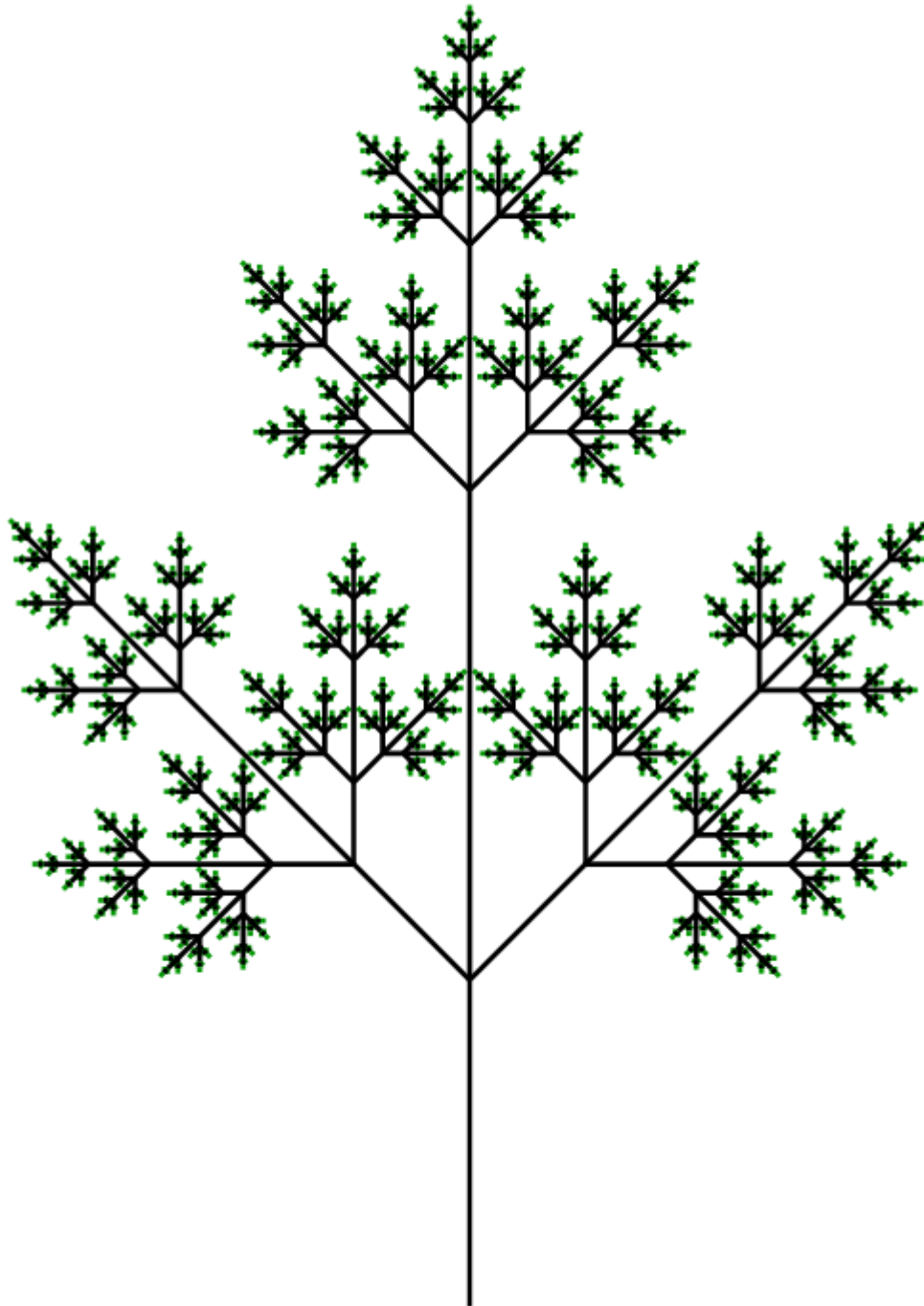


Documentation

This is documentation to provide information how to use web application for L-Systems generation. It is in form of three tutorials which cover all controls found in application.

