

# TINKERCAD



# CODEBLOCKS

# FOR TEACHERS

EDITED BY INTELLIGENT SYSTEMS LABORATORY  
OF THE UNIVERSITY OF THESSALY



CLIMATE CHANGE AND NATURAL DISASTERS AWARENESS RAISING USING VIRTUAL WORLDS



[www.vr4clima.eu](http://www.vr4clima.eu)

This Guide is published as part of the “Climate Change and Natural Disasters Awareness Raisin using Virtual Worlds - VR4Clima”. Project funded by the European Union's Erasmus+ programme (2023-1-EL01-KA220-SCH-000155463). This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

### **Title**

Tinkercad Codeblocks for Teachers

### **Year**

2025

### **Publisher**

Athanasios Kakarountas, University of Thessaly, Greece

Angeliki Karageorgopoulou, University of Thessaly, Greece

### **Reviewers**

Jovana Jankov, Center for Promotion of Science, Serbia

Manolis Voutsakis, National Technical University of Athens, Greece

Sonal Ahuja, VRAcademi Limited, Ireland

Maria Zarkada, 16th Experimental Primary School of Lamia, Greece

Natalia Budinski, Petro Kuzmjak School, Serbia

### **Design and layout**

University of Thessaly, Greece

### **Licence**

CREATIVE COMMONS BY-NC-SA 4.0 DEED

Attribution-Noncommercial-Share Alike 4.0 International

You are free to:

- Share - copy and redistribute the material in any medium or format.
- Adapt - remix, transform and build upon the material.
- The licensor cannot revoke these freedoms as long as you comply with the licence conditions.

Under the following conditions:

- Attribution - You must give appropriate credit, provide a link to the licence and indicate whether any modifications have been made. You may do this in any reasonable way, but not in a way that suggests that the licensor approves of you or your use.
- Non-commercial - You may not use the material for commercial purposes.
- ShareAlike - If you remix, transform or develop the material, you must distribute your contributions under the same licence as the original.
- No additional restrictions - You may not apply any legal conditions or technological measures that legally prevent others from doing anything the licence allows...

Notices:

- You are not required to comply with the licence for material in the public domain or where your use is permitted by an applicable exception or limitation.
- No warranty is given. The licence may not give you all the authorisations necessary for the use you wish to make of it. For example, other rights such as publicity, privacy or moral rights, may restrict your use of the material...

# CONTENT

<b>1</b>	<b>Part 1: Getting Started with Tinkercad Codeblocks</b>	<b>4</b>
<b>1.1</b>	<b>What is Tinkercad?</b>	<b>4</b>
<b>1.2</b>	<b>Introduction to Codeblocks</b>	<b>5</b>
<b>1.3</b>	<b>Setting Up Your Tinkercad Account</b>	<b>7</b>
<b>1.4</b>	<b>Exploring the Codeblocks Workspaces</b>	<b>9</b>
<b>1.4.1</b>	<b>Layout of working space</b>	<b>11</b>
<b>1.4.2</b>	<b>Blocks Panel:</b>	<b>10</b>
<b>1.4.3</b>	<b>Blocks Editor</b>	<b>11</b>
<b>1.4.4</b>	<b>3D Viewer</b>	<b>11</b>
<b>2</b>	<b>Part 2: Core Coding Concepts with Codeblocks</b>	<b>12</b>
<b>2.1</b>	<b>Basic Commands and Shapes</b>	<b>12</b>
<b>2.1.1</b>	<b>Place</b>	<b>13</b>
<b>2.1.2</b>	<b>Modify</b>	<b>14</b>
<b>2.1.3</b>	<b>Control</b>	<b>19</b>
<b>2.1.4</b>	<b>Legacy</b>	<b>23</b>
<b>2.2</b>	<b>Using Variables</b>	<b>24</b>
<b>2.3</b>	<b>Mathematics for Codeblocks</b>	<b>29</b>
<b>3</b>	<b>Part 3: Step-by-Step Projects</b>	<b>31</b>
<b>3.1</b>	<b>Project 1: Creation of a flower</b>	<b>32</b>
<b>3.2</b>	<b>Project 2: Creation of a name Tag</b>	<b>34</b>
<b>3.3</b>	<b>Project 3: Creation of a geometric spiral</b>	<b>37</b>
<b>4</b>	<b>Part 4: Exporting and Troubleshooting</b>	<b>40</b>
<b>4.1</b>	<b>Exporting and Printing 3D Models</b>	<b>40</b>
<b>4.2</b>	<b>Common Troubleshooting Tips</b>	<b>41</b>
<b>5</b>	<b>Appendices</b>	<b>43</b>
<b>5.1</b>	<b>Tinkercad Keyboard Shortcuts</b>	<b>43</b>
<b>5.2</b>	<b>Glossary of Terms</b>	<b>44</b>
<b>5.3</b>	<b>Resources for Further Learning</b>	<b>47</b>

## PART 1

# GETTING STARTED WITH TINKERCAD CODEBLOCKS

## 1. What is Tinkercad?

Tinkercad is a free web app used for 3D Design, Electronics and Coding. It was officially launched in 2011. After being acquired by Autodesk in 2013, Tinkercad rapidly evolved, especially in education and the maker community.

With Tinkercad you can combine basic shapes and create 3d models, manipulate them and export them for 3d printing or for inserting them inside virtual worlds.

You can also design and simulate electronic circuits using the circuits workspace, and then control them using a code editor with support for C++, Javascript, Python etc.

What has made Tinkercad popular among schools and learning institutions is its applications to Education. Students can create simple to complex 3D Objects which can be exported for 3d Printing or for use inside Virtual Worlds. By integrating Science, Technology, Engineering, Arts and Mathematics (STEAM), it enhances its users' problem solving skills. It develops computational and algorithmic thinking by introducing block-based coding for 3D design.

