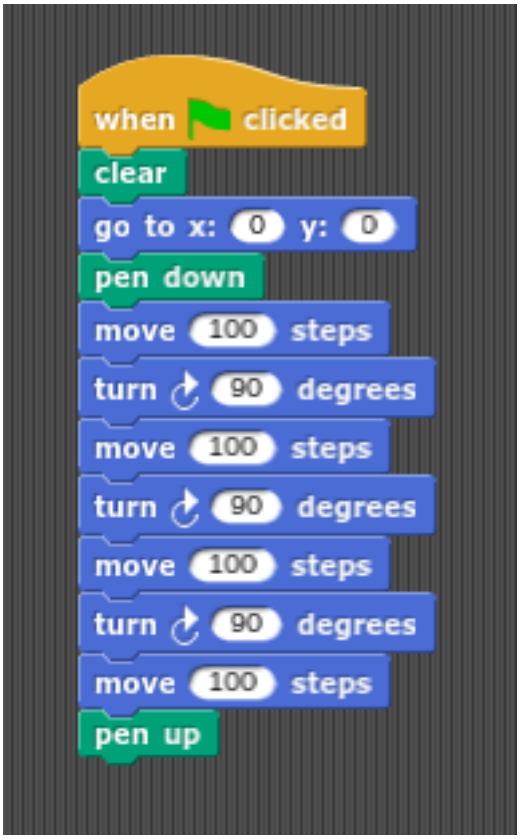
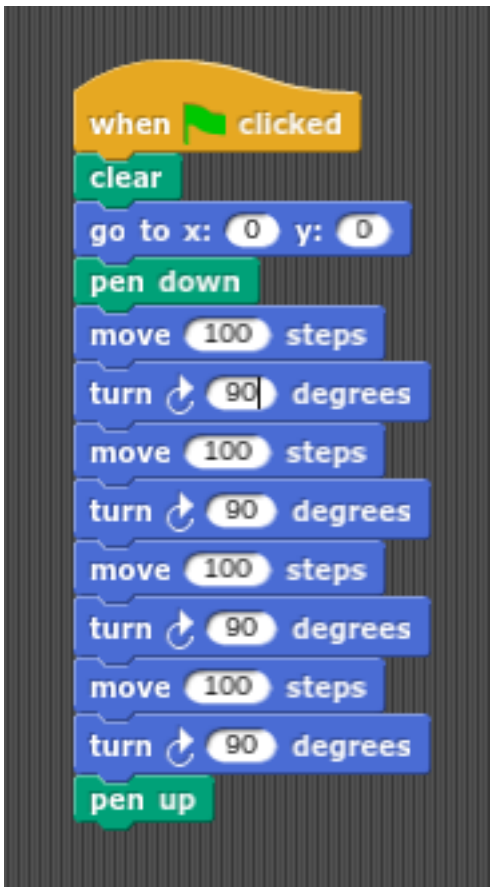
An introduction to programming

Here is a simple programme for drawing a square; we will start with this and try to make more complicated drawings.

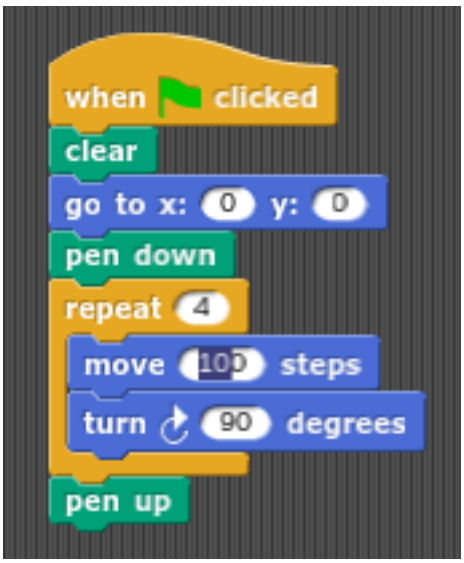


Enter this and make sure it draws a square! One thing about this program is that after it draws the square the arrow ends up pointing a different direction to the direction it started in; can you fix this?