The Basics


This tool will let you use LSystems to create cool images. Don't worry if you don't know what LSystems are, it is not necessary for use of this tool. In this first tutorial we will cover the basics of this tool and create this image of tree:




How are these images created? Everything is based on creating **modules** and **rules** which are later applied over these modules. What are these, so called, modules? You can think about them as actions needed to draw some image. The tool reads all modules you have inputted and transforms them into image. You can imagine this process as if you were drawing with a pen, following orders given to you by someone else. You just read these orders and move your pen accordingly.


One module we will definitely need is a module which will draw a line, as the picture above is all created of lines. When you first open the editor, you will be at the modules screen:



 Modules

 Axiom

 Rules

 Result

Modules

Define modules which will be used in LSystem

No Module was defined, start by clicking on button below.

Create new module

To create new module just click on the blue button. In the form that showed on the screen you just have to input name of your new module. You could name whatever you like, for example *line*, but it is usually better idea to name it after part of the image the module is going to represent. Our module is going to represent branch so let's name it *Branch*. You don't need to worry about parameters stuff yet, but you should change the representation to *Line* and fill in the length. You should not think about length as 1 centimeter or 1 inch but more in relative terms. If something has length 2 it will be twice as large as something with length 1. How much is this 1 or 2 will be decided later based on size of your screen. If you want you can also change the color of the line. Then click on the *Create* button.

Modules

Define modules which will be used in LSystem

No Module was defined, start by clicking on button below.

Create new module

New module

Module name:	Branch
Parameter names:	Add Parameter
Representation	Line ▼
Length:	1
Color:	
Red:	0
Green:	0
Blue:	0
	Create


Now let's create another module. We will also need to rotate the lines. To do so we will create new module and name it *Left*. As representation we will select rotation and set the angle to **45** degrees. In similar way we will create rotation to right, just change the angle to **-45** degrees.

You have maybe noticed that in the image, branches end with green part. To do so, we will need new module that will draw line of the same length as the *Branch* module, but with green color. Create module *Leaf*, which is represented as a line, and set green part of color to something like 150.


To explain how these modules will be drawn


Now we need to create two modules that will tell the tool where to draw next line. The first one will be called *Store* and its representation will be *Save position on stack*. What that means? When the tool reads this module it saves current position and orientation of the imaginary pen for later use. The other one will be called *Load* and its representation will be *Load position from stack* and it basically means that the tool will jump into last saved position and continue drawing from there. After all of this, your modules section should look like this:



 Modules

 Axiom

 Rules

 Result

Modules

Define modules which will be used in LSystem

Name	Parameters	Representation	
Branch		Line with length: 1 and color R: 0 G: 0 B: 0	×
Left		Rotation by angle: 45	×
Right		Rotation by angle: -45	×
Leaf		Line with length: 1 and color R: 0 G: 150 B: 0	×
Store		Saves current position to stack	×
Load		Loads last saved position from stack	×


Create new module


Now let's move to the Axiom part.

Axiom

Axiom is basically description how an image looks. In the blue box are shown all modules you have created. By dragging them into the green box you will create image from them. Lets drag one *Leaf* from blue box into the green box and save the changes.



 Modules

 Axiom

 Rules

 Result

Axiom

Select modules forming axiom by dragging them from the blue box into the green box