It is an ideal tool for rapid prototyping and creating conceptual models of products, gadgets or everyday objects.

# 1.2 Introduction to Codeblocks

Codeblocks is a visual programming environment that allows users to create 3D models using block-based coding (like Scratch or Blockly). It consists of programming instructions represented as movable, colored blocks. Using this feature of Tinkercad, instead of manually drawing shapes, students "code" the design by snapping together blocks that define actions like moving, rotating, scaling, or creating shapes. The sequence of blocks creates parametric 3D models. In that way designs can be quickly modified by changing variables or parameters.

Using Codeblocks, students can build geometric patterns (spirals, towers etc.), create repetitive structures using loops, program the position, size, and rotation of 3D shapes and design dynamic models that can be easily adjusted by changing parameters.

In the following Table the difference between Codeblocks and regular Tinkercad Design is presented:

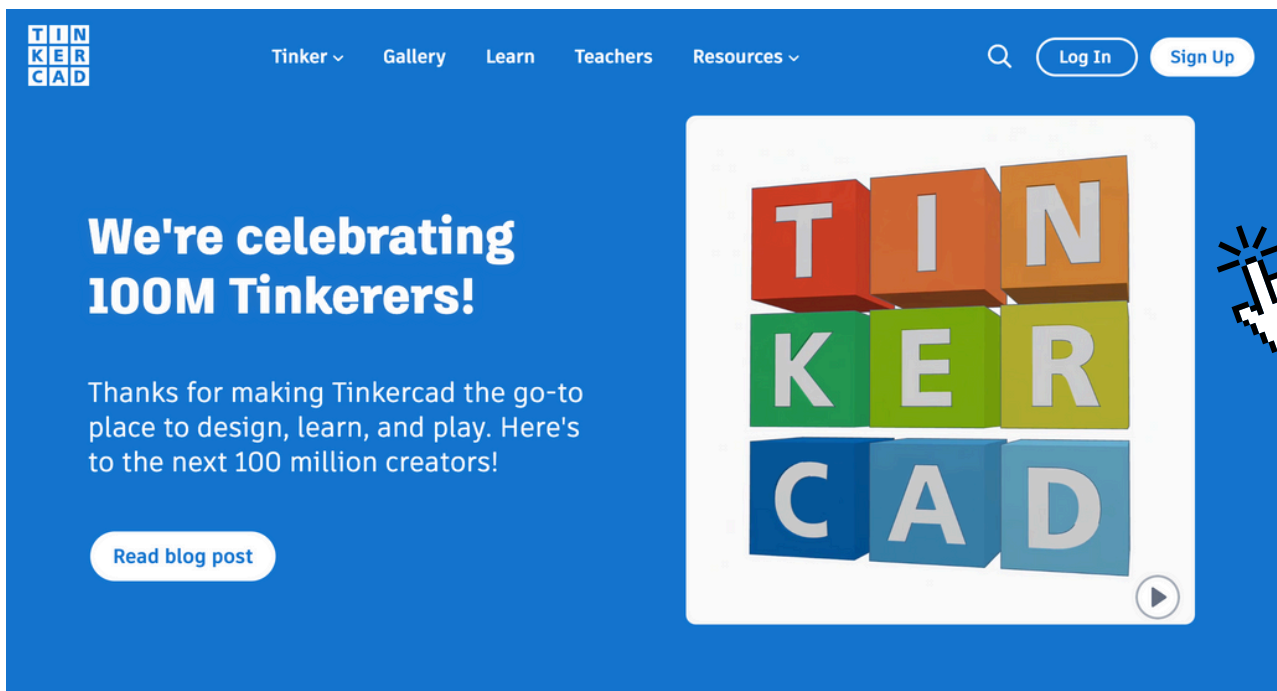
Feature	Codeblocks	Regular Tinkercad Design
<b>Interface</b>	Visual, block-based programming (similar to Scratch)	Drag-and-drop 3D workspace with direct manipulation
<b>Design Method</b>	Create shapes and patterns by coding with block commands	Manually place, move, and adjust shapes in the workspace
<b>Workflow</b>	Step-by-step instructions to generate the model programmatically	Freeform design by dragging, resizing, and aligning objects
<b>Flexibility</b>	Best for patterns, repetitions, parametric designs	Best for unique, organic, or detailed manual designs
<b>Automation</b>	Easy to automate repetitive tasks and transformations	Repetitive tasks must be done manually unless grouped and duplicated
<b>Learning Focus</b>	Develops computational thinking and introduces coding concepts	Develops spatial awareness and manual 3D modeling skills
<b>User Level</b>	Ideal for beginners in coding and algorithmic design	Ideal for beginners in 3D design without coding
<b>Example Projects</b>	Spiral towers, geometric patterns, tiled arrays	Keychains, custom characters, buildings, furniture
<b>Strengths</b>	Automating patterns, fast editing via variables, introducing programming logic	Quick, intuitive modeling, visual creativity without coding
<b>Limitations</b>	Less suited for detailed freeform design	Time-consuming for creating large patterns or parametric changes



# 1.3 Setting Up Your Tinkercad Account

Go to

Click on the **Sign Up** button.



Tinkercad is a free web app for 3D design, electronics, and coding, trusted by over 100 million people around the world.

Build STEM confidence by bringing project-based learning to the classroom.

