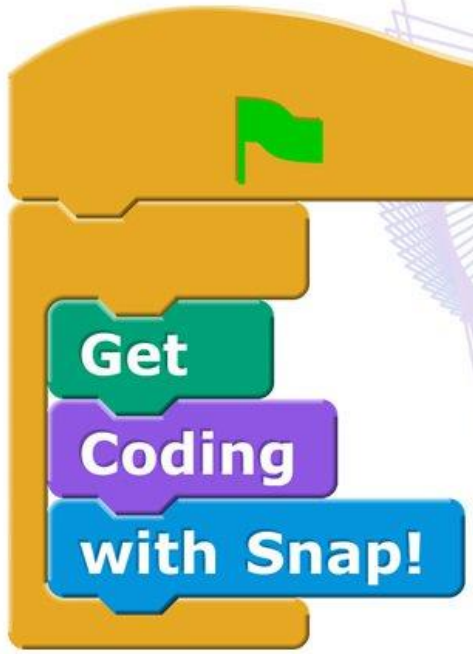
openSAP Get Coding with Snap!

Exercises Week 2 Unit 4



PUBLIC

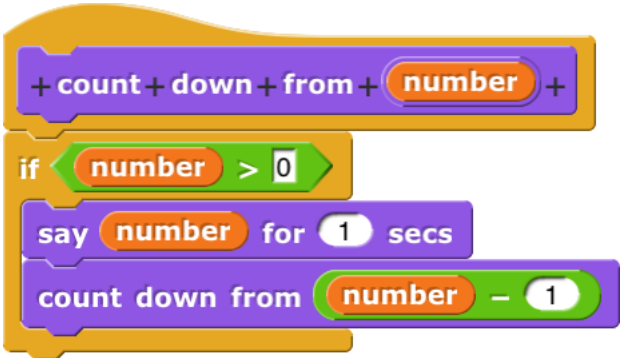
WHAT YOU HAVE LEARNED THIS WEEK

Recursion

We speak of recursion when you use a custom block inside its own definition. Recursive blocks go on forever, unless you specify a condition when they should stop or under which they should continue. Often that condition will be the “base case”, i.e. the part of the algorithm that doesn’t need recursion.

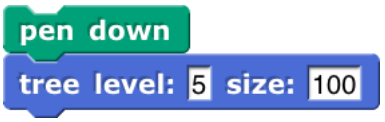
Remember how we made the countdown block:

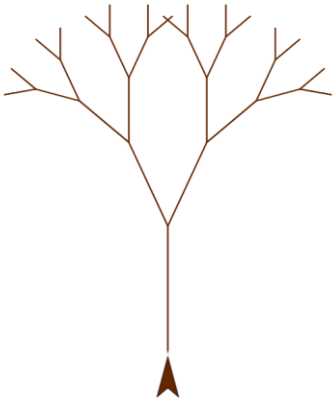
In its definition script we check whether the input number to count down from is greater than zero, otherwise we don’t let the block do anything, because we’re already done. In cases where the input is greater than zero we let the sprite say it for a second, and then we count down from the number that’s one below the current one:



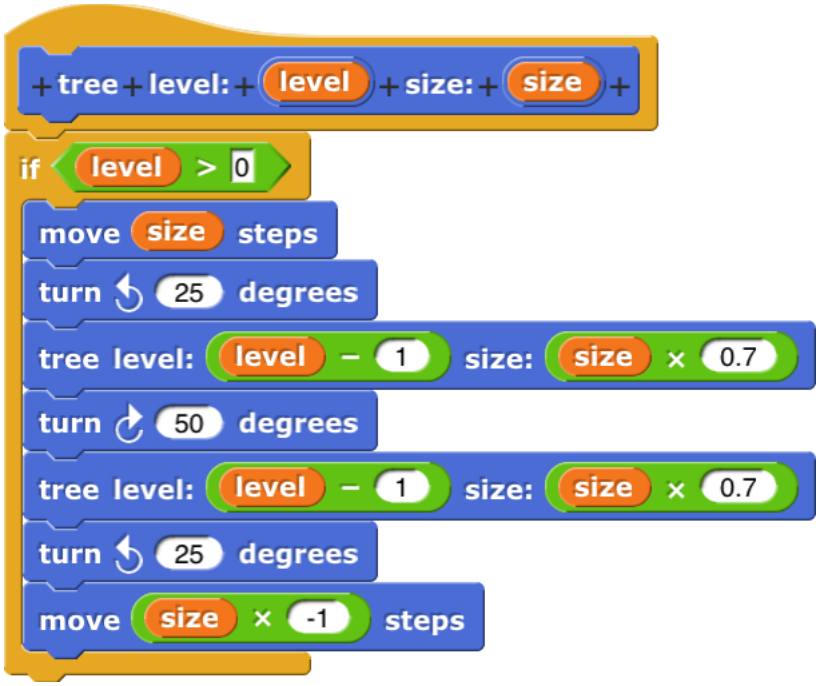
When a block definition only calls its own block once the script accomplishes repetition, and we speak of “linear” recursion. This is cool, because it gives us the superpower to get by without ever having to use any of the loop blocks (forever, repeat, repeat until).

You can also use a custom block more than once inside its own definition. Then we speak of exponential recursion, because instead of just a single repetition we get branches. For this we made a block that draws a tree:





We used an input that we named “level” much in the same way that we counted down the start number in the first example, checking whether it is greater than zero, and stopping (not doing anything) once it is:



YOUR TURN

(1)

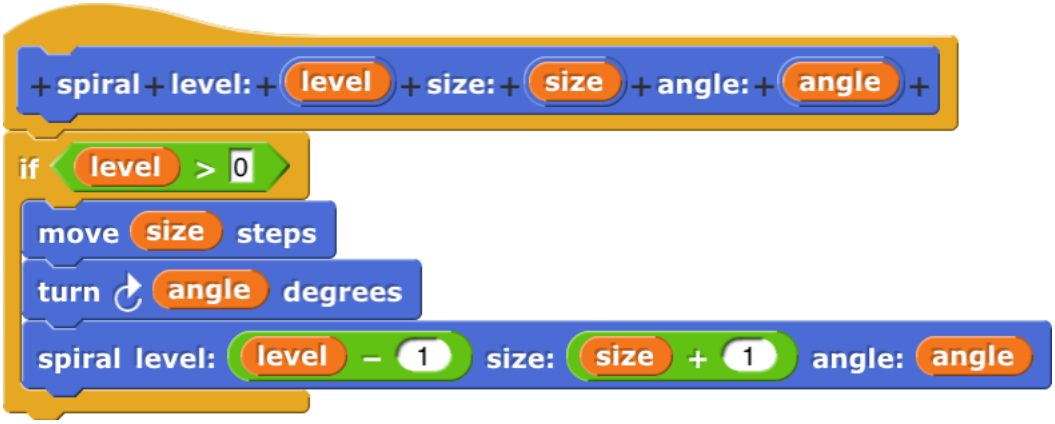
Make a recursive block that moves the sprites over a given distance, going one step at a time, Let's name it "float" and make it look like this:

Here's how it should behave:

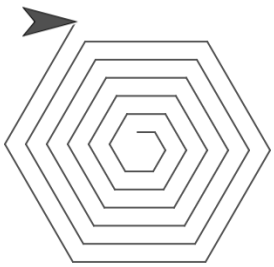
But can you use recursion instead of the "repeat" loop to make it do the same thing?

