

# Algorithmic & Computational Thinking In Design

By Nettrice R. Gaskins, PhD

# Definition

An **algorithm** is a well-defined procedure that takes some value, or set of values, as input and produces some value, or set of values, as output. An algorithm is thus a sequence of steps that transform the input into the output.

**Algorithmic thinking** is a way of getting to a solution through the clear definition of the steps needed. Rather than coming up with a single answer to a problem, students can create algorithms for many solutions.

**Computational thinking** is a set of problem-solving methods that involve expressing problems and their solutions in ways that a computer could execute.

# Introduction



See notes, no. 79.

Dutch artist **MC Escher** used an algorithm (script) to create this image of butterflies:

1. Draw a butterfly.
2. Color the butterfly's wings in such a way that mutually orthogonal circles are formed.
3. Arrange the butterflies in a group or pattern, starting with bigger objects that get smaller and smaller as they approach the center.
4. Rotate each row of butterflies to fill the space.

Image on the left: "Circle Limit with Butterflies" by MC Escher

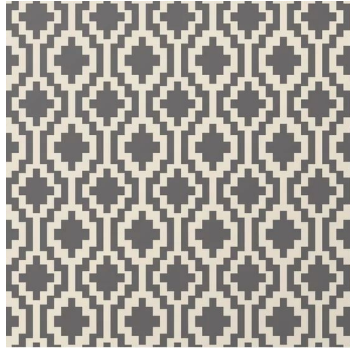
# Introduction



The steps to create Escher's art constitute an algorithm. They are not a computer program—they are just specific directions on how to achieve something.

In order to simulate this art on a computer you will need a programming language that generates patterns as data structures and each butterfly (shape) as an object.

# Introduction



With Escher's algorithmic thinking process in mind we can look at many different real life patterns (data structures) and identify algorithms that can be used by a computer to generate shapes and patterns.

Images counter-clockwise from top left: Mapuche textile print, Mardi Gras Indian patches, Native American star quilt, Kuba cloth.

# General Terms



Program



Variables



Data & Math



Modify & Control

We can think of computational thinking as baking a cake. The finished cake is the program that contains different components (i.e. blocks).