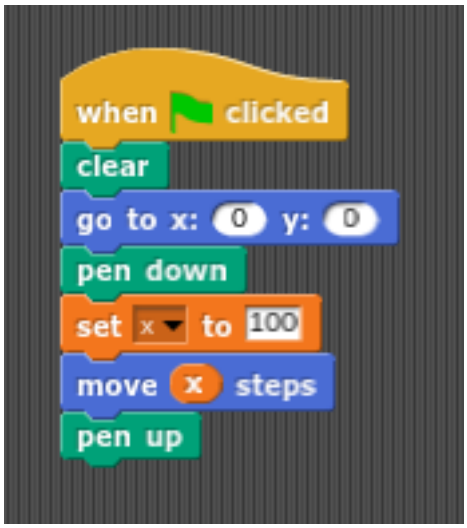


Repeat

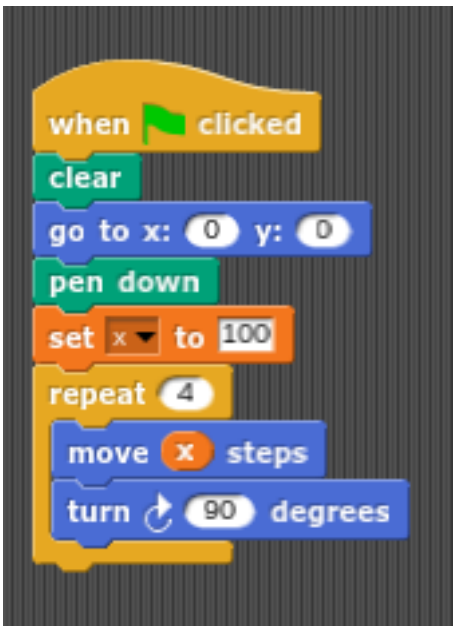
Can you use that to make the programme more succinct and readable?



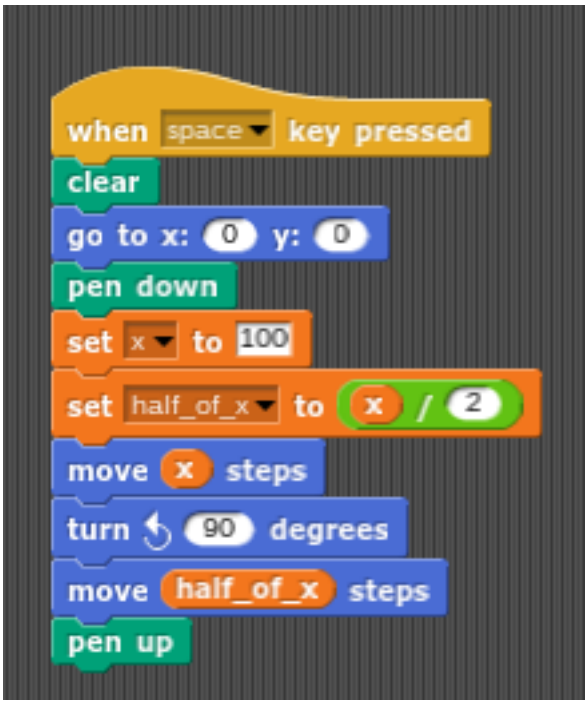
Now, look at this programme



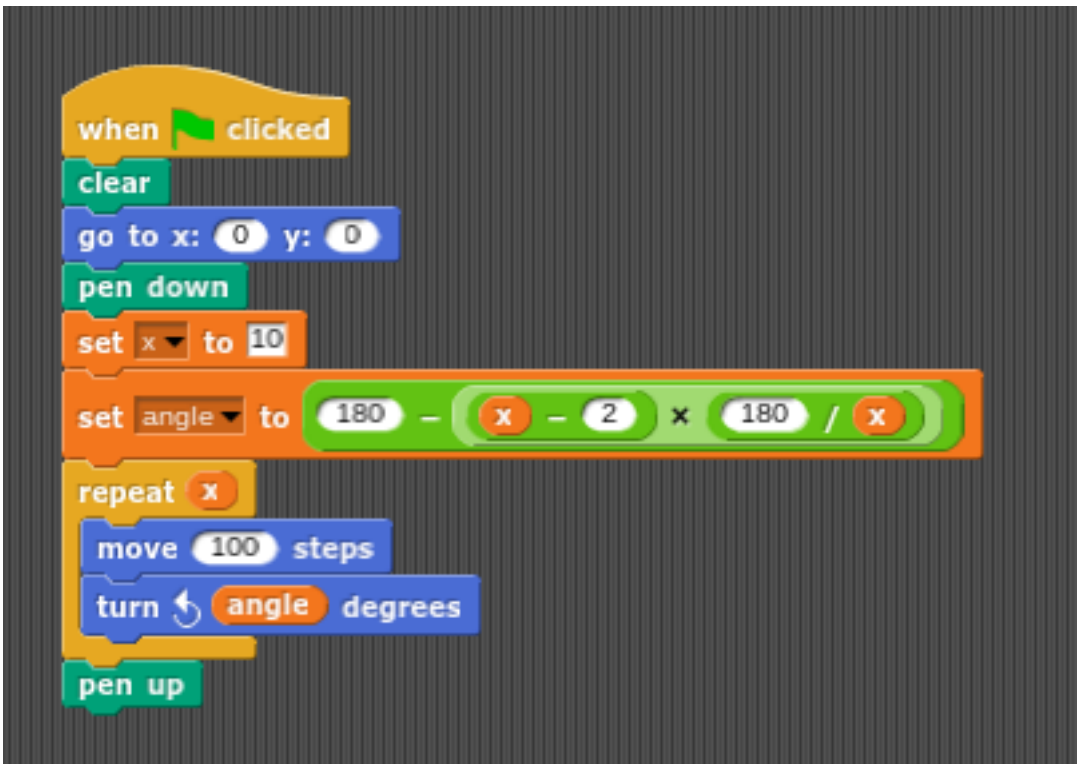
Do the same to your square programme!



