

# Generative Art with Scribble

A Project Based Introduction to Programming



Aidan Lane, Bernd Meyer and Jonathan Mullins  
With contributions by Jon McCormack



MONASH University

# Generative Art with Scribble

Aidan Lane, Bernd Meyer, Jonathan Mullins

## **Generative Art with Scribble**

Aidan Lane, Bernd Meyer, Jonathan Mullins

License: CC Attribution-ShareAlike 3.0 Unported (CC BY-SA 3.0)

Generated by [Alexandria](https://www.alexandriarepository.org) (https://www.alexandriarepository.org) on November 26, 2018 at 1:33 pm AEDT

# Contents

Title .....	i
Copyright .....	ii
<b>1 Introduction</b> .....	<b>1</b>
<b>1.1 Generative Art</b> .....	<b>2</b>
<b>1.2 Scribble</b> .....	<b>6</b>
<b>2 Tutorials</b> .....	<b>9</b>
<b>2.1 Drawing Shapes with the Pen</b> .....	<b>10</b>
<b>2.2 Motion, Repetition and Pen Colour</b> .....	<b>14</b>
<b>2.3 Shapes and Fill Colour</b> .....	<b>18</b>
<b>2.4 Transparency and Fill Colour</b> .....	<b>22</b>
<b>2.5 Variables</b> .....	<b>26</b>
<b>2.6 Random Numbers</b> .....	<b>29</b>
<b>2.7 Conditionals</b> .....	<b>33</b>
<b>2.8 Drawing Functions</b> .....	<b>37</b>
<b>2.9 Custom Blocks</b> .....	<b>41</b>
<b>2.10 Parameters</b> .....	<b>46</b>
<b>2.11 Recursion</b> .....	<b>49</b>

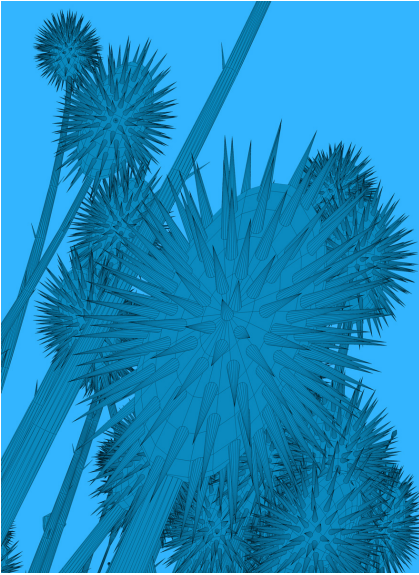


# **1 Introduction**

[1.1 Generative Art](#)

[1.2 Scribble](#)

---



© 2012 Jon McCormack, All rights reserved.

## 1.1 Generative Art

### What is Generative Art?

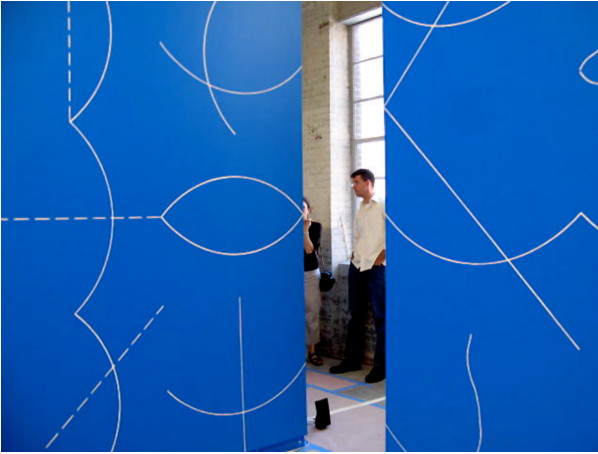
*A brief overview*

Generative art is art made by a process or machine developed by the artist.

We typically think of artists and designers working directly on the artworks they create. A painter paints onto a canvas; a potter moulds clay with her hands as it rotates on a spinning wheel; a photographer frames a composition, deciding the exact moment when to press the shutter release and take the photograph; a composer writes musical notes in sequence on paper. Generative art is different. It is a type of art where the artist first creates a process, and then this process is used to generate the artwork. Generative art is not specific to any medium and has been practiced for centuries in various guises.

So what does it mean to "create a process"? A process might be a set of rules or series of instructions that are carried out by others. Each rule or instruction may be written in everyday language but is generally straightforward and unambiguous.

For example, the American artist Sol Le Witt (1928 - 2007) devised a series of wall drawings that could be made by anyone following the artist's instructions. The instructions included things like "All combinations of two lines crossing, placed at random, using arcs from corners and sides, straight, not straight and broken lines." While each person's exact interpretation of the instructions will be different, the overall features of the drawings produced by different people are remarkably similar.



"Do-it-yourself Sol LeWitt Wall Drawings. It's easy!" -  
Hello Beautiful! © 2008 [Edward Lifson](#)  
(<https://hellobeautifulblog.wordpress.com/2008/06/25/do-it-yourself-sol-lewitt-wall-drawings-its-easy/>), All Rights Reserved. Courtesy of the artist.

*License: Other (see caption)*

The English composer, Cornelius Cardew (1936 - 1981) created a musical work that consisted of a series of written instructions to singers, rather than conventional sheet music with notes. The instructions are the same for each singer and include choosing your own first note (silently) at random and singing a line from a list of short phrases "for the length of a breath." Subsequent notes are chosen by selecting a note heard that is being sung by a nearby colleague. When performed, the composition starts out somewhat chaotically but slowly converges as singers pick the notes they hear being sung by others around them. As with Le Witt's drawings, each performance of the composition is different, yet the overall features make it clearly recognisable as Cardew's composition.

The advent of digital computers brought many new possibilities for generative art. This was because programming a computer is another way to create a process. Rather than having people or mechanical devices carry out the artist's instructions, the computer interprets program instructions to generate the art. Of course, computers have a lot of advantages over people for interpreting processes: they carry out the instructions exactly; they work quickly; they don't get bored or require breaks for meals, and so on. But computers have no innate intelligence or bodies, so they can only interpret instructions that can be implemented as computer code.

One of the first artists to explore generative computer art was the German artist, Manfred Mohr (1938-). He held an exhibition in 1971 in Paris of 28 computer-generated drawings. An example is shown in the figure.

As each of the 28 drawings was generated by a different computer program (or "algorithm"), each was quite different. The computer carried out the program's instructions, controlling a mechanical pen that could be moved anywhere over the surface of the canvas.

As computers became more powerful, more complex processes could be implemented. Output technology also developed, permitting high resolution colour images to be generated. The image on the following page by Jon McCormack was created using a special type of generative system known as "L-systems". These virtual plants were grown in the computer, and a process similar to biological evolution was used to evolve and change the form and appearance of each plant. A handful of relatively simple rules are all that is needed to generate each virtual plant.



Generative art can also be interactive, allowing a process to change its behaviour according to the world around it. We might traditionally think of purposeful interaction with a computer using a touch interface or a mouse, but many different sensors can also be connected to a computer, allowing it to respond to heat, light or movement for example. A built-in digital camera allows the computer to "see" the world around it and respond to what it sees. For example, a musical composition could change according to the colours seen by the camera. Imagine hearing a composition that changes depending on the colour of the clothes you are wearing each day!



P-021, "band-structure", ink/paper.

© 1970 [Manfred Mohr](http://www.emohr.com/) (<http://www.emohr.com/>), All Rights Reserved. Courtesy of the artist.

While generative art has changed with technology, all generative art (and what are known as "generative systems") share a number of interesting properties. One of these properties is that the specification of the process is far simpler than the outcomes generated. In McCormack's Morphogenesis Series for example, the instructions to build each plant can fit on a single A4 page, yet the 3D geometric model the instructions generate is millions of times more complex. This type of phenomena is also found in the natural world: a person's DNA sequence will fit on a standard CD. DNA codes for the sequences of amino acids which build proteins, and progressively cells, organs, and the individual - each significantly more complex than the original instructions encoded in the DNA from which they were specified.

Another interesting property of some generative art is that, in carrying out the process or procedure, new and interesting things emerge in the artwork produced. These things - patterns, forms and behaviours - are not explicitly programmed by the artist. They emerge through the complex interaction of elements generated by the process. Even the artist may not understand exactly how they come about because the process is too complex to comprehend directly. This also implies that some behaviour cannot be predicted in advance. We need to run the program many times and examine the results it produces to understand it.

Generative art also raises interesting questions about art and creativity. In Sol Le Witt's drawings for example, who is the artist? Is it Le Witt himself, who devised the instructions for making the drawings (but did not create them), or the person interpreting his instructions to actually draw the drawing?

When a computer is interpreting the artist's instructions we might be less inclined to think of it as contributing any creativity to the artwork. But as computers become more sophisticated this view might change, particularly as artists start to use techniques from artificial intelligence research in their works. Eventually, computers might be able to create their own art, independently of people.

Even with today's technology, generative art is interesting because it harnesses the generative power of the computer and can often lead to artworks that surprise even the artist who designed the generative art program.

## Interactive: Branches

[Branches](http://labs.christianwannerstedt.com/#branches/) (<http://labs.christianwannerstedt.com/#branches/>) is an "engine" for generating different types of visual patterns, using branching and recursion techniques.