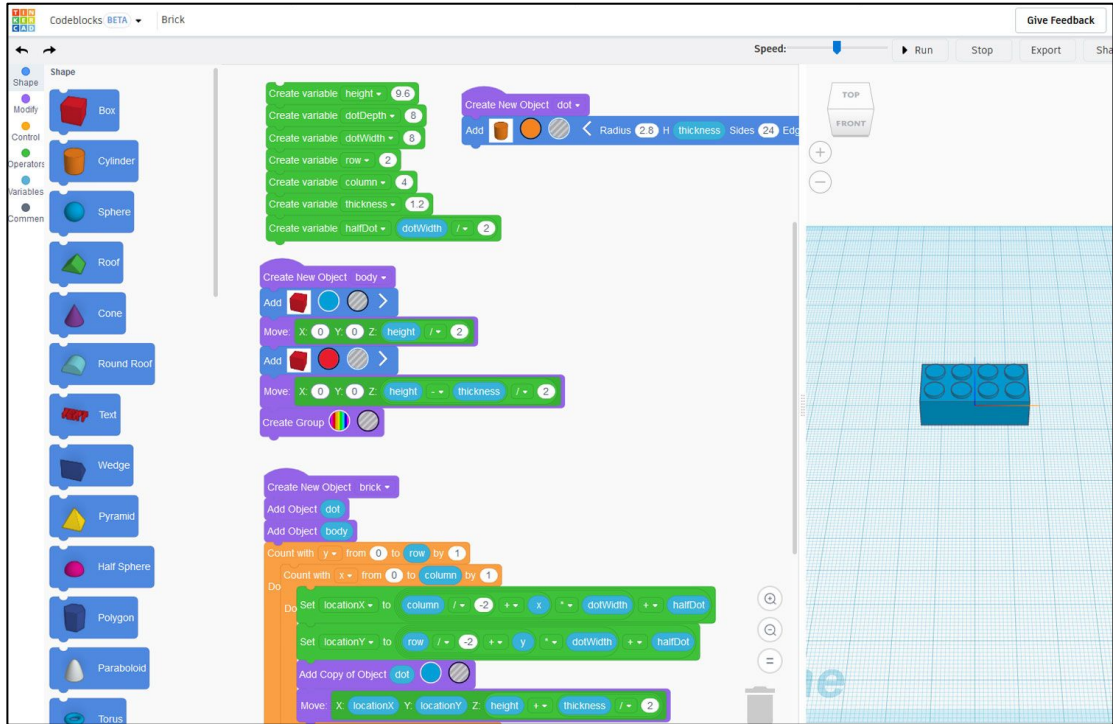
# The Program



In computing, a program is a specific set of ordered operations for a computer to perform or follow, to make something.

People create programs using **programming languages.**

# The Program



Codeblocks uses a visual programming language to create 3D designs that can be 3D printed.

The program generates a 3D object.



# Variables



Variables are components that perform calculations or execute instructions.

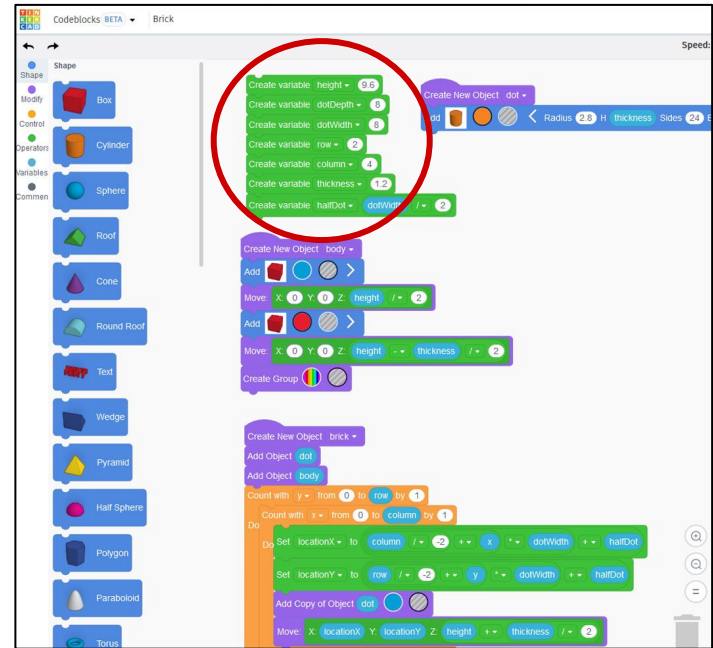
In visual programming, **variable blocks** are used to hold values and strings in variables... like bowls or cups are used to hold ingredients.

# Variables

Variables can be created and used in other blocks.

This is like using the same measuring cup for different tasks.

- > create variable “cup”
- > indicate how many cups or what kind of cup
- > use this variable (in script) when needed



# Data & Math



Data lists all the variables created in the program (like the ingredients list for a cake).

Math defines variables, changes their values, and generates random numbers.

# Data & Math

Codeblocks NEW Mardi Gras Patch

Speed

Shapes

Modify

Control

Math

Data

Mark Up

Data

- i
- length2
- offset
- ray number
- ray rotate
- thickness
- ray
- base
- pattern
- object0

Comment //

Say Message

Create Variable thickness 3

Create Variable length2 10

Create Variable offset 5

Create Variable ray number 10

Create Variable ray rotate 25

Create New Object base

Add Radius 15 H thickness Sides 20 edge 0 Edge

Move: X: 0 Y: 0 Z: 0

Create New Object ray

Add W thickness L 1 H 1 edge 0 Edge Steps 10

Move: X: 0 Y: 0 Z: 2

Random between 0 and 10

90

Data (blue) lists all the variables created in the program.