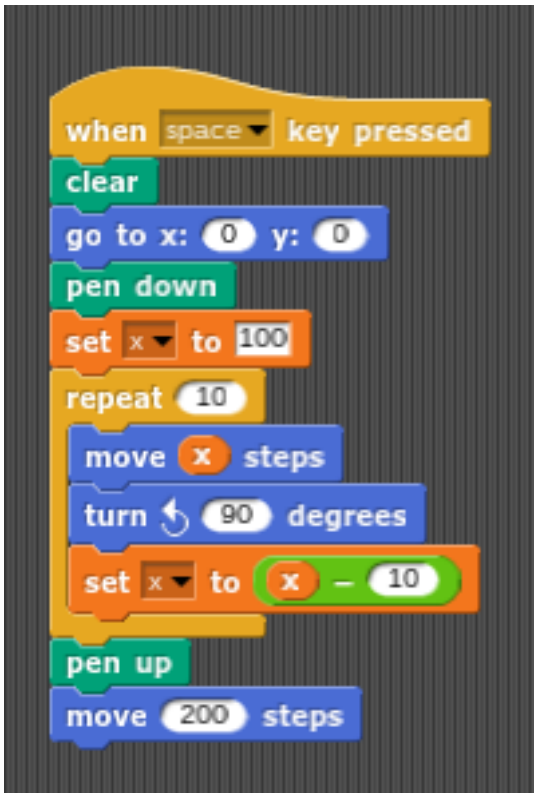
This programme does something slighly more useful with a variable.



Try modifying your programme in a similar way so that it draws an n-gon.

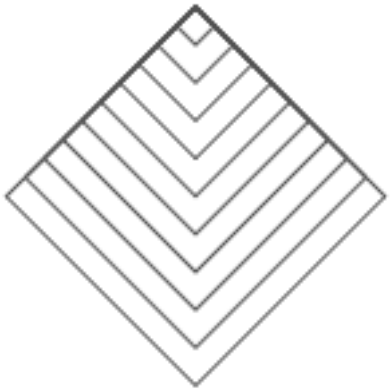


In this programme the variable is changed in the *loop* so the line is shorter each time:

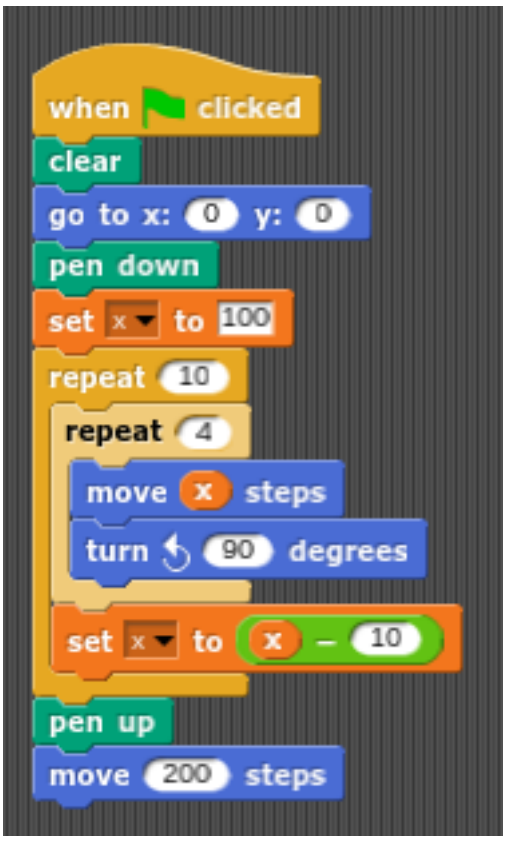


giving a spiral

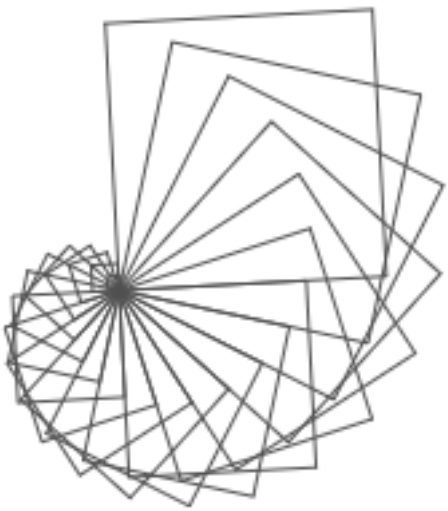
Try modifying your programme in the same way so that you get smaller and smaller squares retreating into one corner, like this



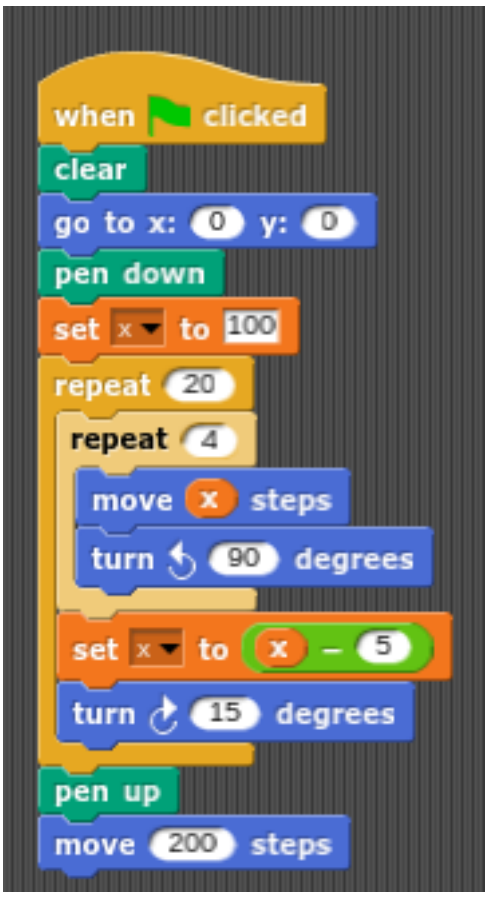
so



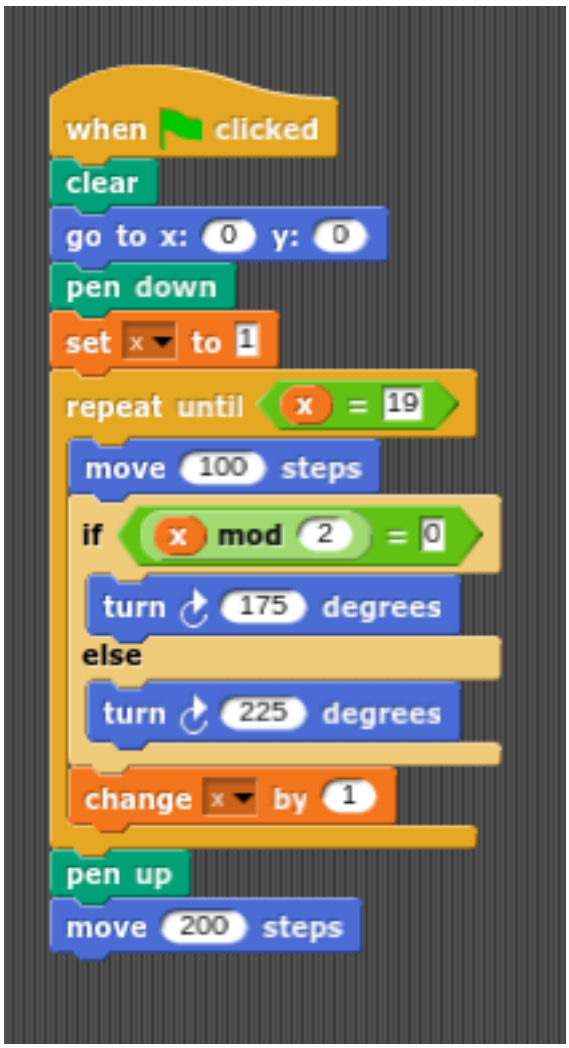
If you want to you can try playing with you program a bit to give other patterns, like this