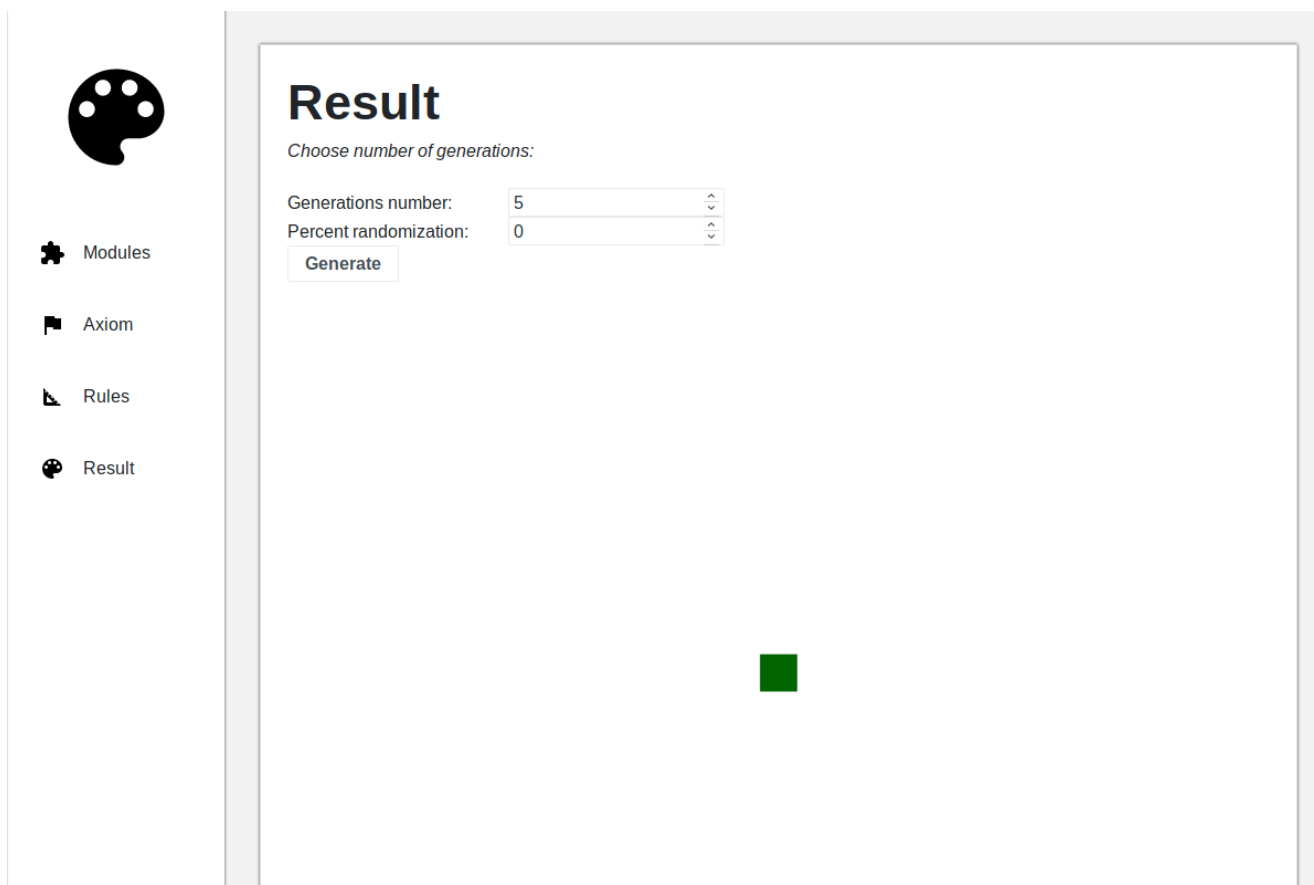
Branch Left Right Leaf Store Load

Leaf



Save

Now let's jump into the Result screen for a moment. You should see one green line or maybe a box (depending on the length you have chosen)



Well that was a lot of work just to get one line drawn. But don't worry, everything will get much quicker from here. Lets look into the Rules screen.

Rules

You can think of rules as creating images from other images. This tool will generate image you have defined in axiom screen and then apply rules over it. And then again. And again. You can choose how many times the rules are going to be applied, but about that later.

Lets create a new rule. Source module determines which rule will be applied. When the image is generated, the tool goes over it, and when it finds corresponding module, replaces it with other modules. Lets choose *Branch* as our source module and drag two other *Branch* modules from the blue box to the green box. Don't worry about *Advanced Options* and click on *Add rule*. We have now created rule that will transform one branch into two branches which are directly one after other.