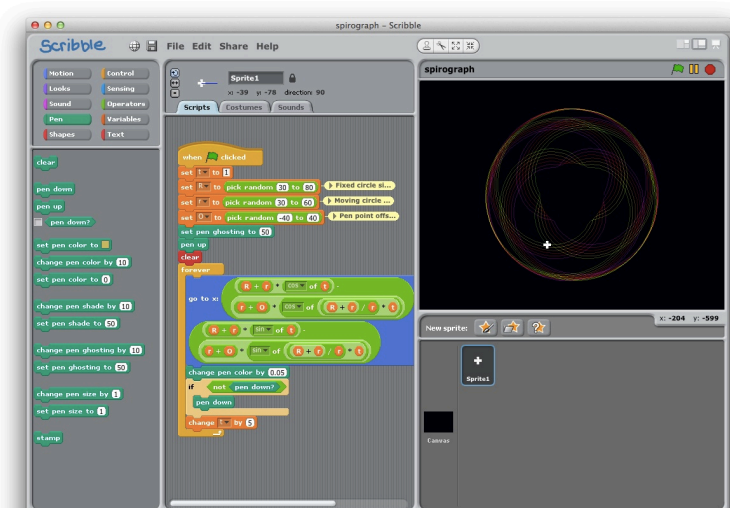
It has several presets, allowing for the immediate generation of flower, coral and tree like patterns. And for the more adventurous, there is large selection of sliders (under "settings"), which allows for a enormous array of new creations.

Tap anywhere in the white area of the window below to begin!



 Click to open

Christian Wannerstedt (used with permission)



## 1.2 Scribble

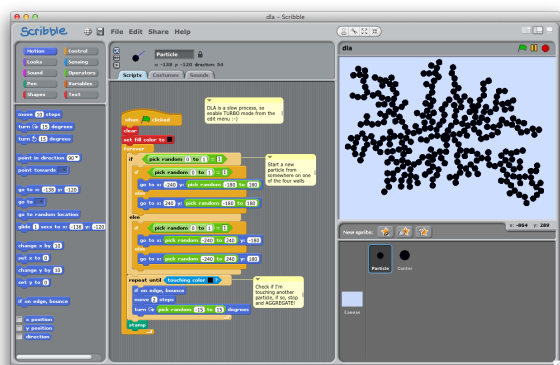
[Scribble](http://flipt.org/#scribble) (<http://flipt.org/#scribble>) is a fun way for people of all ages to learn about generative art, using a simple drag-and-drop programming interface. Scribble programs can be written by anyone, even without prior coding experience.

Along with [Enchanting](http://flipt.org/#enchancing) (<http://flipt.org/#enchancing>) and [Cellular](http://flipt.org/#cellular) (<http://flipt.org/#cellular>), Scribble is an open-source extension to the fantastic [BYOB](http://byob.berkeley.edu/) (<http://byob.berkeley.edu/>) (Build Your Own Blocks) platform, which is developed by the [University of California, Berkeley](http://berkeley.edu/index.html) (<http://berkeley.edu/index.html>). This extension in turn is developed by the [Monash University Faculty of Information Technology](http://www.infotech.monash.edu.au/) (<http://www.infotech.monash.edu.au/>). For more information, please see SNAP! - The Future of BYOB.

To get started, simply click on the HTML 5 web app [here](http://snapapps.s3-website-ap-southeast-2.amazonaws.com/scribble.html) (<http://snapapps.s3-website-ap-southeast-2.amazonaws.com/scribble.html>). (Firefox or Chrome work better than Safari for OSX users; if viewing this page in another browser, you can copy the link here and paste it in Firefox or Chrome).

Also, Scribble is available as desktop application. Download Scribble for Mac OS X and Windows from the link below and check out the example files included with the program.

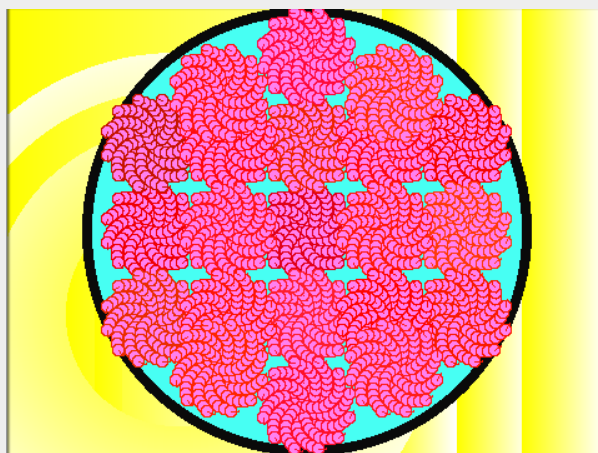
<http://flipt.org>



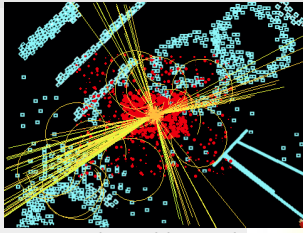
## Student Gallery

The following student gallery contains a selection of artwork from [Generative Art 2012](https://sites.google.com/a/jmss.vic.edu.au/generative-arts-virtual-fair/) (<https://sites.google.com/a/jmss.vic.edu.au/generative-arts-virtual-fair/>). These pieces of artwork serve as inspiration of what you can archive with Scribble!

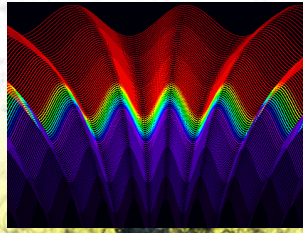
"Generative Art 2012 is a virtual exhibition of student art created at John Monash Science School as part of the subject Emerging Technologies. Its visual and audiovisual pieces are inspired by the sciences, mathematics and humanities".



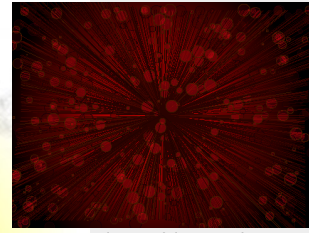
*The Aesthetics of Muscle Fibre*  
© 2012 Raghav Rana, All Rights Reserved.



(<https://www.alexandriarepository.org/media/rah0001-farhanrahman-higgsboson>)  
*Farhanrahman, Higgs  
Boson*



(<https://www.alexandriarepository.org/media/sir0004-electromagnetic-radiation>)



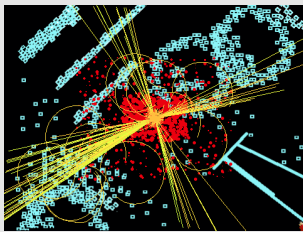
(<https://www.alexandriarepository.org/media/ton0003-conflict>)



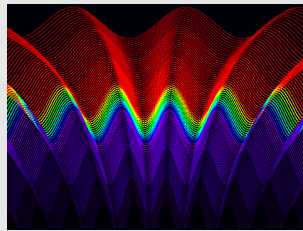
[Growth](https://sites.google.com/a/jmss.vic.edu.au/generative-arts-virtual-fair/10tent1e/tru0009)

(<https://sites.google.com/a/jmss.vic.edu.au/generative-arts-virtual-fair/10tent1e/tru0009>) © 2012 Mel Truong, All Rights Reserved.

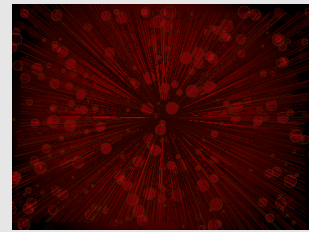
Here is even more Scribble student art:



(<https://www.alexandriarepository.org/media/rah0001-farhanrahman-higgsboson>)  
*Farhanrahman, Higgs  
Boson*



(<https://www.alexandriarepository.org/media/sir0004-electromagnetic-radiation>)



(<https://www.alexandriarepository.org/media/ton0003-conflict>)

# 2 Tutorials

[2.1 Drawing Shapes with the Pen](#)

[2.2 Motion, Repetition and Pen Colour](#)

[2.3 Shapes and Fill Colour](#)

[2.4 Transparency and Fill Colour](#)

[2.5 Variables](#)

[2.6 Random Numbers](#)

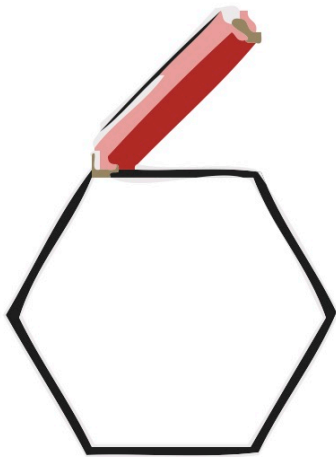
[2.7 Conditionals](#)

[2.8 Drawing Functions](#)

[2.9 Custom Blocks](#)