so

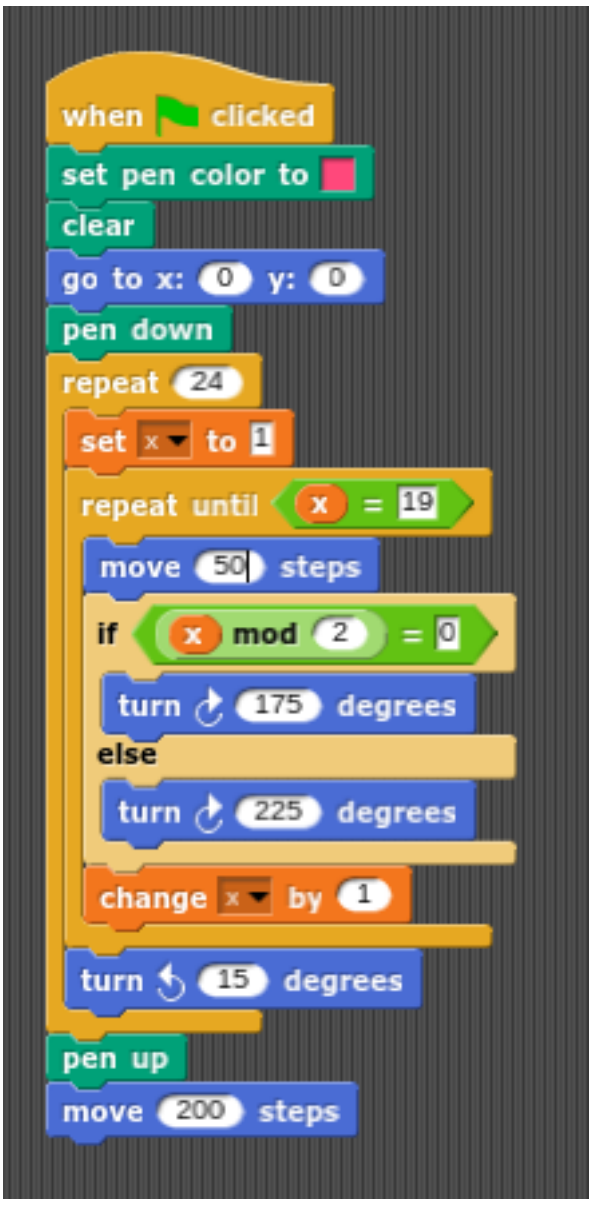


This next programme draws a star;
This next programme draws a star

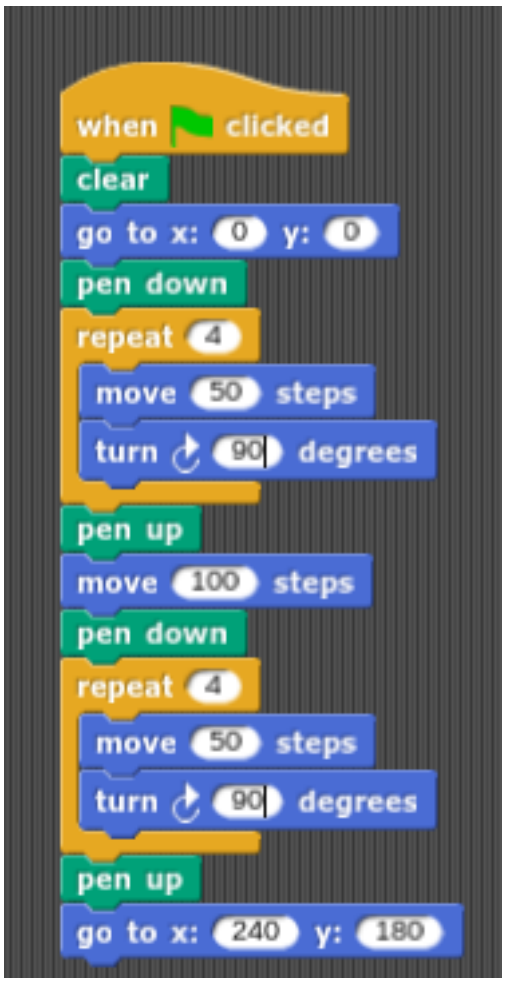


You can mess with programme a bit, maybe changing the angles or putting the whole thing in a loop to give something like this

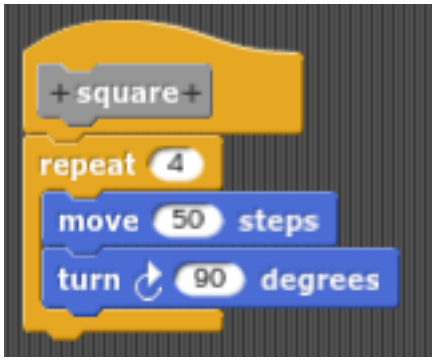
so



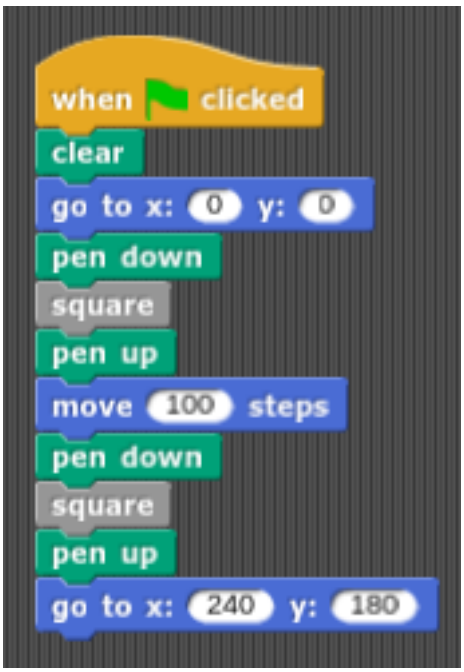