Now let's create the second rule. Select *Leaf* as the source module and drag modules from the blue box to correspond to this image:

New rule

Source module: Leaf

Next generation:

BranchLeftRightLeafStoreLoad

BranchStoreLeftLeafLoadStoreRightLeafLoadBranchLeaf

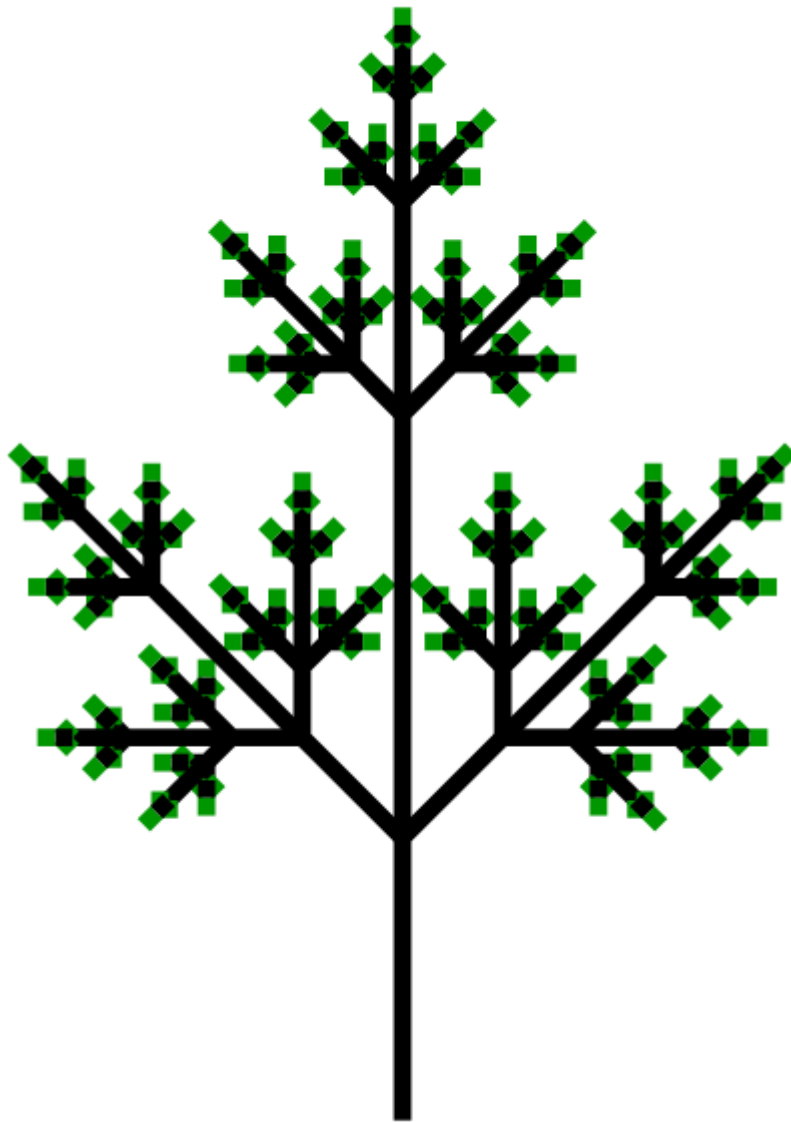


Show advanced options

Add rule

Now just click on *Add rule*. What does the rule we have just created means? Well it tells to replace any *Leaf* with Branch, then turn Left and draw Leaf, return to trunk, turn right and draw another leaf, and return to trunk again and draw one more branch ended with leaf. It may seem complex at first but it really isn't.

Now just switch to the *Result* screen again. You should see this image:



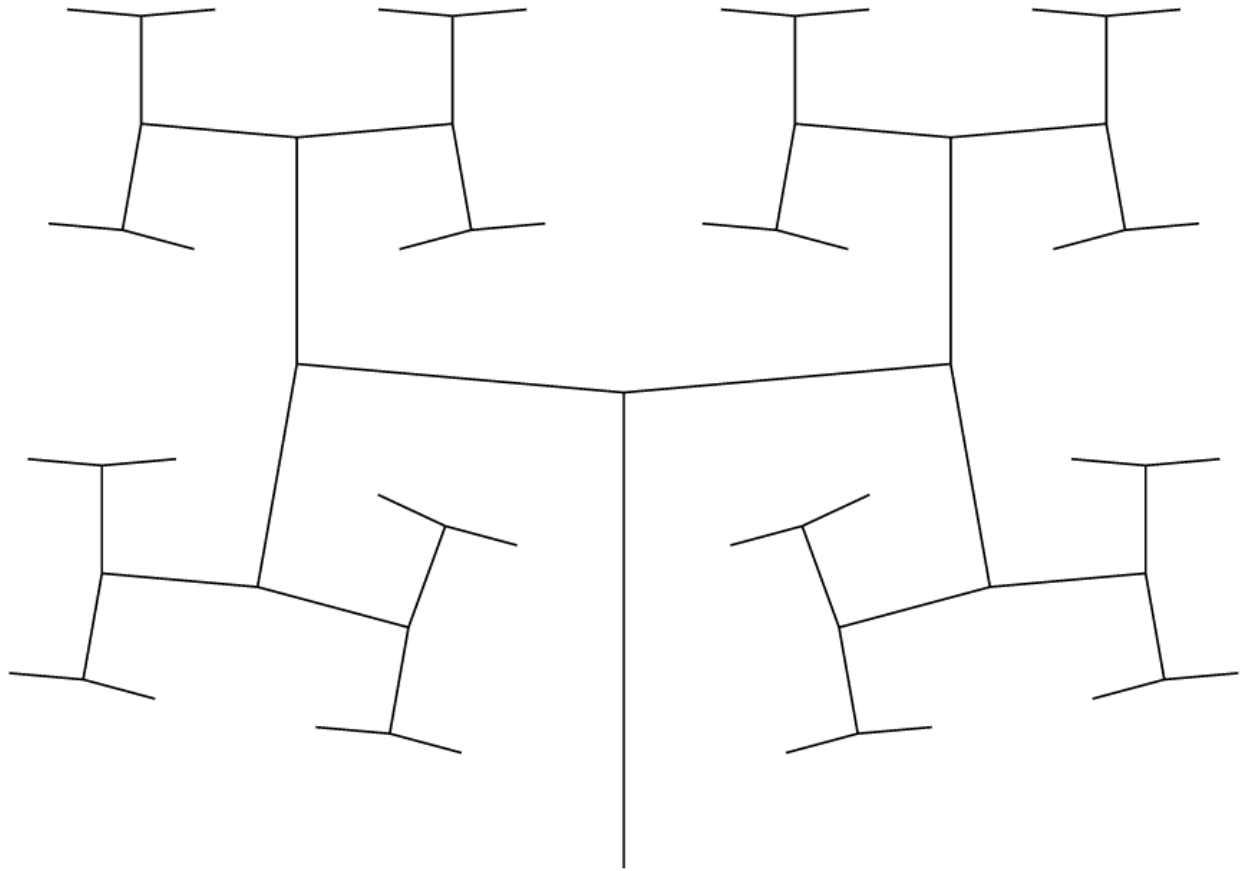
It looks almost as the image that was promised. You can decide how many times the rules are applied by the number in *Generations number* form. If you change it to 0 you should get old green line/box. When it is set to 5, it means the rules were applied 5 times. To get the image from beginning of this tutorial set the number to 7.

You can also try changing the number in *Percent randomization* form. If you set it to 20 it basically means that every time some rotation or line is found the angle or length is randomized by 20%.

* Note that if you change the number of generations to something large, the image becomes too complex to be drawn on the screen. This complexity is often not necessary and can be prevented by using parameters, which you will learn in next tutorial. *

Parameters

In this part of the tutorial, we will cover parameters. This part assumes you have learned *The Basics*. The goal of this part will be to create H-Tree:



First let's create modules for rotation and saving/loading position. This process is same as the one covered in previous part, so we will not describe it into detail. Only change is that instead of 45 degrees this time we will rotate by 85 degrees. You should get modules equivalent to those on following image:

Modules

Define modules which will be used in LSystem

Name	Parameters	Representation	
+		Rotation by angle: 85	×
-		Rotation by angle: -85	×
[Saves current position to stack	×
]		Loads last saved position from stack	×

Now the new stuff. Create new module and name it *H*. Now click on *Add parameter*. In the newly created form fill in the name of the parameter. Let's call it *x*. Now change the representation to *Line*. This time instead of filling in number in *Length* form, fill in *x*. Now every time this module will be drawn length will be determined by parameter associated with the module.