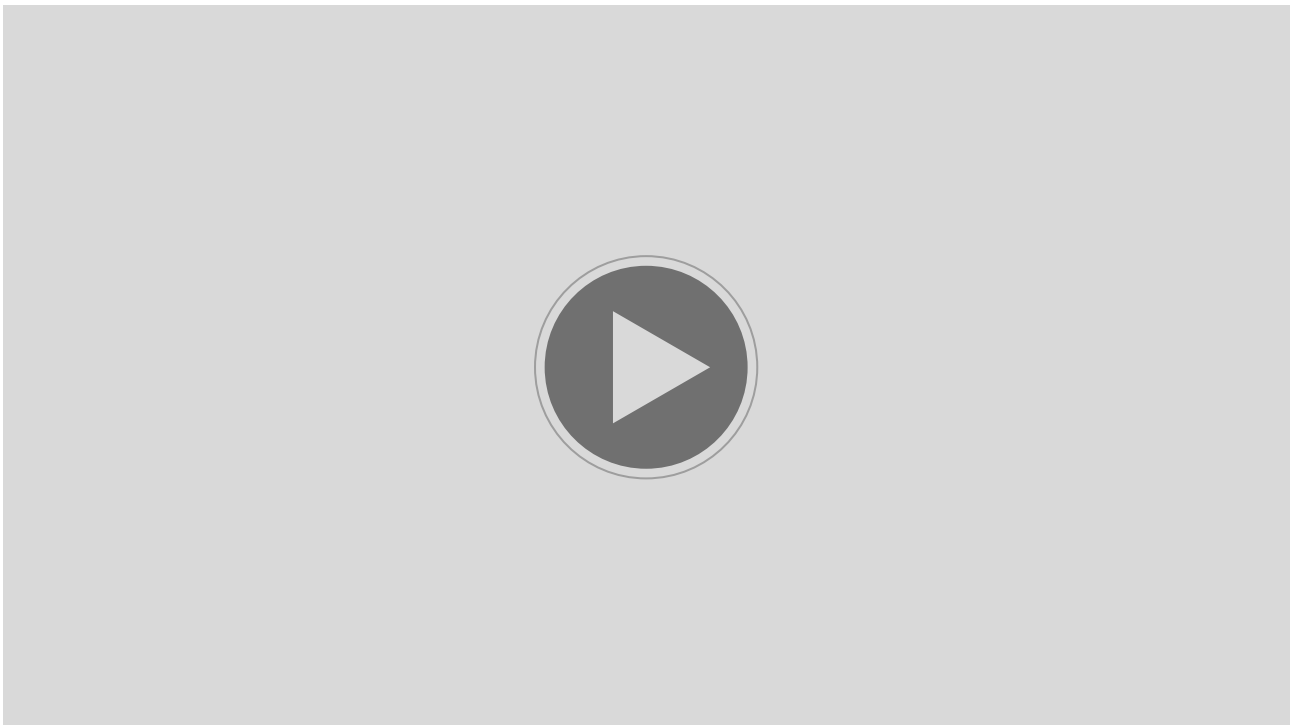
[2.10 Parameters](#)

[2.11 Recursion](#)



## 2.1 Drawing Shapes with the Pen


In this tutorial you will learn how to use Scribble's pen tool to draw simple shapes. Watch the short tutorial below and then try the questions and activities that follow.





(<https://www.alexandriarepository.org/wp-content/uploads/20150728090108/tutorial1-BlockBooks-Pass1-BlockBooks-Pass2.mp4.mp4>)  
License: CC Attribution-ShareAlike 3.0 Unported (CC BY-SA 3.0)

### Questions

For the following quiz, refer to the Scribble project in the below window. Use the Run button to view the demonstration, and the Show Blocks button to see how it works.




**Legend:**  Run.




*License: CC  
Attribution-  
ShareAlike  
3.0 Unported  
(CC BY-SA  
3.0)*

Pause.




*License: CC  
Attribution-  
ShareAlike  
3.0 Unported  
(CC BY-SA  
3.0)*

Stop.



*License: CC  
Attribution-  
ShareAlike  
3.0 Unported  
(CC BY-SA  
3.0)*

Show blocks.



*License: CC  
Attribution-  
ShareAlike  
3.0 Unported  
(CC BY-SA  
3.0)*

Expand stage.

(If this project does not open, see notes at bottom of page).

Run the script by pressing the green flag in top right-hand corner of the screen.

Note that you can pick up the pencil with the mouse and move it around the canvas. The hexagon will be drawn wherever you place it.

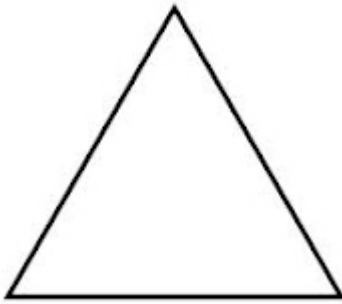
**Start Quiz** (<https://www.alexandriarepository.org/app/WpProQuiz/1>)

## Activities

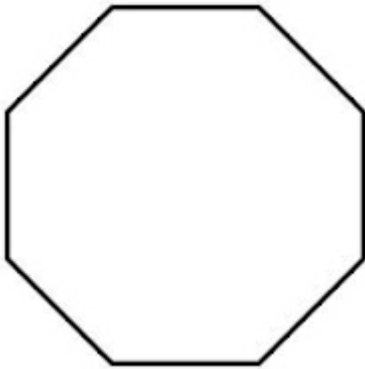
Modify the tutorial script to draw the following shapes. (Click the double arrow icon to edit the script, then click the icon again the explode the canvas).



For example:



Now try these:



License: CC Attribution-ShareAlike  
3.0 Unported (CC BY-SA 3.0)

**HINT** - A true circle has no corners, but you can create the illusion of a circle using many small corners (maybe 360 of them?)

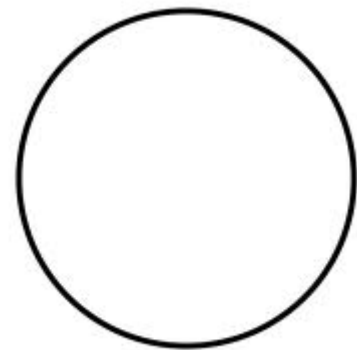
## Extension Exercise

All the previous shapes have one thing in common, they are **equiangular**, meaning that the angle of every corner is the same (a true circle doesn't have any corners, but the one you've drawn on your canvas has many small ones).

Think of a shape that has is not equiangular, and write a script to draw it. Two examples are an **isosceles triangle** and a **parallelogram**.

**HINT** - You won't be able to simply repeat the same two blocks over and over, but depending on the shape you choose, you might be able to repeat some of them!

What shape did you choose?



License: CC Attribution-ShareAlike 3.0  
Unported (CC BY-SA 3.0)

## Notes

For the Scribble HTML 5 web application, Firefox works better than Safari for OSX users.

If the link under Questions does not open, use the desktop Scribble application, and go to the following file.

(File → Open... →) Examples → tutorials → Tutorial1

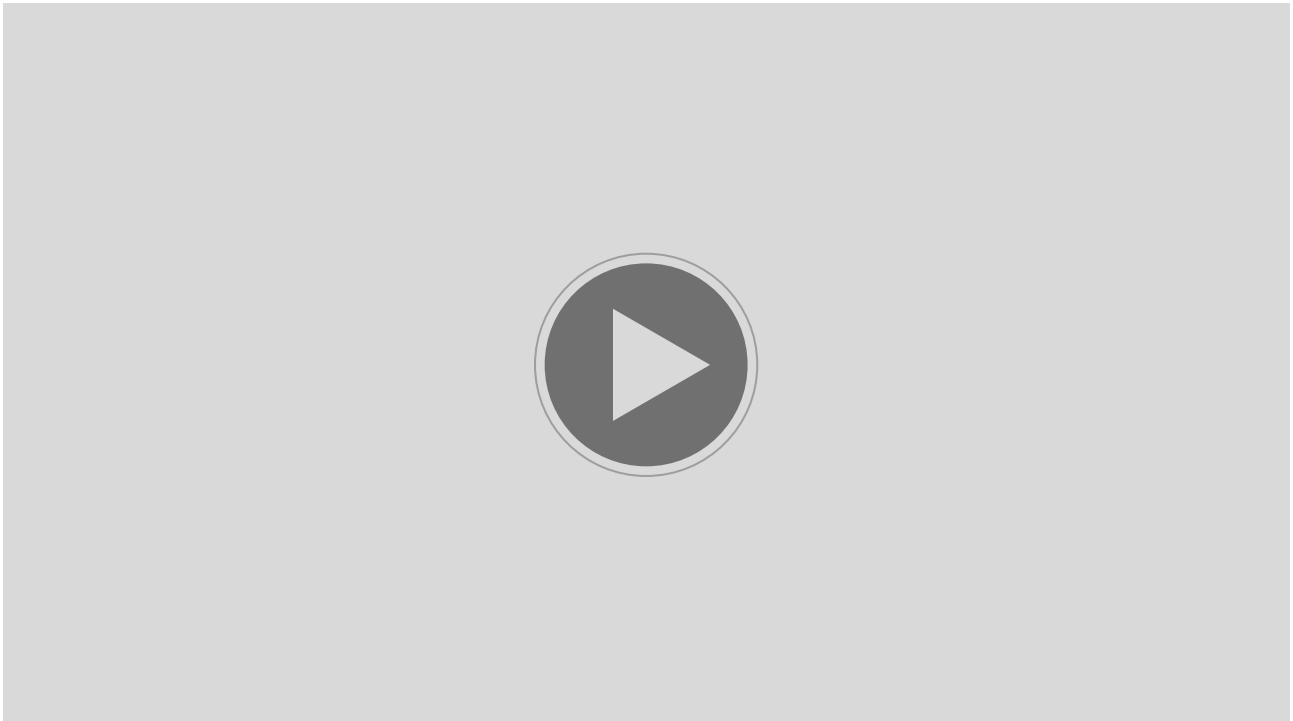
---



## 2.2 Motion, Repetition and Pen Colour

In this tutorial you will learn how to use combinations of loops and motion to draw shapes repetitively across the canvas. Watch the short tutorial below and then try the questions and activities that follow.

### Tutorial Motion, Repetition and Pen Colour





(<https://www.alexandriarepository.org/wp-content/uploads/20150728090059/tutorial2-BlockBooks-Pass1-BlockBooks-Pass2.mp4.mp4>)

### Questions


For the following quiz, refer to the Scribble project in the below window:




Legend:  Run.

  
 License: CC  
 Attribution-  
 ShareAlike  
 3.0 Unported  
 (CC BY-SA  
 3.0)


Pause.

  
 License: CC  
 Attribution-  
 ShareAlike  
 3.0 Unported  
 (CC BY-SA  
 3.0)

Stop.

  
 License: CC  
 Attribution-  
 ShareAlike  
 3.0 Unported  
 (CC BY-SA  
 3.0)

Show blocks.

  
 License: CC  
 Attribution-  
 ShareAlike  
 3.0 Unported  
 (CC BY-SA  
 3.0)

Expand stage.

(If this project does not open, see notes at bottom of page).

Run the script by pressing the green flag in top right-hand corner of the screen.