Imagine you want to use the same commands a few times; in this programme for example we draw a square, move over a bit and draw another:



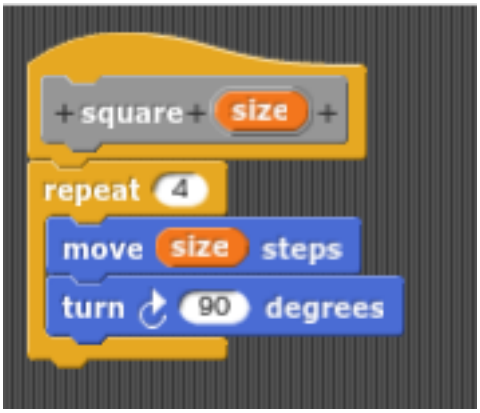
Here we will make a block called square that draws a square: the block commands are at the bottom of the variable menu:



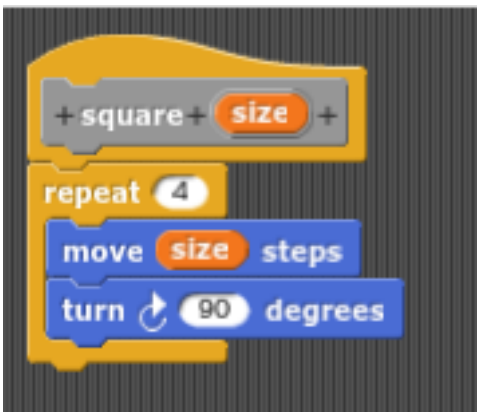
This block draws a square, so our two square code becomes a bit neater, quicker to input and easier to read:



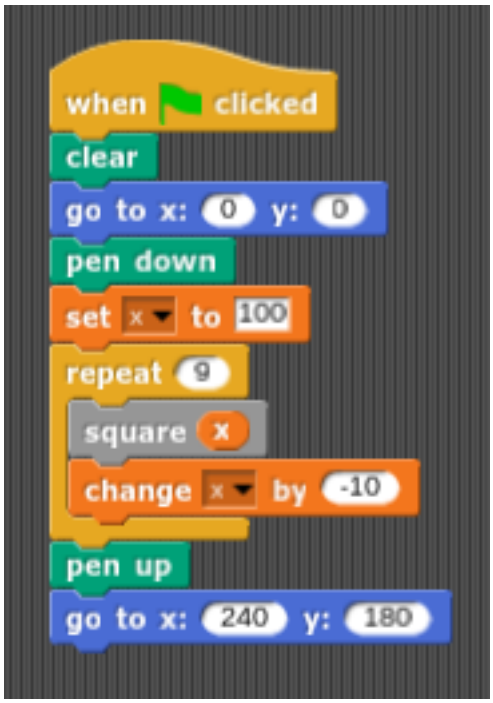
You can make your blocks more flexible by adding *arguments*; these are variables that work inside the block that you can send from the main programme, you make them by clicking the plus by the block name.



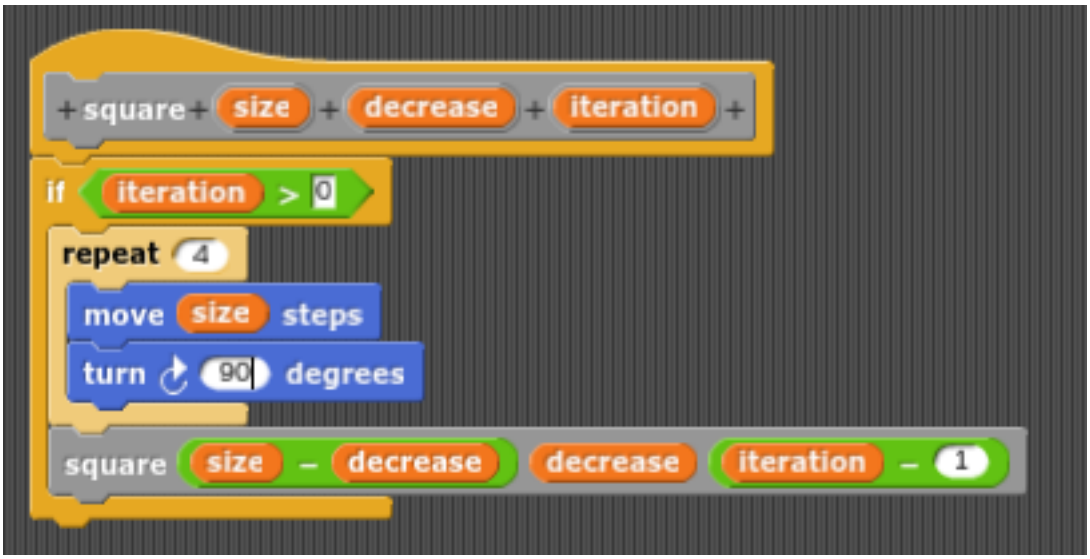
and



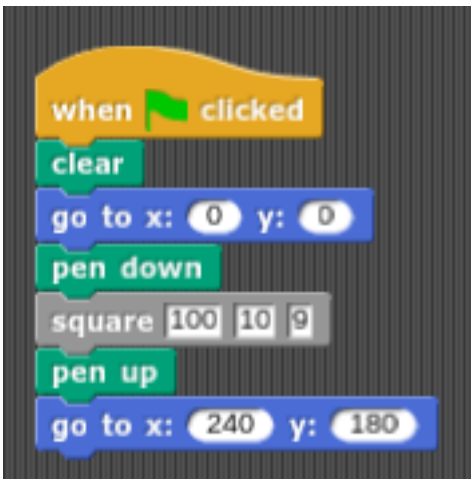
draws the two squares different sizes. Try rewriting your original shrinking square code with blocks.