[Start Tinkering](#)

[Join Class](#)



**Free for everyone**



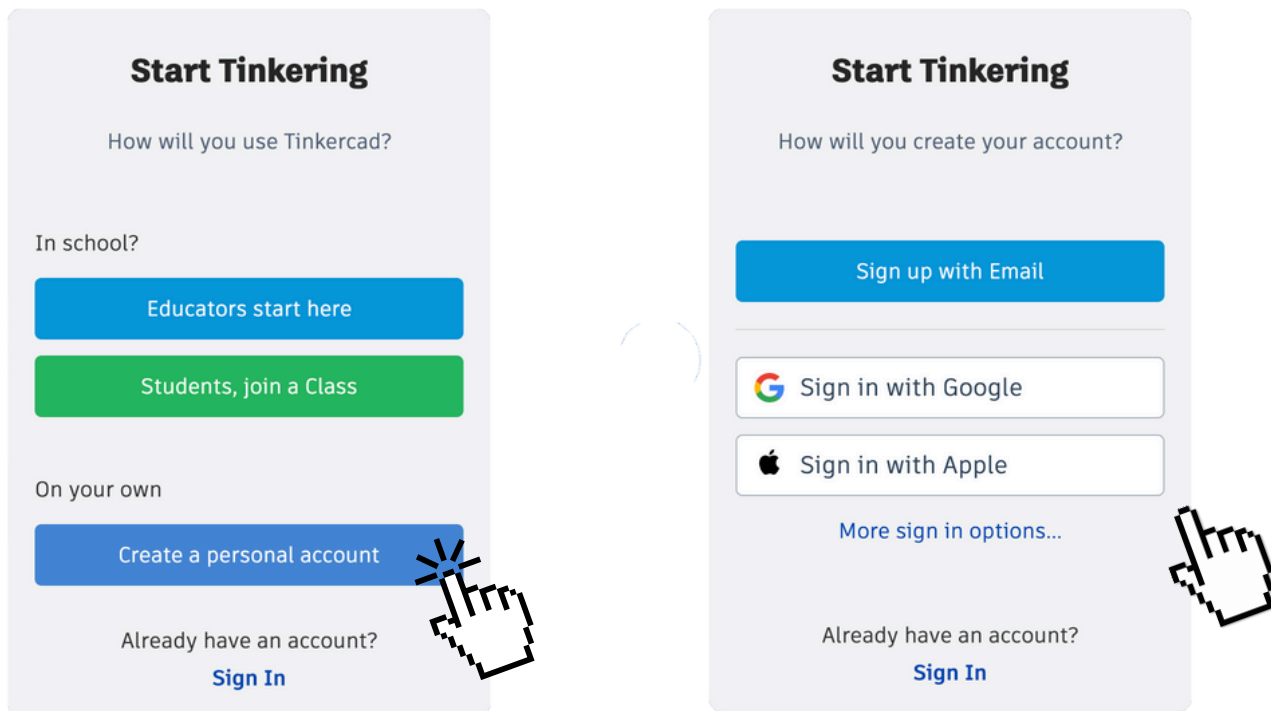
**Learn by doing**



**Safe for all ages**

Click on “Create Personal Account”

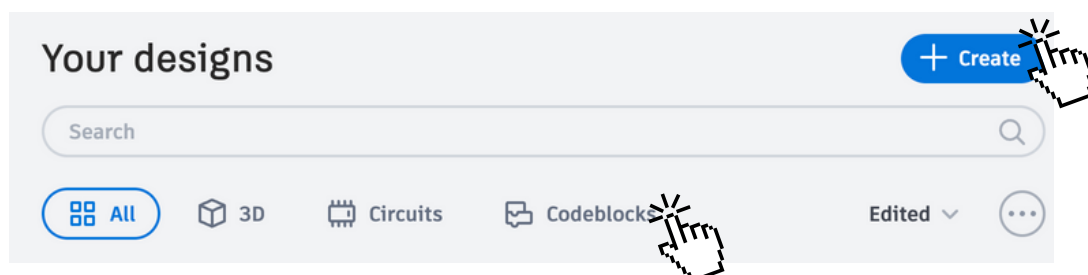
Select a way to Sign Up, eg. Sign in with Google



Follow further steps to create the account.

Once in Tinkercad **Home page**, click **Create** button and choose to create your first  **3D Designs**