**Start Quiz** (<https://www.alexandriarepository.org/app/WpProQuiz/45>)

## Activities

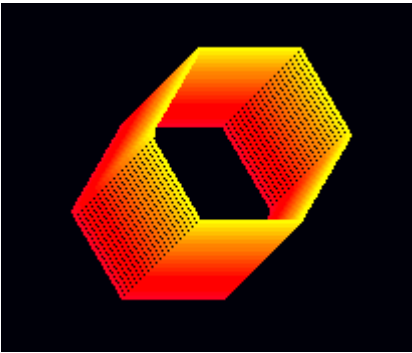
Modify the script to draw the following images:



License: CC Attribution-ShareAlike  
3.0 Unported (CC BY-SA 3.0)

### HINTS:

- Use another **repeat** \_\_ block to draw multiple hexagons
- Use the **set pen colour to** \_\_ block at the start of the script



### HINTS:

- To change the canvas colour, double click anywhere on the canvas and select the **backgrounds** tab. Edit the background and use the fill tool. Double click the pencil again when you are done.
- Use the **change pen colour by** \_\_ to change the pen colour after each hexagon is drawn.

## Extension Exercise

Time to get creative! Come up with a design of your own, by making changes to the shape, the motion and the pen colour. Try to visualise what you want to construct first, and then attempt to write the script. Discuss the aspects of your design, making sure to reference:

- Shape
- Motion / position
- Colour

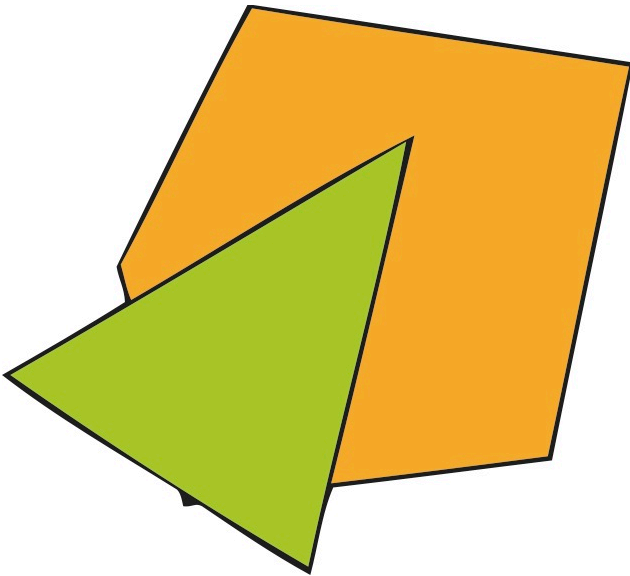
## Notes

For the Scribble HTML 5 web application, Firefox works better than Safari for OSX users.

If the link under Questions does not open, use the desktop Scribble application, and go to the following file.

(File → Open... →) Examples → tutorials → Tutorial2





## 2.3 Shapes and Fill Colour

In this tutorial, you will learn about the process of creating shapes on the canvas using the predefined shape types, including how to set the border and fill colours. Watch the short tutorial below and then try the questions and activities that follow.

### Tutorial Shapes and Fill Colour



(<https://www.alexandriarepository.org/wp-content/uploads/20150728090127/tutorial3-BlockBooks-Pass1-BlockBooks-Pass2.mp4.mp4>)  
License: CC Attribution-ShareAlike 3.0 Unported (CC BY-SA 3.0)

## Questions

**Start Quiz** (<https://www.alexandriarepository.org/app/WpProQuiz/29>)

## Activities

For each of the flags pictured below, write a script that draws it on the canvas. You can either make the flag fill the entire canvas, or you can make the flag smaller and draw a flagpole attached to it.



A: Australian Aboriginal



B: Jamaica

License: CC Attribution-ShareAlike 3.0  
Unported (CC BY-SA 3.0)



C: Vietnam

License: CC Attribution-ShareAlike 3.0  
Unported (CC BY-SA 3.0)

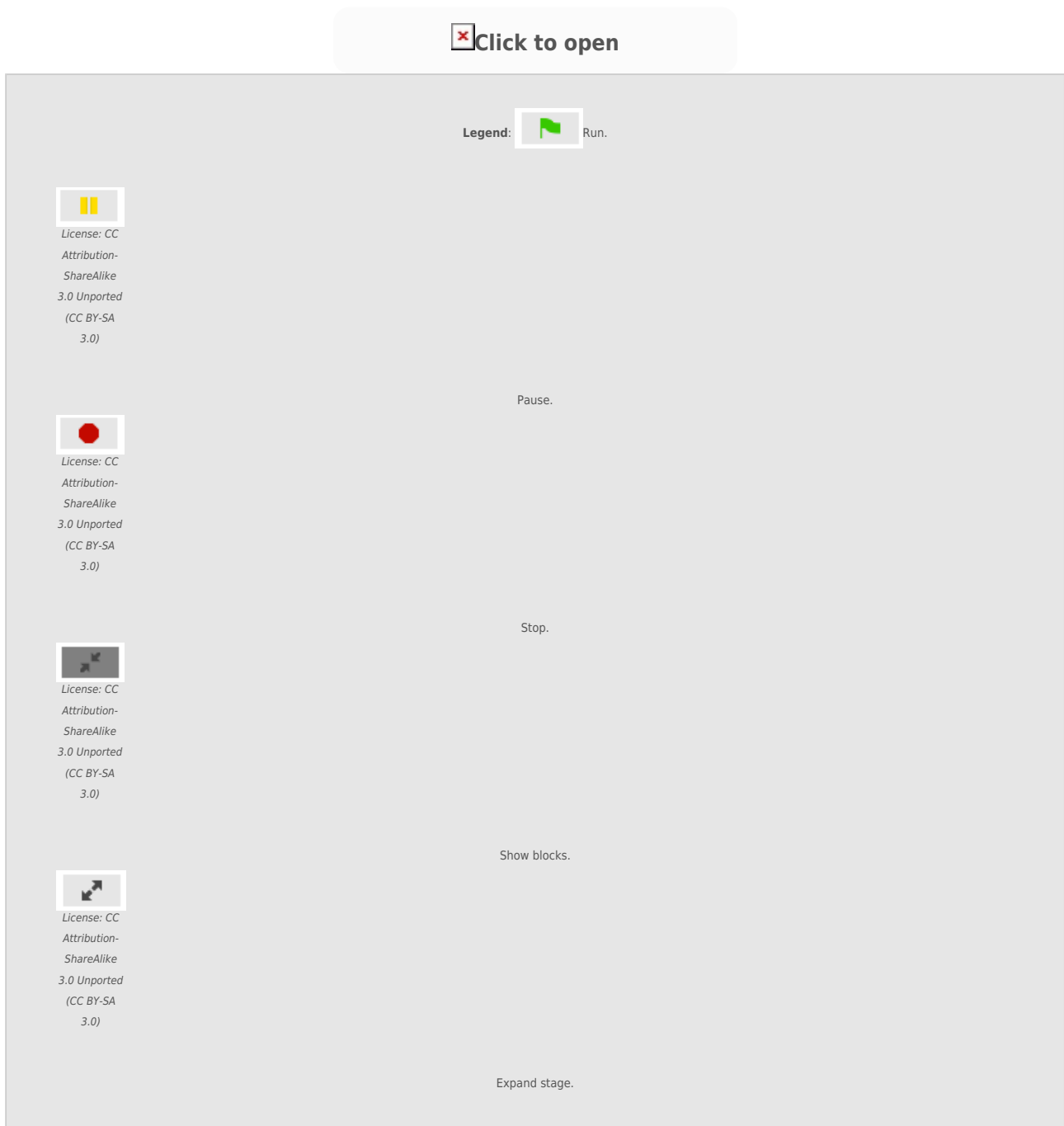
### HINTS:

- Identify the different shapes that need to be drawn, then construct each shape using three steps:
  1. Position the pencil (using x and y coordinates)
  2. Set the fill colour
  3. Draw the shape
- To hide the pencil before saving a picture of your canvas, double click the **hide** block in the **looks** category.



Double click the **show** block to restore the pencil.

- Setting the pen size to zero will prevent a border from appearing around shapes
- An intelligent selection of canvas background colour will make your job easier!

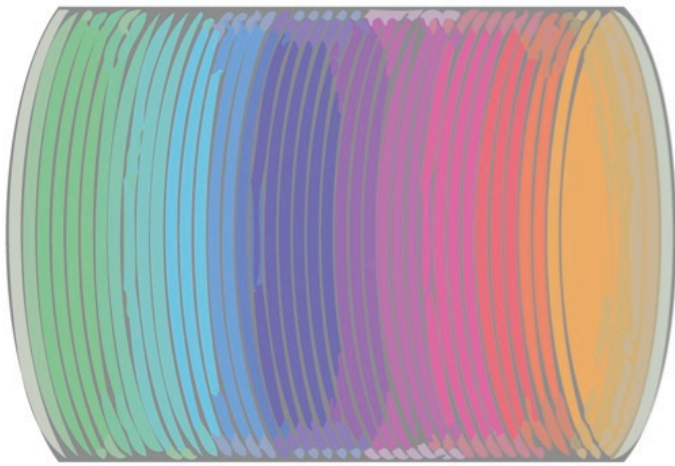


(If this project does not open, see notes at bottom of page).

## Note

If you have problems with Scribble, try viewing this page in Firefox.