(2)

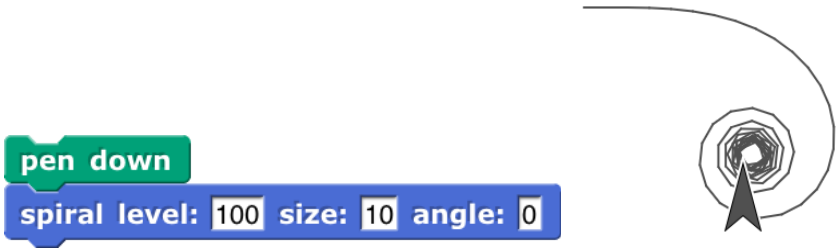
Make a "spiral" block (copy this code):



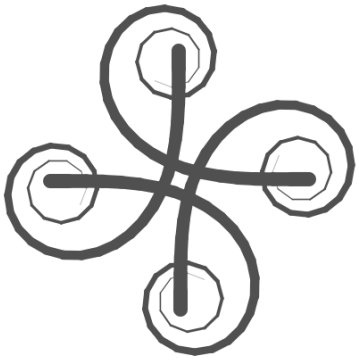
And test it:



Now play with the block definition and change how the values for “angle” and “size” are modified in the recursive call. Can you come up with something like this? (Hint: only change the angle instead of the size, and also look at these numbers…):



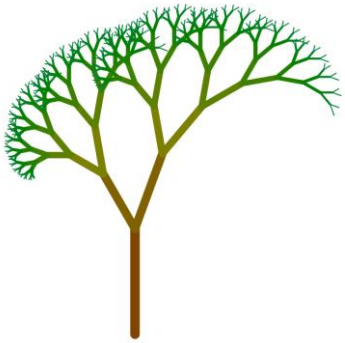
Are you having fun yet? Then how about you refine your “Euler Spiral” (that’s what this figure is called) to draw something calligraphic:



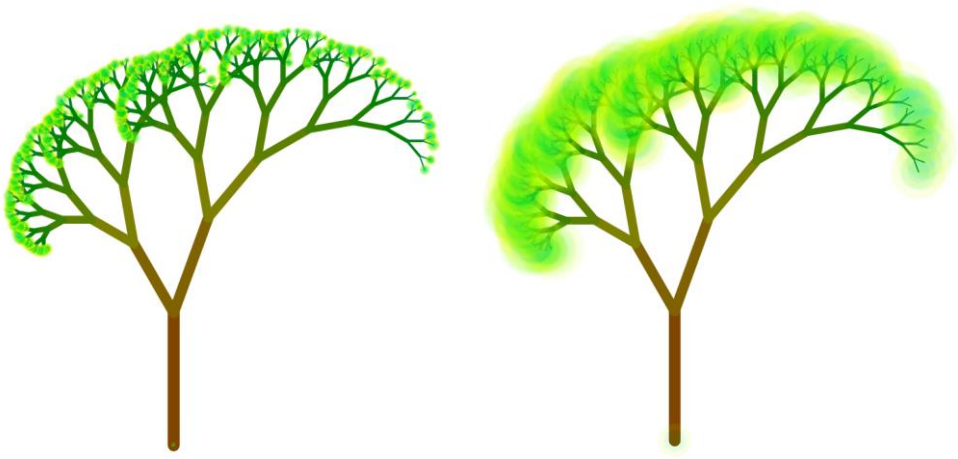
Can you come up with your own calligraphic figures?

(3)

Let’s play with the “tree” block! Can you reproduce the block from the video? It’s the same definition that you can also find above in this document. Remember how in the video we made the tree look more like a natural tree by setting the pen size to the level of the block definition? Experiment with the angles and the sizes in the recursive calls, and also perhaps with the pen color. Can you make your tree more life-like yet?



You could also try drawing a costume for a leaf and then changing the tree definition so it stamps the sprite whenever the base-case is reached. And you can play with adding some randomness and a graphical ghost effect before stamping the leaves:



These are just examples. Why don't you fool around with the code and share your interesting findings in the discussion forum. We're curious what you'll invent!

Coding Samples
Any software coding or code lines/strings ("Code") provided in this documentation are only examples and are not intended for use in a production system environment. The Code is only intended to better explain and visualize the syntax and phrasing rules for certain SAP coding. SAP does not warrant the correctness or completeness of the Code provided herein and SAP shall not be liable for errors or damages cause by use of the Code, except where such damages were caused by SAP with intent or with gross negligence.

www.sap.com/contactsap

THE BEST RUN SAP®