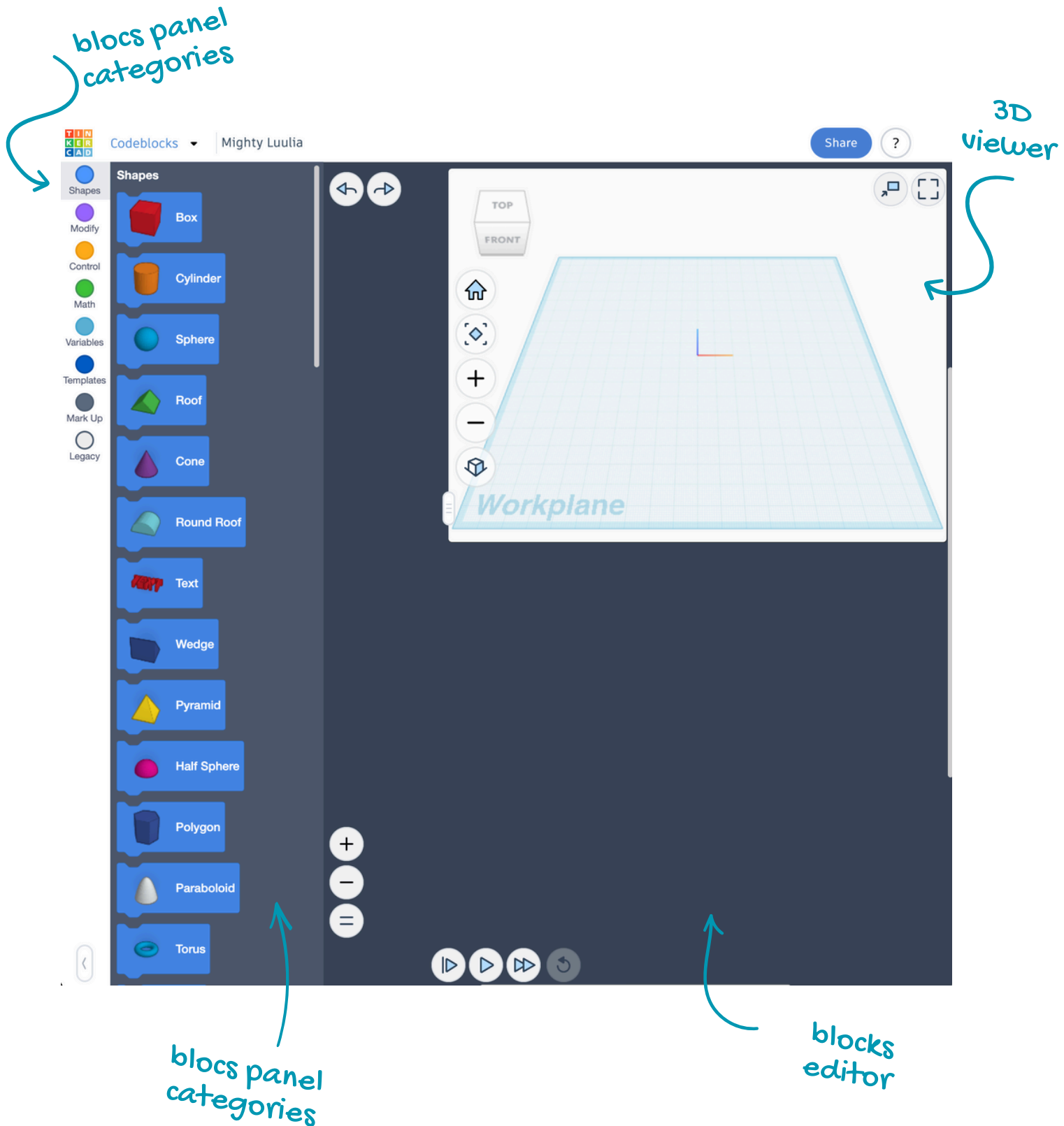
Or, on the left side menu go to Designs, and from there chose Codebloccs.



There is Tinkercad’s [Help Center](#) If you get stuck at this point or If you have further questions.

# 1.4 Exploring the Codeblocks Workspace









## 1.4.1 Layout of working space



## 1.4.2 Blocks Panel

The Blocks Panel is the area in Tinkercad Codeblocks where you can find all the coding blocks you need to build your design. It is located on the left side of the screen and is organized into different categories, making it easy to find the type of block you want to use.

Each block represents a specific action, control, or object you can add to your project, and you can drag and drop these blocks into your workspace to create your 3D models.

Categories of the Blocks Panel	
 Shapes	Contains basic 3D shapes like boxes, cylinders, spheres, wedges, and more. These are the building blocks of your design.
 Modify	Includes blocks that allow you to change the size, position, and rotation of shapes. This is how you control where and how each object appears.
 Control	Provides logic structures like loops (repeat), which help you automate actions, create patterns, or repeat shapes without manually duplicating them.
 Math	Offers blocks to perform mathematical operations. Useful when you want to set dynamic values or calculate positions.
 Variables	Create, manage, and use custom variables that store values for your designs and code
 Templates	Provides a collection of pre-built code structures that help you quickly start or organize your designs
 Mark Up	Provides tools that help you organize, label, and document your code visually within the workspace. These blocks do not affect the 3D design itself but are used to make your code easier to understand and manage
 Legacy	Contains older or deprecated blocks that were used in previous versions of Codeblocks but are no longer recommended for new projects

## 1.4.3 Blocks Editor

It is the visual programming interface where users create 3D designs using drag-and-drop coding blocks instead of traditional 3D modeling tools. It uses a system similar to Scratch or Blockly, where you build code by stacking and connecting visual blocks. These blocks represent commands for creating and transforming 3D shapes. The Blocks Editor is designed especially for beginners and students to learn computational thinking, geometry, and design automation in a hands-on way.

Using the Blocks Editor, you can create repeating patterns or parametric designs, automate complex models that would be hard to make manually or learn coding concepts while making 3D objects.

## 1.4.4 3D Viewer

It is the part of the interface where you can see and interact with your 3D model as it is generated by your code.

3D Viewer Controls		
	Home view	Snaps the camera back to the default view and centers your model
	Fit all in view	Frames your entire 3D model in the viewer
	Shrink 3D view	Reduces the size of the 3D viewer area
	Full screen 3D view	Expands the 3D viewer to cover the entire screen
	View cube	Interactive tool that helps you navigate and change the orientation of your 3D model
	Start	Starting point of the design

## PART 2

# CORE CODING CONCEPTS WITH CODEBLOCKS

## 2.1 Basic Commands and Shapes

### 2.1.1 Place

In order to **add** a command into the Blocks Editor, you drag it from the Blocks Panel.

In order to **delete** a command from the Blocks Editor, you drag it to the Blocks Panel

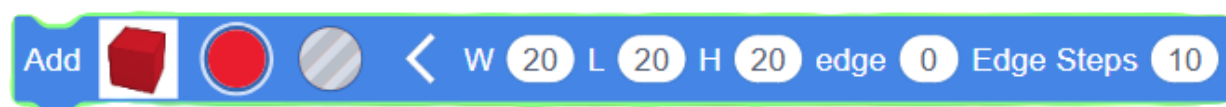
### Cube



The Add Command places the shape at the center of the 3D Viewer.

The shape is added with its default parameters.

In order to change parameters, you click on the arrow placed at the end of the command.

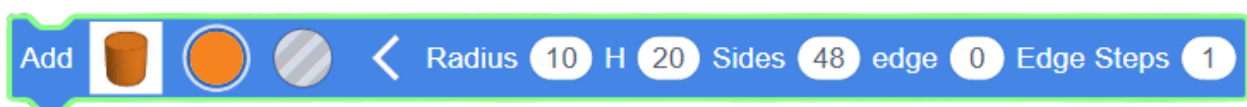


This way you can specify the 3 dimensions of the Cube, the Edges and number of edge steps.