Math (green) defines variables, changes their values, and generates random numbers.

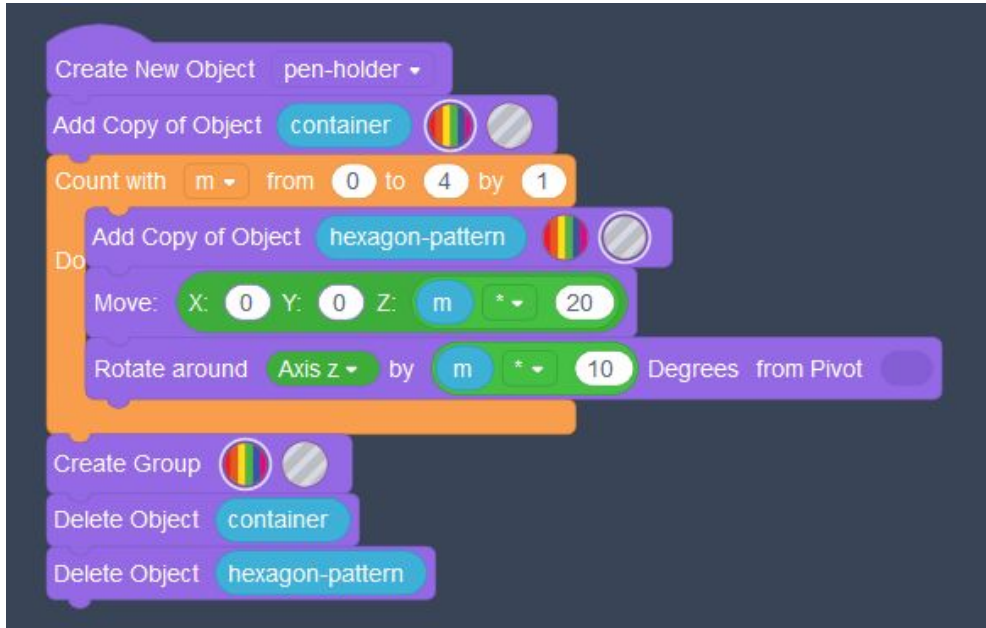
# Modify & Control



Modify allows you to transform or change something in a recipe or script.

Controls help you create a more efficient recipe (script).

# Modify & Control

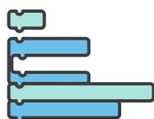




Modify blocks (purple) allow you to change the attributes of existing shapes, like their size, color, position, and rotation.

Control blocks (orange) let you define loops to repeat actions.

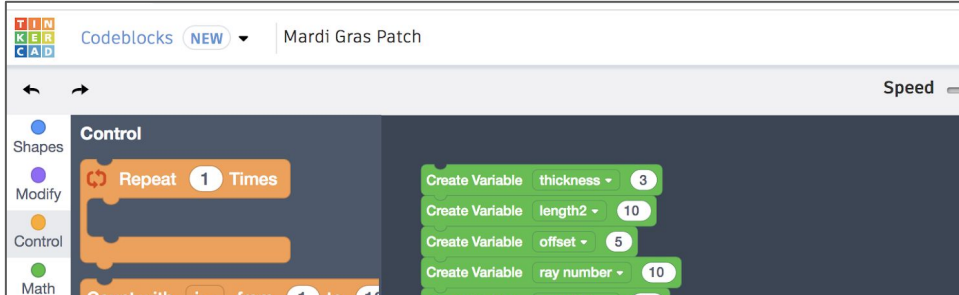
# Tinkercad Codeblocks

It's as easy as...

|                                                                                                                                                     |                                                                                                                                             |                                                                                                                                                        |
|-----------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>Stack</p>  <p>Drag and drop blocks together to make shapes.</p> | <p>Move</p>  <p>Re-order blocks to refine your design.</p> | <p>Run</p>  <p>Run the code and watch your creation come to life.</p> |
|-----------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|