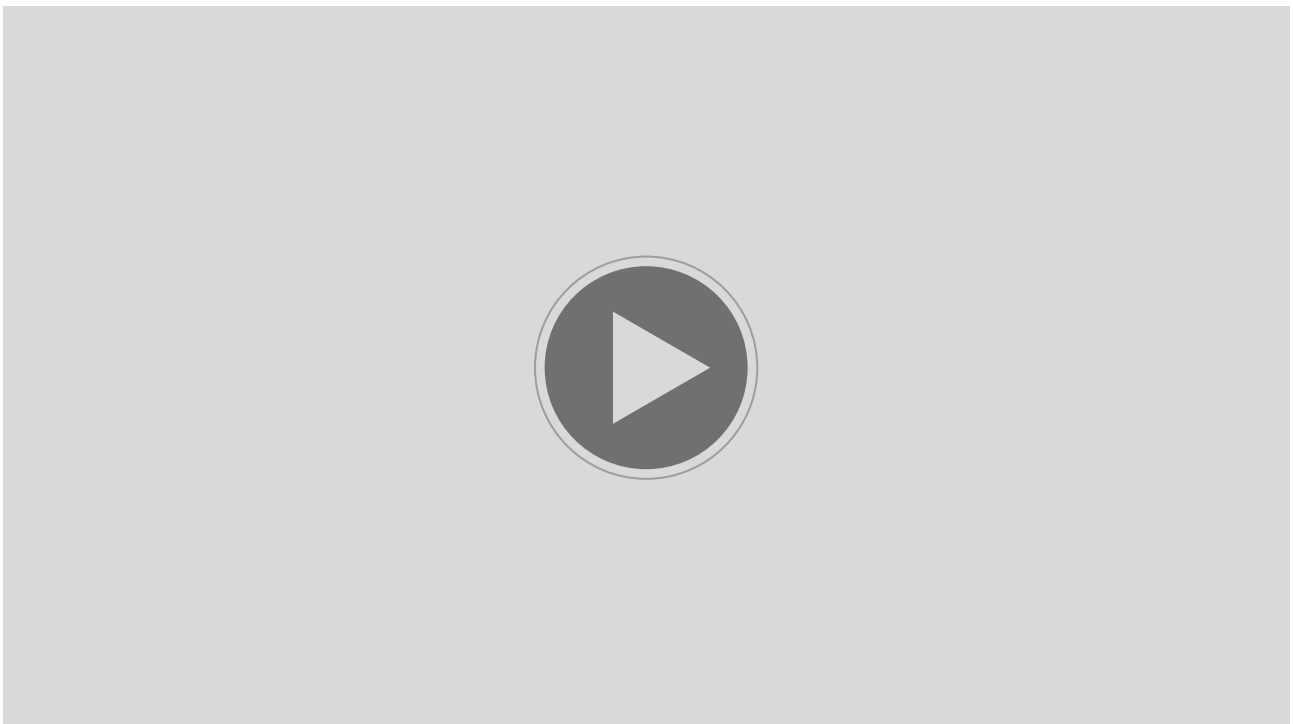




## 2.4 Transparency and Fill Colour

In this tutorial you will learn how about the use of pen and fill colours, as well as using transparency to create the illusion of depth. Watch the short tutorial below and then try the questions and activities that follow.

### Tutorial Transparency and Fill Colour



(<https://www.alexandriarepository.org/wp-content/uploads/20150802063610/tutorial4-BlockBooks-Pass1-BlockBooks-Pass2.mp4.mp4>)  
License: CC Attribution-ShareAlike 3.0 Unported (CC BY-SA 3.0)

## Questions

**Start Quiz** (<https://www.alexandriarepository.org/app/WpProQuiz/30>)


## Activities


For each of the following descriptions, create a script that produces your interpretation of it:


1. A row of red circles that overlap, each circle completely obscuring the circle behind it.
2. A column of blue squares that overlap, each square partially obscuring the square behind it.
3. A row of triangles that overlap, each triangle partially obscuring the triangle behind it.
4. The colour of the triangles should change from red through to yellow.

 Click to open


**Legend:**

  
Run.

  
*License: CC Attribution-ShareAlike 3.0 Unported (CC BY-SA 3.0)*  
Pause.

  
*License: CC Attribution-ShareAlike 3.0 Unported (CC BY-SA 3.0)*  
Stop.

  
*License: CC Attribution-ShareAlike 3.0 Unported (CC BY-SA 3.0)*  
Show blocks.

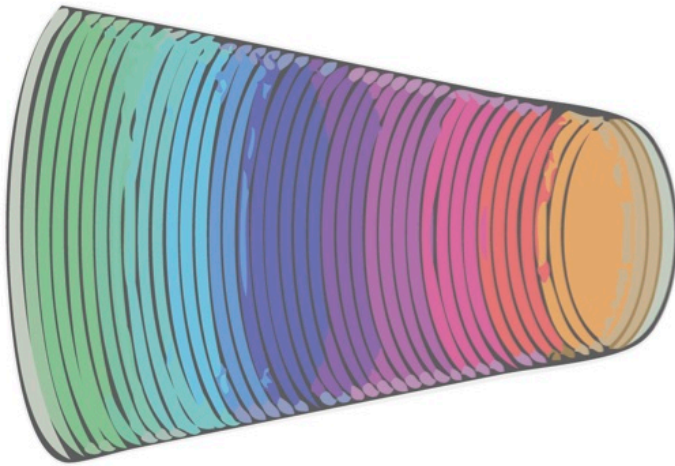
  
*License: CC Attribution-ShareAlike 3.0 Unported (CC BY-SA 3.0)*  
Expand stage.

(If this project does not open, see notes at bottom of page).

## Note

If you have problems with Scribble, try viewing this page in Firefox.

---



## 2.5 Variables

In this tutorial you will learn what variables are and how they can be used to modify block parameters dynamically as the script is executed. Watch the short tutorial below and then try the questions and activities that follow.

### Tutorial 2 Variables



(<https://www.alexandriarepository.org/wp-content/uploads/20150802063711/tutorial5-BlockBooks-Pass1-BlockBooks-Pass2.mp4.mp4>)  
License: CC Attribution-ShareAlike 3.0 Unported (CC BY-SA 3.0)

## Questions

**Start Quiz** (<https://www.alexandriarepository.org/app/WpProQuiz/31>)

## Activities

In the tutorial video at the start of this section, a slinky is transformed from a **cylindrical** shape into a **conical** shape using a variable to adjust the oval width.



All Gizah Pyramids © 2006 Ricardo Liberato, CC BY-SA 2.0

Your task is to create an artwork featuring **pyramids**. Once you have developed a script that draws a pyramid, experiment with fill colour, transparency, and even changing the canvas background.

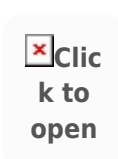
### HINTS:

- Think of what changes you could make to the conical slinky example to make a pyramid
- The name of a variable is irrelevant, it's what you do with the variable that matters
- In the following script fragment, the loop will repeat **until** the variable called **width** is less than 1... this might be more useful than a specified number of repeats?



- The repeat until loop can be found further down in the control category

Use the Scribble window below to generate your pyramid.





**Legend:**

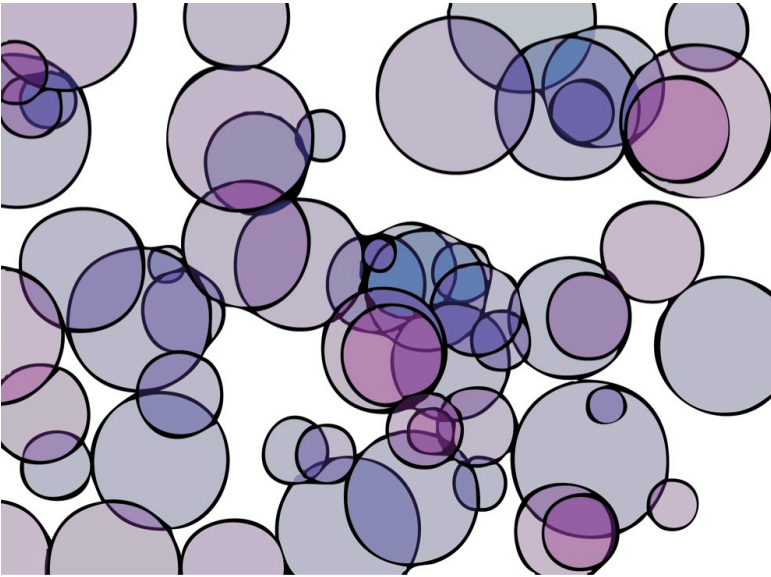
  
Run.

  
License: CC Attribution-ShareAlike 3.0 Unported (CC BY-SA 3.0)  
Pause.

  
License: CC Attribution-ShareAlike 3.0 Unported (CC BY-SA 3.0)  
Stop.

  
License: CC Attribution-ShareAlike 3.0 Unported (CC BY-SA 3.0)  
Show blocks.