The number of Edges represent how many segments each edge of the cube is divided into for rounding purposes. In the same manner, the Edge steps determine how smooth the rounded edges appear.

You can also choose between several colors for your shape or select the hole. The hole is used for subtracting the shape from another shape it is grouped with.

## Cylinder

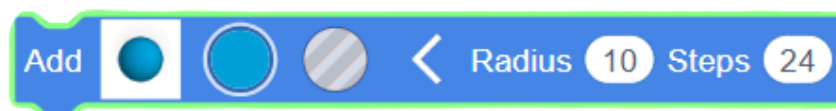


The number of sides determine the number of flat faces (segments) used to approximate the circular shape of the cylinder. A higher number of sides makes the cylinder look smoother and more circular while a lower number gives it a polygonal look.

The number of edges represents the number of segments along the height of the cylinder. Higher values allow smoother deformation if you're modifying or distorting the shape later.

The number of edge steps controls the smoothness of the rounding between the cylinder's faces.

## Sphere



The number of Steps represents the number of segments used to build the sphere's surface.

## Other shapes

There is a variety of shapes that can be added in Codeblocks and their parameters follow the same logic.

## 2.1.2 Modify

The x axis is the one represented with red color.

The y axis is represented with blue color.

The z axis is represented with green color.

### Select All



It is used to select every object currently present in the design workspace at that point in the code sequence.

It allows you to apply modifications or group actions (like moving, rotating, scaling, grouping, or aligning) to all objects that have been added so far in your code.

### Move

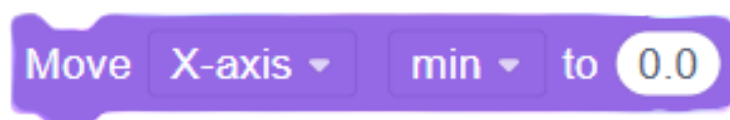


It is used to reposition shapes in the 3D workspace by shifting them along the X, Y, and/or Z axes.

The "Move" block translates (moves) a selected shape or group of shapes without rotating or scaling them. Note that the move is relative to the current position of the shape and that negative values move the shape in the opposite direction.

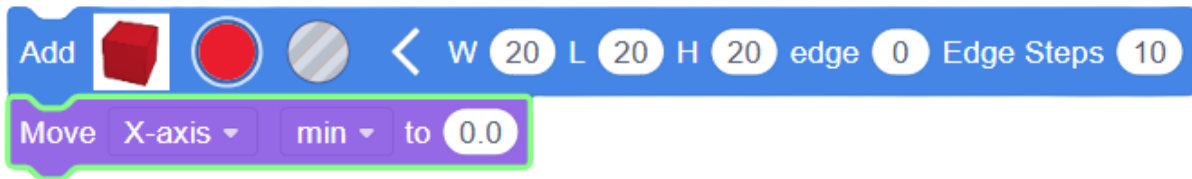
The block works on the most recently added or selected shape unless preceded by a "Select" or "Select All" block.

### Move [X/Y/Z] axis, [min/center/max], to [value]



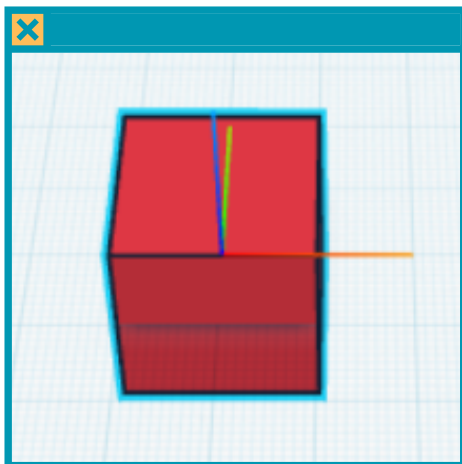
This block is used to precisely align a specific part of a shape (like its minimum edge, center, or maximum edge) to a specific coordinate along an axis (X, Y, or Z). Namely, it moves a specific anchor point (min, center, or max) of a shape to an exact position on the chosen axis.

# Example 1

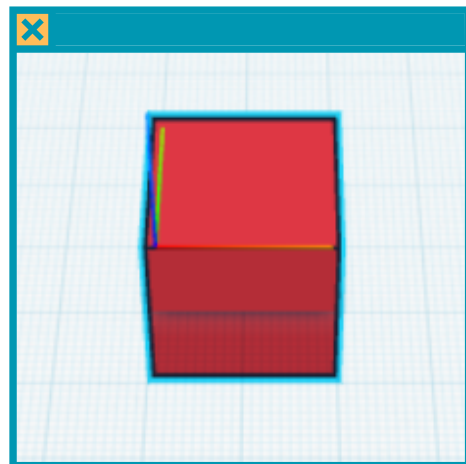


This example moves the left edge (Min) of the cube to  $X = 0$ , so the cube now sits from  $X = 0$  to  $X = 20$ .