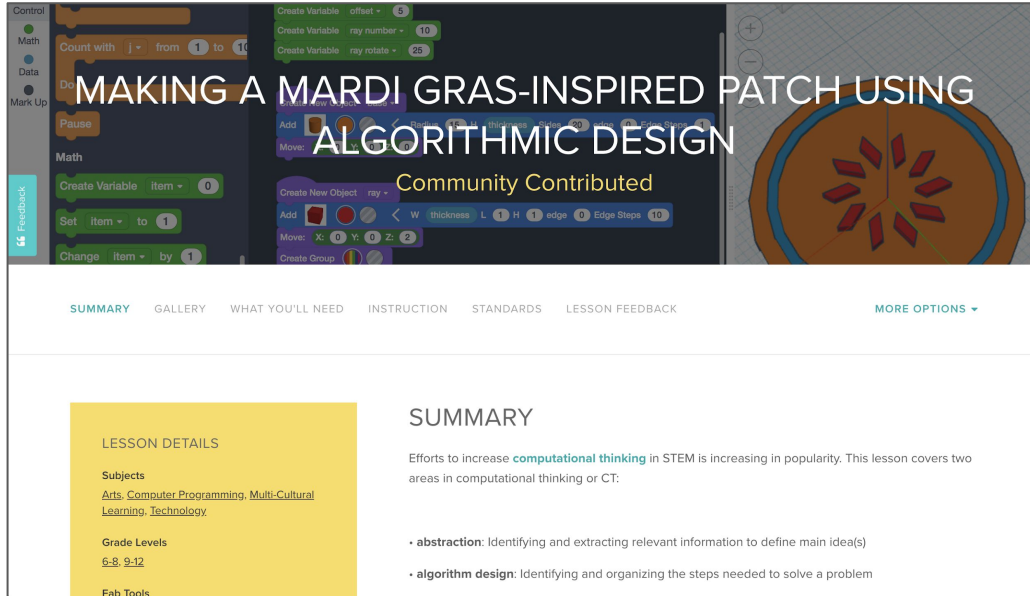
Tinkercad Codeblocks procedurally generates 3D shapes using a visual programming language or VPL.

The resulting model can be exported and 3D printed.



Tutorial: <https://maker.pro/custom/tutorial/an-introduction-to-tinkercad-codeblocks>

# SCOPES-DF Lesson



The image shows a screenshot of a Codeblocks interface. The top part displays a 3D model of a Mardi Gras mask with a blue and red pattern. Below the model, the text "MAKING A MARDI GRAS-INSPIRED PATCH USING ALGORITHMIC DESIGN" is overlaid in large white letters. The interface includes a sidebar with categories like Control, Math, Data, and Mark Up. The main area shows a "Community Contributed" section with various code blocks such as "Create Variable", "Count with", "Set item to", and "Change item by". Below the code blocks, there is a navigation menu with options: SUMMARY, GALLERY, WHAT YOU'LL NEED, INSTRUCTION, STANDARDS, LESSON FEEDBACK, and MORE OPTIONS. The "SUMMARY" section is highlighted, showing the following details:

**LESSON DETAILS**

**Subjects**  
[Arts](#), [Computer Programming](#), [Multi-Cultural Learning](#), [Technology](#)

**Grade Levels**  
6-8, 9-12

**Fab Tools**

**SUMMARY**

Efforts to increase **computational thinking** in STEM is increasing in popularity. This lesson covers two areas in computational thinking or CT:

- **abstraction**: Identifying and extracting relevant information to define main idea(s)
- **algorithm design**: Identifying and organizing the steps needed to solve a problem

This lesson covers two areas in computational thinking or CT:

- **abstraction**
- **algorithmic design**

Students use algorithmic and computational thinking, with Codeblocks to generate 3D designs.

Lesson Plan: <https://www.scopesdf.org>