Let's create another module *F*. Add parameter to it and name it *y*. Change representation to *Line* and fill in *y* as length.

Axiom

In the axiom screen you can see that modules H and F have form associated with them. Value in this form will be used as initial value for the module. Now drag one F module into the blue box and fill in 200 as initial value for parameter y . Save it.

Axiom

Select modules forming axiom by dragging them from the blue box into the green box

H

x

F

y

F

y 200

Rules

To generate H-Tree you need only one rule. As source module choose F . Now drag modules into the blue box to correspond with following image:

New rule

Source module:

F

Next generation:

H

x

F

y

H

x y

F

y y*0.69

[

+

F

y y*0.69

]

[

-

H

x y

F

y y*0.69

[

+

F

y y*0.69

]

[

-

Show advanced options

Add rule

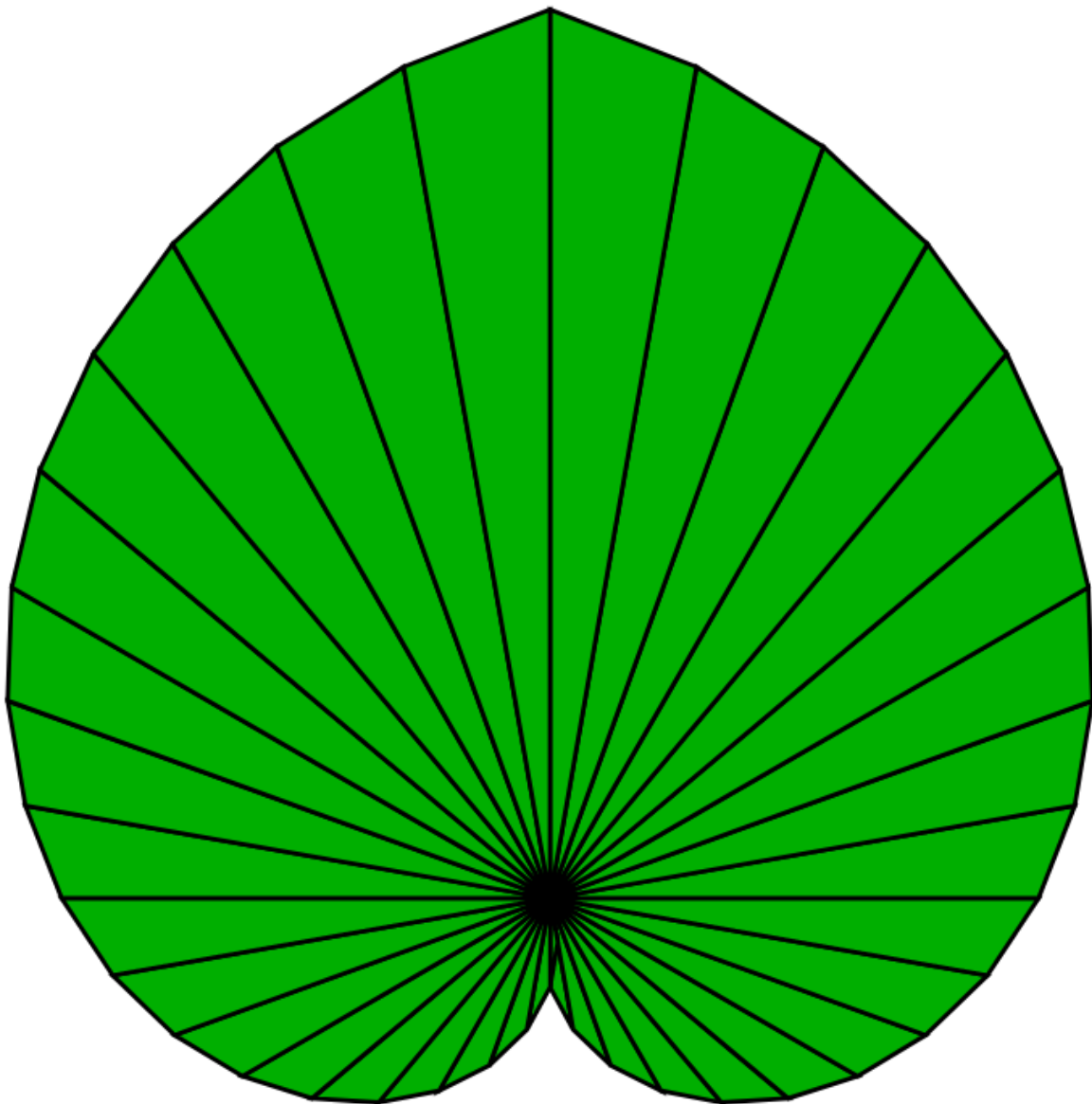
You can see that modules here also have forms associated with them. You can fill in number or mathematical expression. As you can see in the image you can also use variable names in the expressions. Note that you can only use variables which are associated with source module.