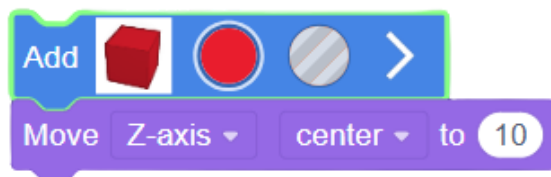
Before adding the move block



After adding the move block

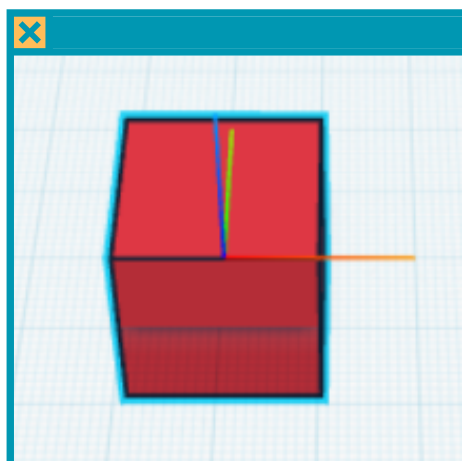


# Example 2

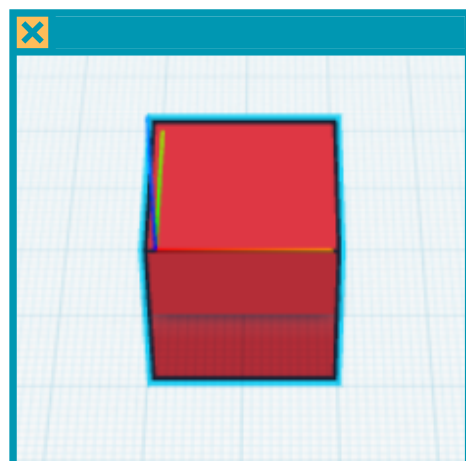


This example moves the center height of the shape to  $Z = 10$  mm. Therefore, if the shape is 20mm tall, the bottom is now at  $Z = 0$ , and the top at  $Z = 20$ .

Before adding the move block



After adding the move block

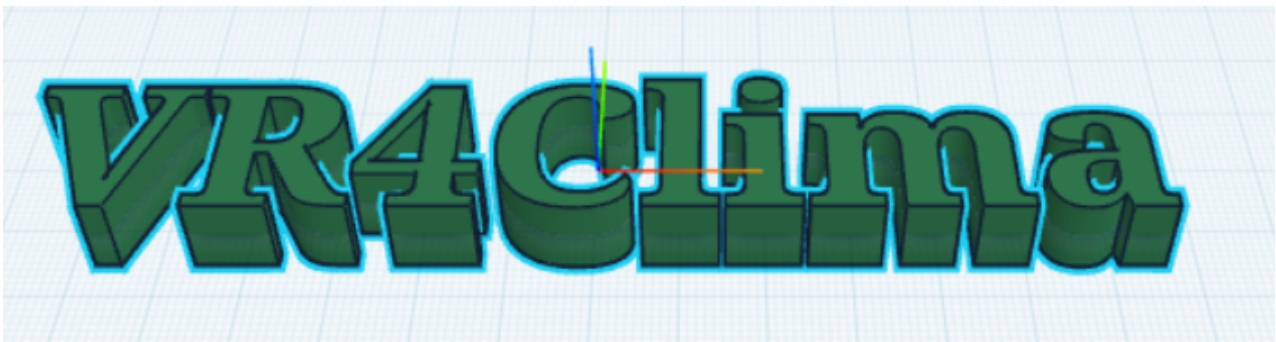


## Rotate around X/Y/Z axis by [degrees] from pivot [x, y, z]

It is used to rotate a shape or group around a specified axis and a custom pivot point in 3D space.

The block rotates the selected shape around one of the 3 axes (X, Y, or Z). The rotation is performed from a specific pivot point — not just the shape's center which gives you precise control over how and where the rotation happens.

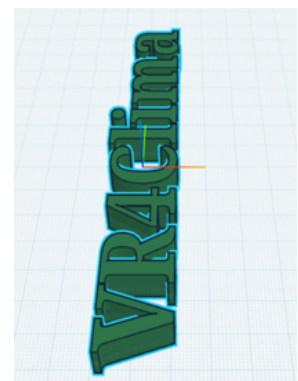
Let's work with our project's logo this time. We will insert Text "VR4Clima" in Green color in the workspace.



### Example 3



This example rotates the text 90 degrees around the Z-axis (vertical) as if it's spinning like a top.



### Scale [value]



The number of Steps represents the number of segments used to build the sphere's surface.

## Scale [X/Y/Z]



- ✓ center
- bottom center
- custom point

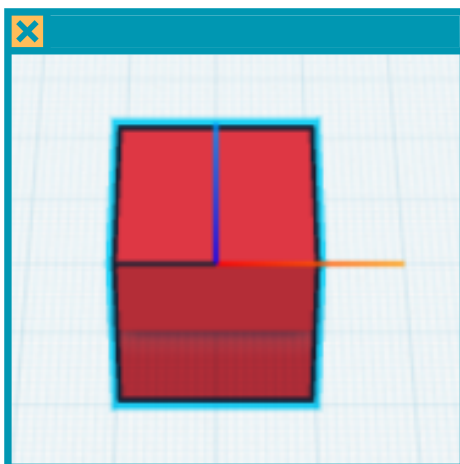
It is used to resize an object non-proportionally along one or more axes. It multiplies the original size of the selected shape by a scale factor you choose for each axis (X, Y, and Z). This allows you to make the object larger or smaller in specific directions.

## Example 4

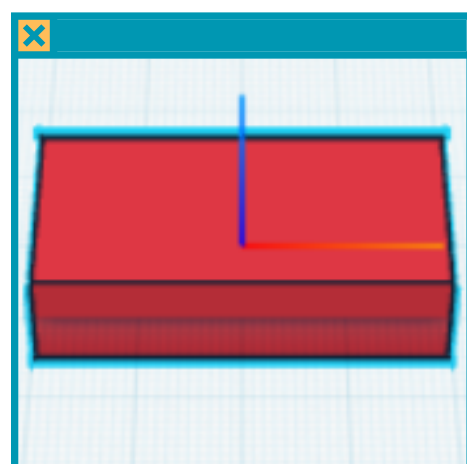


The originally inserted cube will be twice as wide (in the X-axis), have the same depth (Y-axis) and it will be half as tall (Z-axis).