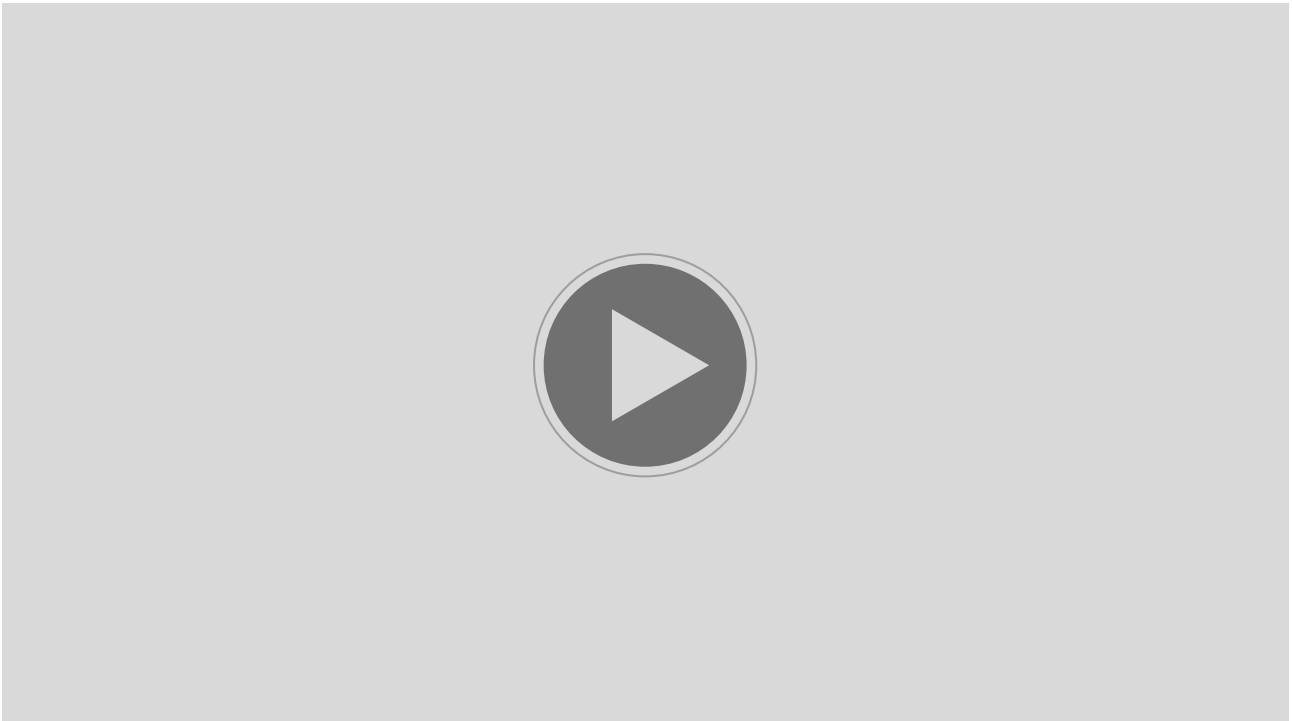
  
License: CC Attribution-ShareAlike 3.0 Unported (CC BY-SA 3.0)  
Expand stage.



## 2.6 Random Numbers

In this tutorial you will learn how to use randomness in your scripts for creating unpredictable and interesting images. Watch the short tutorial below and then try the questions and activities that follow.

### Tutorial Random Numbers



(<https://www.alexandriarepository.org/wp-content/uploads/20150802063654/tutorial6-BlockBooks-Pass1-BlockBooks-Pass2.mp4.mp4>)  
License: CC Attribution-ShareAlike 3.0 Unported (CC BY-SA 3.0)

## Questions

**Start Quiz** (<https://www.alexandriarepository.org/app/WpProQuiz/32>)

## Activities

### Activity A

Create a script that draws a triangle randomly on the canvas.

HINT- You will need to use the **start shape** and **end shape** blocks, as well as a **repeat \_\_ times** loop.

### Activity B

Create an image that utilises randomness.

Ensure that you demonstrate the use of at least 2 of the following:

- Randomising the location of the pencil
- Setting the size of a shape randomly
- Setting the width of the pen randomly
- Setting the fill colour randomly

Use the below Scribble window to generate your images.

 Click to open

**Legend:**

  
Run.

  
License: CC Attribution-ShareAlike 3.0 Unported (CC BY-SA 3.0)  
Pause.

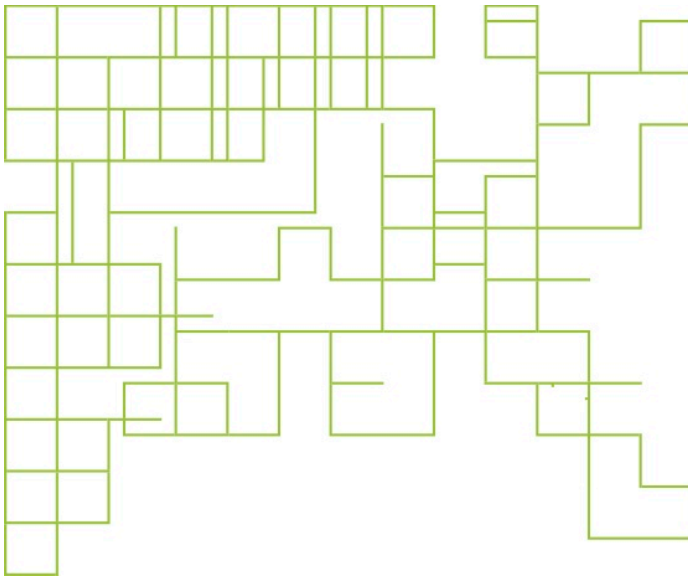
  
License: CC Attribution-ShareAlike 3.0 Unported (CC BY-SA 3.0)  
Stop.

  
License: CC Attribution-ShareAlike 3.0 Unported (CC BY-SA 3.0)  
Show blocks.

  
License: CC Attribution-ShareAlike 3.0 Unported (CC BY-SA 3.0)  
Expand stage.

(If this project does not open, see notes at bottom of page).





## 2.7 Conditionals

In this tutorial you will learn how to use conditional statements to change the behaviour of the pencil according to simple rules. Watch the short tutorial below and then try the questions and activities that follow.

### Tutorial Conditionals



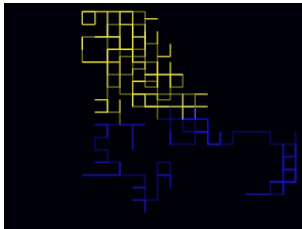
(<https://www.alexandriarepository.org/wp-content/uploads/20150802063638/tutorial7-BlockBooks-Pass1-BlockBooks-Pass2.mp4.mp4>)  
License: CC Attribution-ShareAlike 3.0 Unported (CC BY-SA 3.0)

## Questions

**Start Quiz** (<https://www.alexandriarepository.org/app/WpProQuiz/14>)

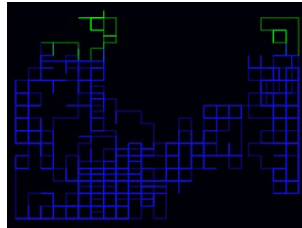
## Activities

Construct a script that draws a "drunkard's" walk, like the one used in the tutorial. Modify it to generate images similar to the ones on the following pages:



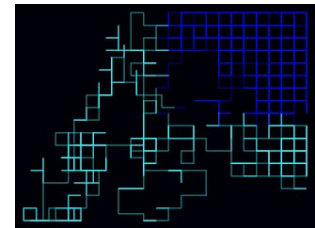
(<https://www.alexandriarepository.org/media/1>)

*License: CC Attribution-ShareAlike 3.0 Unported (CC BY-SA 3.0)*



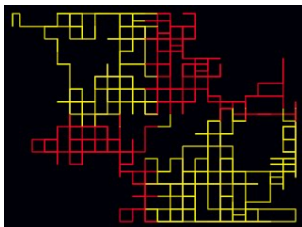
(<https://www.alexandriarepository.org/media/2-36>)

*License: CC Attribution-ShareAlike 3.0 Unported (CC BY-SA 3.0)*



(<https://www.alexandriarepository.org/media/3-26>)

*License: CC Attribution-ShareAlike 3.0 Unported (CC BY-SA 3.0)*



(<https://www.alexandriarepository.org/media/4-23>)

*License: CC Attribution-ShareAlike 3.0 Unported (CC BY-SA 3.0)*

### HINTS:

- Use the and block to specify two conditions that must be true. For example, if this is true AND that is true, do something.

### HINTS:

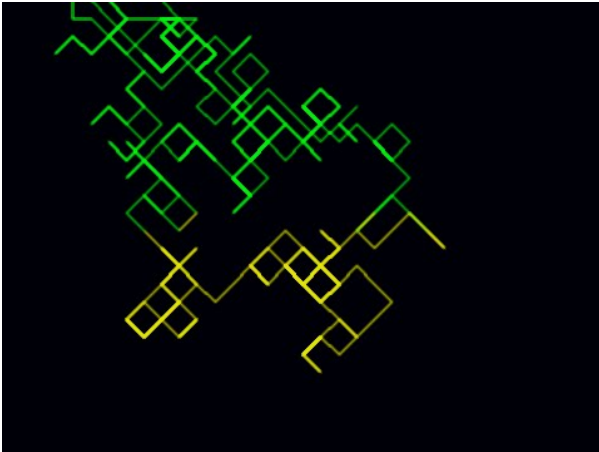
- Use the or block to specify two conditions that can be satisfied. For example, if EITHER this is true OR that is true, do something.
- Conditions can be nested within conditions, like in this example:



## Extension Exercise

Change your drunkard's walk script so the that walking happens on an angle, like the image below.

**HINT-** You should only have to make one change to your script!



Use the Scribble window below to generate you images.

 Click to open



**Legend:**

  
Run.

  
*License: CC  
Attribution-  
ShareAlike  
3.0 Unported  
(CC BY-SA  
3.0)*


Pause.

  
*License: CC  
Attribution-  
ShareAlike  
3.0 Unported  
(CC BY-SA  
3.0)*

Stop.

  
*License: CC  
Attribution-  
ShareAlike  
3.0 Unported  
(CC BY-SA  
3.0)*

Show  
blocks.

