Recursion Activity

Recursion is the process of breaking a problem down into smaller versions of itself. Today we’re going to experiment with various fractal shapes and create our own.

In particular, notice that:

* Recursion involves each of the functions calling itself at least once
* We use a “base case” to avoid calling ourselves *forever*

Assignment:

First, we’re going to experiment with the tree sprite. Click on the sprite and look at the create\_tree function.

1. What is the input and the output of the create\_tree function?
2. How many recursive calls does the create\_tree function have?
3. Augment the program (using “when green flag clicked”) to draw the tree. Be sure to put the pen down. If you’re not happy with the output, change the min\_branch\_size variable.
4. In a 2-3 sentences, explain how different values of min\_branch\_size affect the overall shape of the tree?
5. Alter the create\_tree function so its branches are at right angles. Call Mr. Rizzi over to check the result

Check box for Mr. Rizzi to verify the tree is correct

1. Alter the create\_tree function so at the end of each branch the sprite stamp’s itself (use the stamp block in the pen section). Call Mr. Rizzi over to check the result.

Check box for Mr. Rizzi to verify the tree is correct

Now, we’re going to experiment with the snowflake sprite. Click on the sprite and look at the create\_koch\_flake function.

1. What is the input and the output of the create\_koch\_flake function?
2. How many recursive calls does the create\_koch\_flake function have?
3. Create a program (using “when green flag clicked”) to draw following shape – be sure to set min\_koch\_distance.

A colorful pattern on a white background

Description automatically generated

1. In a 2-3 sentences, explain how different values of min\_koch\_distance affect the overall shape of the tree.

Now it’s your turn. In the space below, draw a self-similar pattern. Then, encode it in a program and see how it turns out.