Before scaling

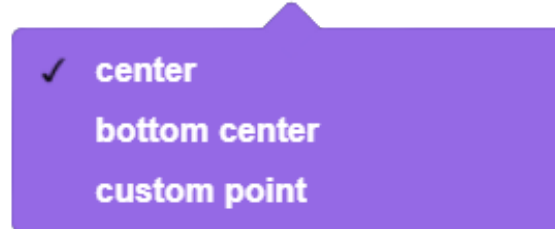


After scaling



Note: Scaling is relative to the object's original size, not absolute measurements.

## Set size [X, Y, Z]



It resizes a shape to exact dimensions along the X, Y, and Z axes, regardless of its original size. Unlike the “Scale” block (which multiplies dimensions), this block overwrites the current size of the shape with the specific values you provide. The size will be represented in mm.

## Copy



It creates a duplicate (copy) of an existing shape with a specific color or Hole, which you can then modify, move, scale, or group independently from the original.

## Set color to [color]



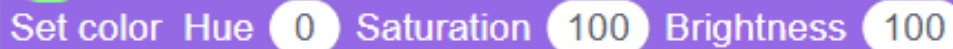
It sets the visual color of a shape in the Tinkercad 3D view. It doesn't affect the shape's geometry—just how it appears in the workspace.

## Set color to RGB [R, G, B]



It allows you to define the exact color of a shape by specifying its Red, Green, and Blue (RGB) values — each ranging from 0 to 255.

## Set color to HSB [Hue, Saturation, Brightness]



Set color Hue 0 Saturation 100 Brightness 100

It sets the color of a shape using the HSB color model, which defines:

- Hue – the type of color (0–360° on the color wheel, e.g., red, green, blue)
- Saturation – the intensity or purity of the color (0–100)
- Brightness – how light or dark the color is (0–100)

## Group selection

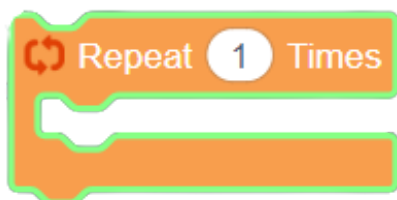


It combines all shapes above into one unified shape. This is equivalent to using the “Group” command in the regular Tinkercad editor.

## 2.1.3 Control

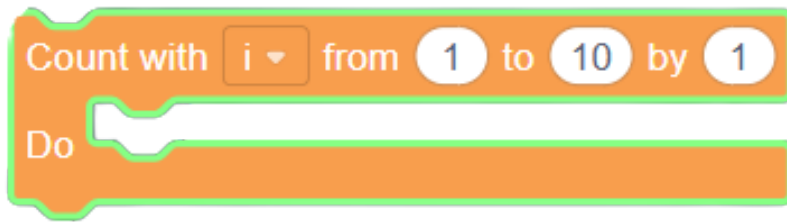
The Control group of blocks in Tinkercad Codeblocks plays a central role in managing program flow - how and when blocks are executed. These blocks allow you to add logic, repetition and timing to your 3D model creation.

### Repeat [number] times



It allows you to run a group of blocks multiple times, creating loops. Each time through the loop, the inner blocks are executed once. It is useful for patterns, arrays or repeated structures.

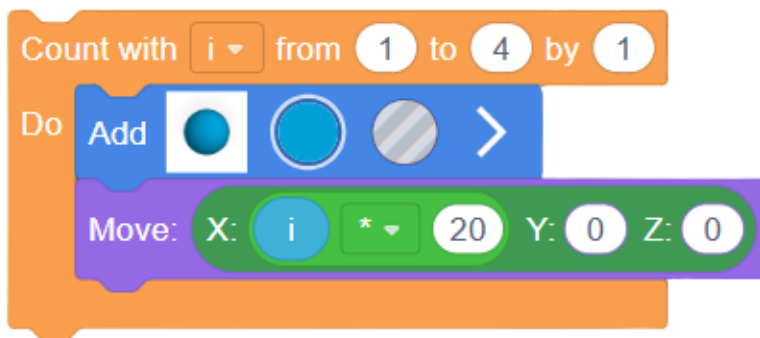
## Count with [variable] from [start] to [end] by [step]



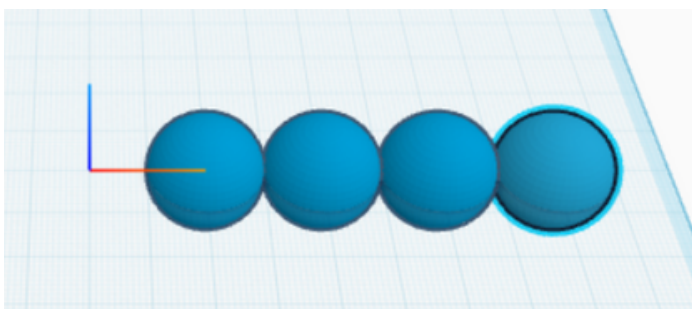
This block is a loop with a counter variable. It:

- Repeats a set of actions
- Uses a variable (like i) that increases or decreases with each repetition
- Lets you control the starting number, ending number, and the step size (positive or negative)

## Example 5



This will create 4 spheres spaced 20 mm apart in the X axis:

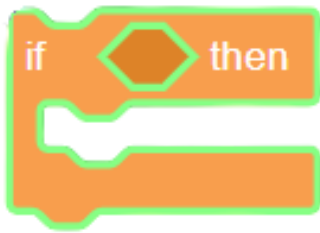


## Pause



It pauses the execution of the code so that you can visualize steps more clearly during animation. You have to click Space button to continue.