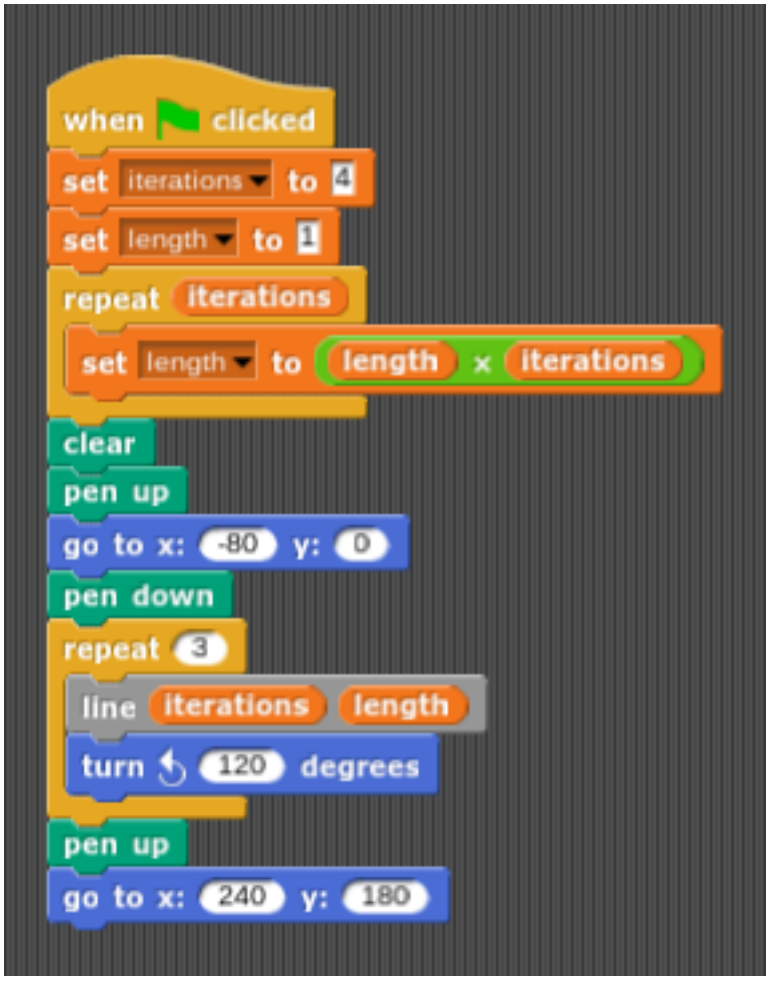
If they finish this, maybe try recursion:



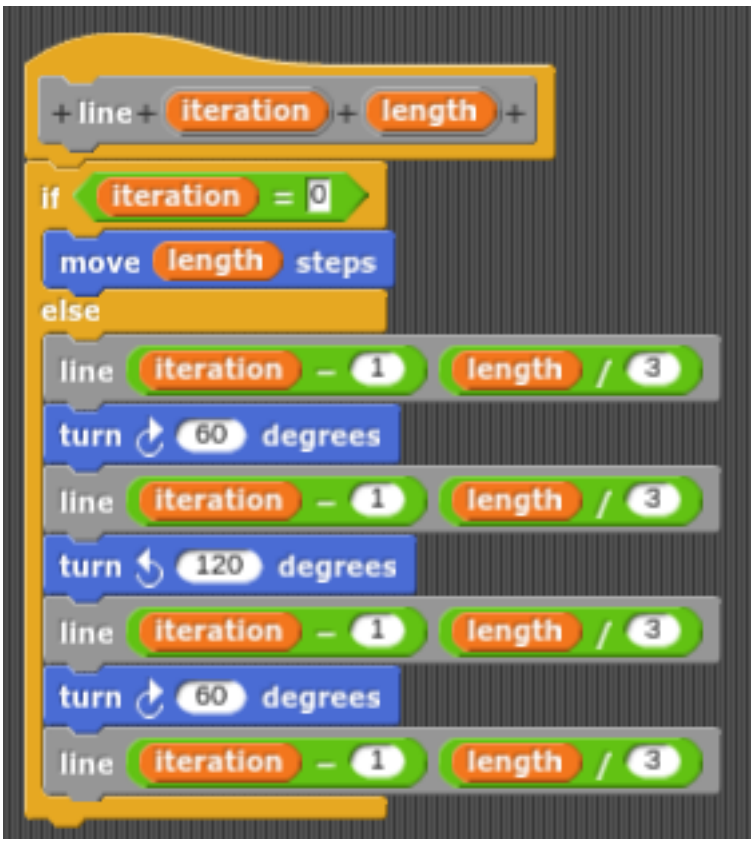
draws the two squares different sizes. Try rewriting your original shrinking square code with blocks.



If anyone is finished with all of this maybe go on to this



with block


+ line + iteration + length +
if iteration = 0
move length steps
else
line iteration – 1 length / 3
turn 60 degrees
line iteration – 1 length / 3
turn 120 degrees
line iteration – 1 length / 3
turn 60 degrees
line iteration – 1 length / 3

draws