In this example, the rule on the image will transform one F module which has some value set in parameter y into one module H and set value of parameter x to the value of parameter y . Then it will create two F modules and set their parameters y to 0.69 times value of y associated with source module.

Now save the changes and move to the result page. You should see image corresponding to first image on this page.

Polygons

Up to this point we were drawing only lines. In this part we will learn how to draw polygons. In the end we will create image of leaf:



To create polygons you need at least three modules. One module will describe the colors of polygon, one will add points to polygon and the last will draw the polygon on the screen.

But first let's create other modules we will need. We will need modules for saving and loading positions, and modules for rotation by 10 degrees in both directions. Then we will need two modules without representation, let's call them *Left* and *Right*. These will denote if we are drawing left or right part of leaf. The last non polygon module will be moving forward based on its only parameter. We will call it *F* and its only parameter will be called *x*. You should end with something like this:

Modules

Define modules which will be used in LSystem

Name	Parameters	Representation	
[Saves current position to stack	×
]		Loads last saved position from stack	×
+		Rotation by angle: 10	×
-		Rotation by angle: -10	×
Left		None	×
Right		None	×
F	x	Move forward by: x	×

Now we need modules denoting polygons. Let's create module called *Start polygon* and change representation to *Polygon*. Leave *Polygon action* as *Create* and change *Fill color* to R: 0, G: 175, B: 0. That's all needed for starting polygon.

Now create module with name *Add point* and change its representation to *Polygon*. Change *Polygon action* to *Add current point to polygon*.

Last module related to polygons will be named *Draw polygon*. Change its representation to *Polygon* and *Polygon action* to *Close polygon*. You should end up with these modules:

Modules

Define modules which will be used in LSystem

Name	Parameters	Representation	
[Saves current position to stack	×
]		Loads last saved position from stack	×
+		Rotation by angle: 10	×
-		Rotation by angle: -10	×
Left		None	×
Right		None	×
F	x	Move forward by: x	×
Create polygon		Creates new polygon with stroke color R: 0 G: 0 B: 0 and fill color R: 0 G: 175 B: 0	×
Add point		Adds current point to currently created polygon	×
Draw polygon		Closes current polygon and draws it	×

Set your axiom to be same as on the following image:

Axiom

Select modules forming axiom by dragging them from the blue box into the green box

[] + - Left Right F x Create polygon Add point Draw polygon

[Left] [Right]

✕

Now we need to create rules to draw left and right side of the leaf. Let's start with Left. Select *Left* as source module. Than drag modules into the blue box to correspond to following image:

New rule

Source module:

Left

Next generation:

[] + - Left Right F x

Create polygon Add point

Draw polygon

[+ Left Create polygon Add point] Add point F x 10

Add point Draw polygon

We will do the same for the right side, just mirror the rotation.

Last rule we need to change is to make moves larger for older modules. Let's create rule with source module *F* and next generation that looks like this:

New rule

Source module:

F

Next generation:

[] + - Left Right F x

Create polygon Add point

Draw polygon

F x x+10

Now it is time to look at the result. Change *Generations number* to 20 and your leaf is done.