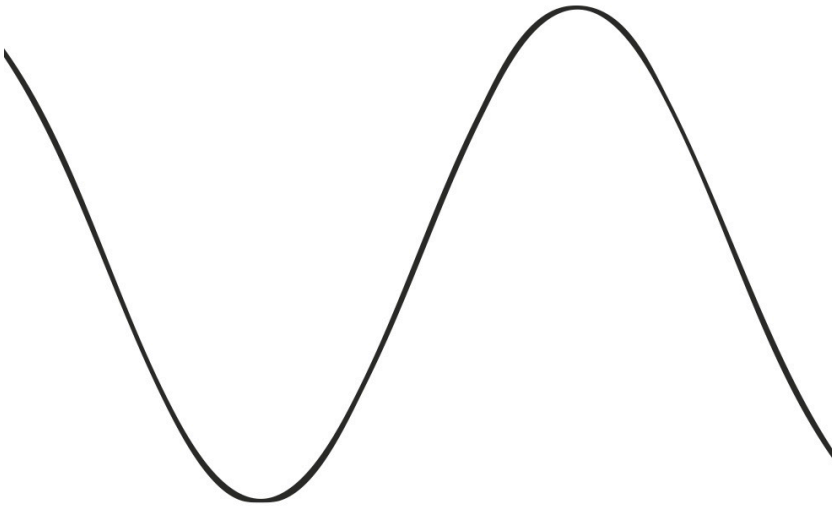
  
*License: CC  
Attribution-  
ShareAlike  
3.0 Unported  
(CC BY-SA  
3.0)*

Expand  
stage.

(If this project does not open, see notes at bottom of page).

## Note

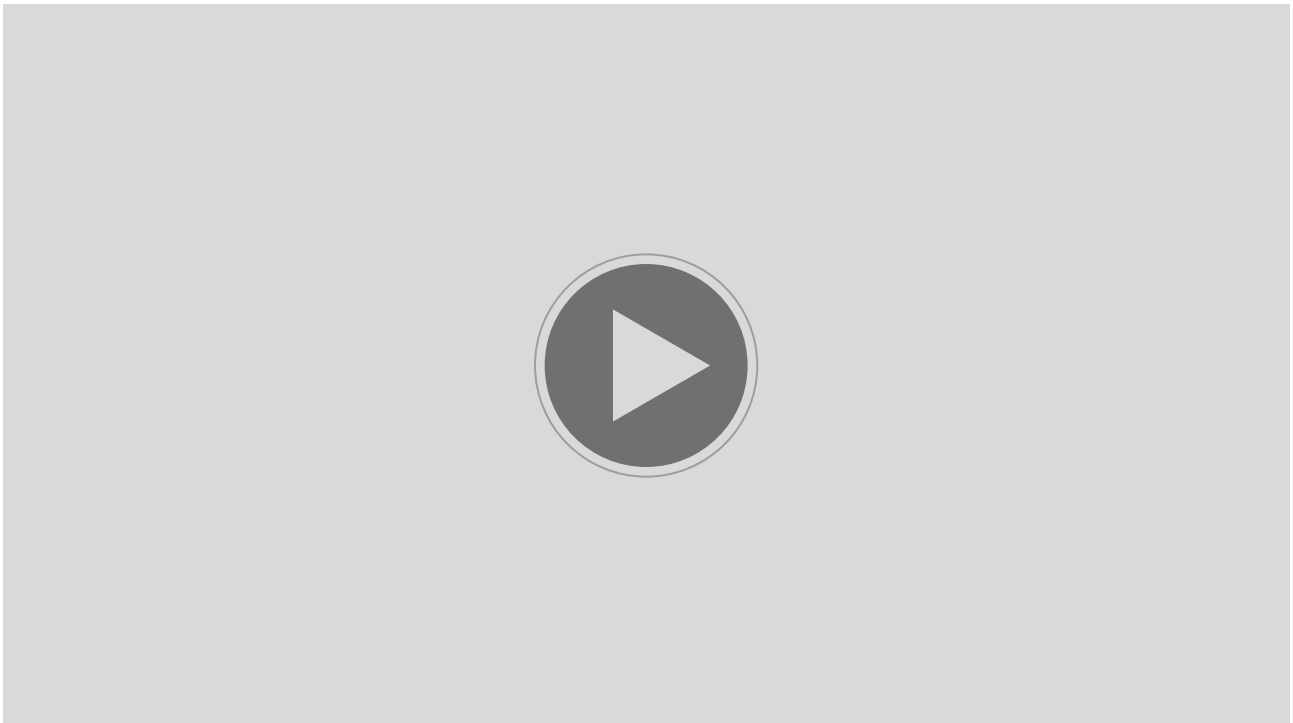
If you have problems with Scribble, try viewing this page in Firefox.



## 2.8 Drawing Functions

In this tutorial you will learn how to use the output of a mathematical function to draw lines and curves on the canvas. Watch the short tutorial below and then try the questions and activities that follow.

### Tutorial Drawing Functions



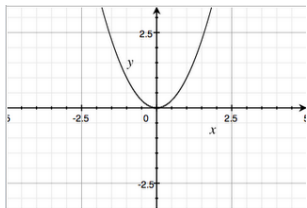
(<https://www.alexandriarepository.org/wp-content/uploads/20150802063703/tutorial8-BlockBooks-Pass1-BlockBooks-Pass2.mp4.mp4>)  
License: CC Attribution-ShareAlike 3.0 Unported (CC BY-SA 3.0)

## Questions

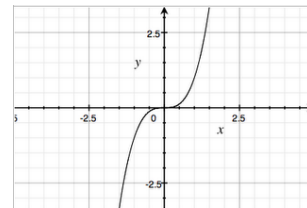
**Start Quiz** (<https://www.alexandriarepository.org/app/WpProQuiz/44>)

## Activities

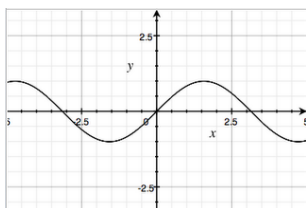
Construct a program that draws mathematical functions, similar to the one used in the tutorial. For each graph below, modify the script such that it draws the graph's function:



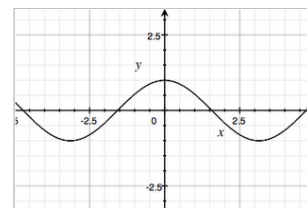
(<https://www.alexandriarepository.org/media/screen-shot-2012-03-16-at-1-24-26-pm>)



(<https://www.alexandriarepository.org/media/screen-shot-2012-03-16-at-1-25-46-pm>)



(<https://www.alexandriarepository.org/media/screen-shot-2012-03-16-at-1-28-38-pm>)



(<https://www.alexandriarepository.org/media/screen-shot-2012-03-16-at-1-28-48-pm>)

Use the Scribble window below to generate your images.

 Click to open

**Legend:**



Run.



Licence: CC  
Attribution-  
ShareAlike 3.0  
Unported (CC)

BY-SA  
3.0)

Pause.



Licens  
e: CC  
Attribu  
tion-  
ShareA  
like 3.0  
Unport  
ed (CC  
BY-SA  
3.0)

Stop.



Licens  
e: CC  
Attribu  
tion-  
ShareA  
like 3.0  
Unport  
ed (CC  
BY-SA  
3.0)

Show  
blocks



Licens  
e: CC  
Attribu  
tion-  
ShareA  
like 3.0  
Unport  
ed (CC  
BY-SA  
3.0)

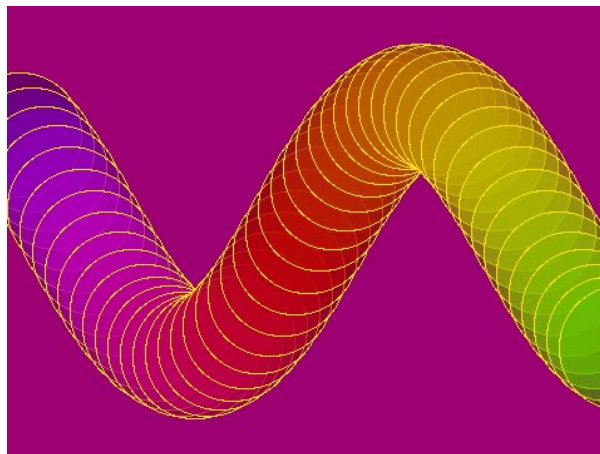
Expan  
d  
stage.

(If this  
project  
does  
not  
open,  
see

notes  
at  
botto  
m of  
page).

## Extension Exercise

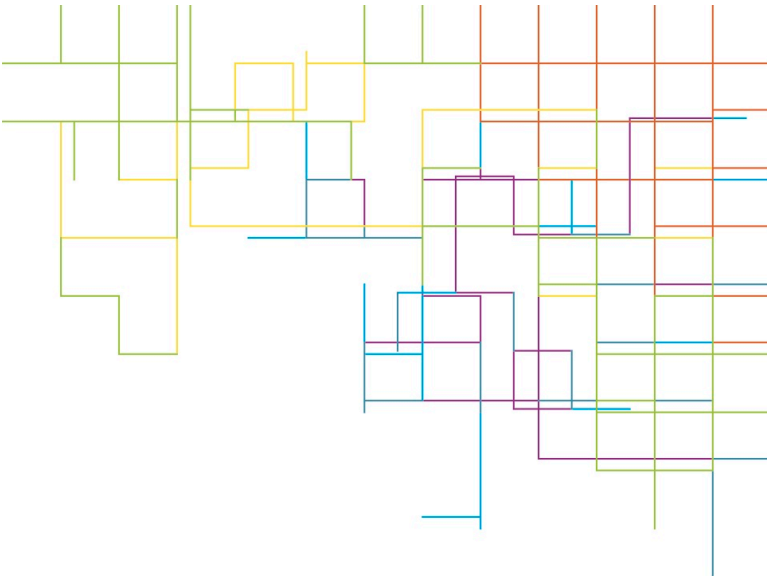
Modify the program you created for the exercises above, so that instead of drawing a function with a line, it draws shapes instead. Get creative with the colour of the canvas, and the colour of the shape. Here's an example using the sine wave function:



*Sine Wave Function with Shapes*

## Note

If you have problems with Scribble, try viewing this page in Firefox.

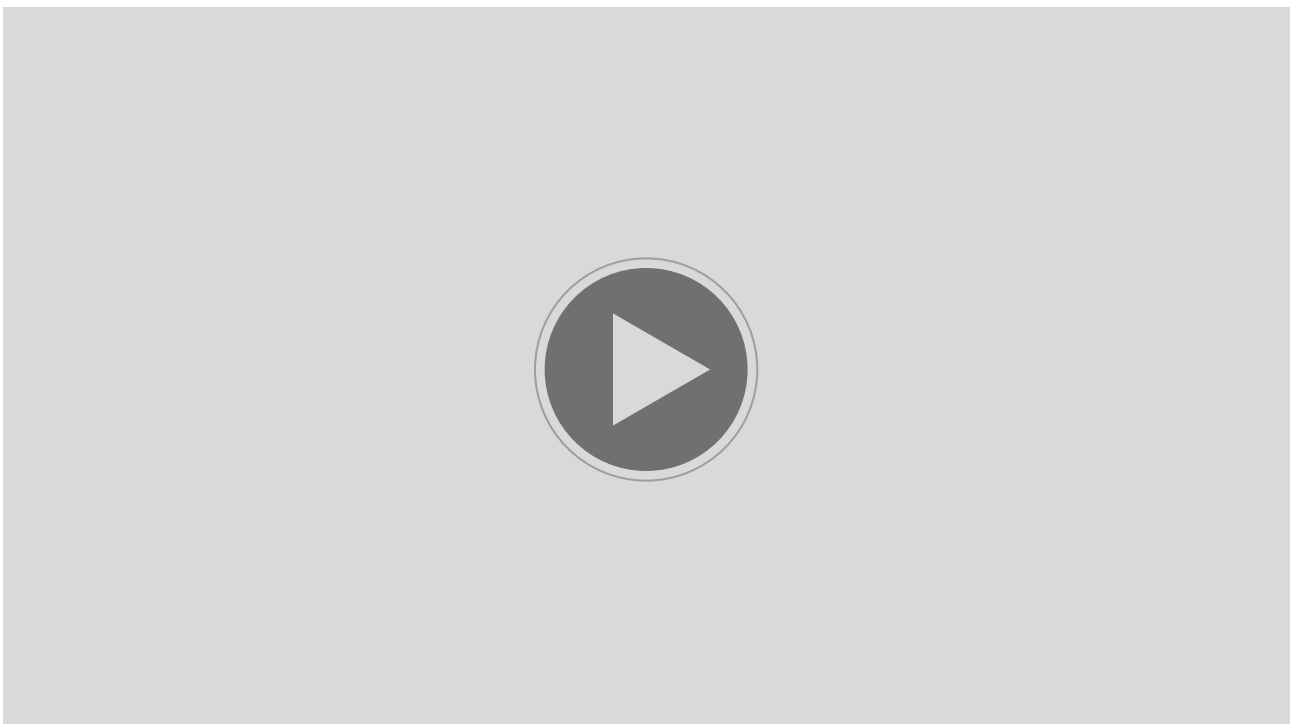


## 2.9 Custom Blocks

In this tutorial you will learn how to create custom blocks, which can reduce the size of your script and make it easier to understand and edit.

Watch the short tutorial below and then try the questions and activities that follow.

### Tutorial Custom Blocks



(<https://www.alexandriarepository.org/wp-content/uploads/20150802063628/tutorial9-BlockBooks-Pass1-BlockBooks-Pass2.mp4.mp4>)  
License: CC Attribution-ShareAlike 3.0 Unported (CC BY-SA 3.0)

## Questions

**Start Quiz** (<https://www.alexandriarepository.org/app/WpProQuiz/46>)

## Activities

Create a custom block to perform the task described.

For example:

Task:

- Move the pencil to the centre of the canvas
- Set the pen size to 5.

Custom block:

### Activity A

Create a custom block for the following task:

- Change the y position of the pencil by 5
- Change the fill colour by 20
- Draw a rectangle, 50 by 100

### Activity B

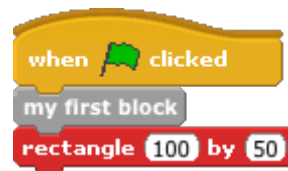
Create a custom block for the following task:

- Set the pen colour to black
- Point in a random direction
- Move 200 steps

## Extension Exercise

Two custom blocks are created, my first block and my second block:

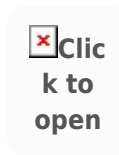
What will be drawn on the canvas when the following script is run? Why?



*License: CC Attribution-ShareAlike 3.0 Unported  
(CC BY-SA 3.0)*

**HINT-** There might be a serious flaw in this program!

Use the Scribble window below to generate you images.







**Legend:**

  
Run.

  
License: CC Attribution-ShareAlike 3.0 Unported (CC BY-SA 3.0)  
Pause.

  
License: CC Attribution-ShareAlike 3.0 Unported (CC BY-SA 3.0)  
Stop.

  
License: CC Attribution-ShareAlike 3.0 Unported (CC BY-SA 3.0)  
Show blocks.

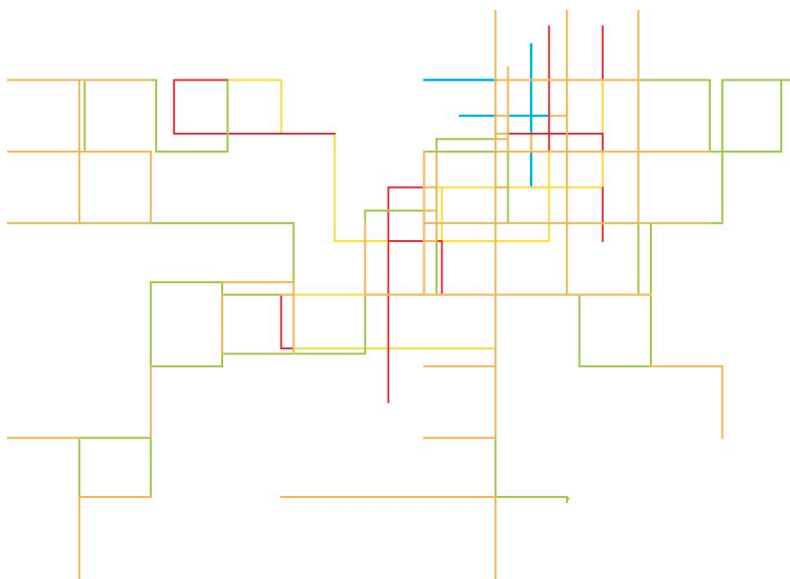
  
License: CC Attribution-ShareAlike 3.0 Unported (CC BY-SA 3.0)  
Expand stage.

(If this project does not open, see notes at bottom of page).

## Note

If you have problems with Scribble, try viewing this page in Firefox.

---



## 2.10 Parameters

In this tutorial you will learn how to add parameters to custom blocks to provide more flexibility. Watch the short tutorial below and then try the questions and activities that follow.

### Tutorial Parameters



(<https://www.alexandriarepository.org/wp-content/uploads/20150802063646/tutorial10-BlockBooks-Pass1-BlockBooks-Pass2.mp4.mp4>)  
License: CC Attribution-ShareAlike 3.0 Unported (CC BY-SA 3.0)

## Questions

**Start Quiz** (<https://www.alexandriarepository.org/app/WpProQuiz/43>)

## Activities

Use the project window (below) to create parameterised custom blocks for the following tasks. Double check that your block works correctly by placing it in a script and running it.


A bold letter in place of a number indicates that a parameter should be used for that value.


1. Move  $x$  amount of steps and draw a rectangle 100 by 20.
2. Move  $x$  amount of steps, and draw a rectangle  $y$  by  $z$ .
3. Put the pen down and repeat the following  $x$  times: move 50 steps and rotate  $(360 / x)$  degrees.

Describe what this custom block does.


 Click to open

**Legend:**

  
Run.

  
*License: CC Attribution-ShareAlike 3.0 Unported (CC BY-SA 3.0)*


Pause.

  
*License: CC Attribution-ShareAlike 3.0 Unported (CC BY-SA 3.0)*

Stop.

  
*License: CC Attribution-ShareAlike 3.0 Unported (CC BY-SA 3.0)*

Show blocks.

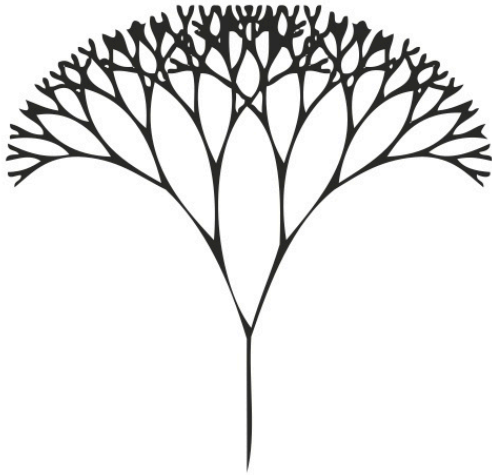
  
*License: CC Attribution-ShareAlike 3.0 Unported (CC BY-SA 3.0)*

Expand stage.

(If this project does not open, see notes at bottom of page).

## Note

If you have problems with Scribble, try viewing this page in Firefox.



## 2.11 Recursion

In this tutorial you will learn about the concept of recursion, which can be used to draw images that are composed of similar sub-images, like fractals and trees. Watch the short tutorial below and then try the questions and activities that follow.

### Tutorial Recursion



(<https://www.alexandriarepository.org/wp-content/uploads/20150802063619/tutorial11-BlockBooks-Pass1-BlockBooks-Pass2.mp4.mp4>)  
License: CC Attribution-ShareAlike 3.0 Unported (CC BY-SA 3.0)

## Questions

**Start Quiz** (<https://www.alexandriarepository.org/app/WpProQuiz/42>)

---