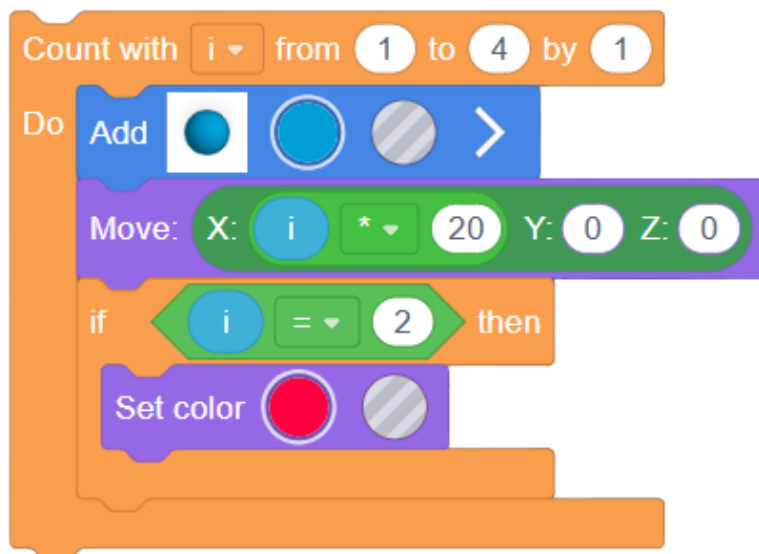
## If [condition] then



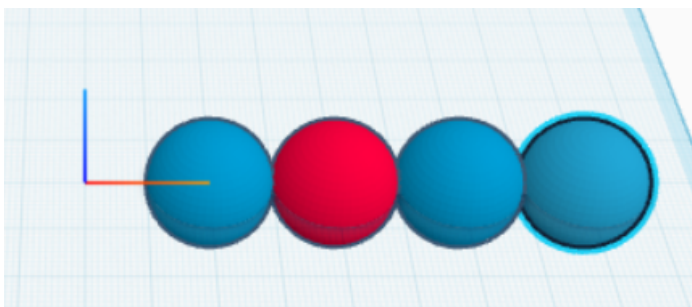
It allows your code to make decisions. It checks if a condition is true, and if so, it executes the blocks inside it. If the condition is false, it skips the enclosed actions.

## Example 6

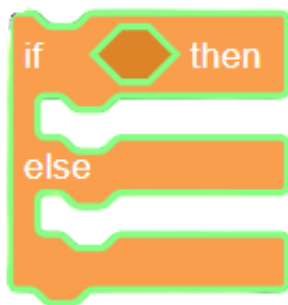
Extending the previous Example with an If Block



There will still be created 4 spheres, but the second one will be of color red:



## If [condition] then ... else ...

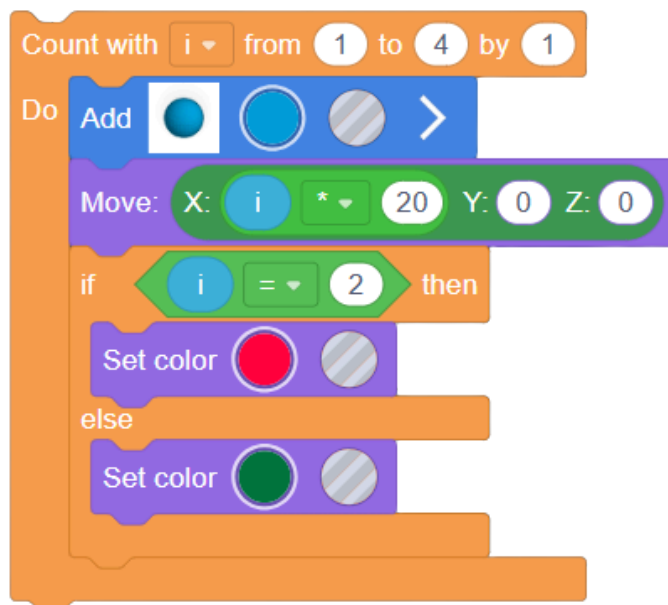


It's a decision-making control block that provides two alternative paths for the code to follow:

- "Then" block runs when the condition is true
- "Else" block runs when the condition is false

## Example 7

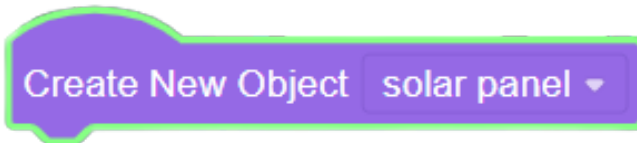
Extending the Example 5 with an If Then Else Block



There will still be created 4 spheres, but the second one will be of color red and the remaining ones will be of color green.

## 2.1.4 Legacy

### Create New Object



It begins a new shape sequence. Anything you add after this block will be treated as a separate object. It is useful when you want to group, modify or export objects independently.

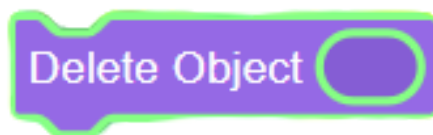
The name of your New Object will be also added in the Variables Group.

### Add Copy of Project



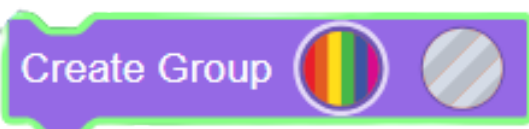
This block allows you to insert a duplicate of an object you previously created using the "Create New Object" block. It reuses that object in a new position, transformation, or group without rebuilding it step by step.

### Delete Object



It removes an object that was previously created with the Create New Object block. This is useful for cleaning up intermediate objects you no longer want to be part of the final design.

### Create Group



It takes all shapes used previously and combines them into a single grouped object. It functions like manually selecting shapes in Tinkercad and pressing the Group button, but in a coded sequence. You can also choose a color for the new group.

## 2.2 Using Variables

What is a variable? A variable is like a container that stores a value (number, text, etc.) that you can reuse or change in your code.

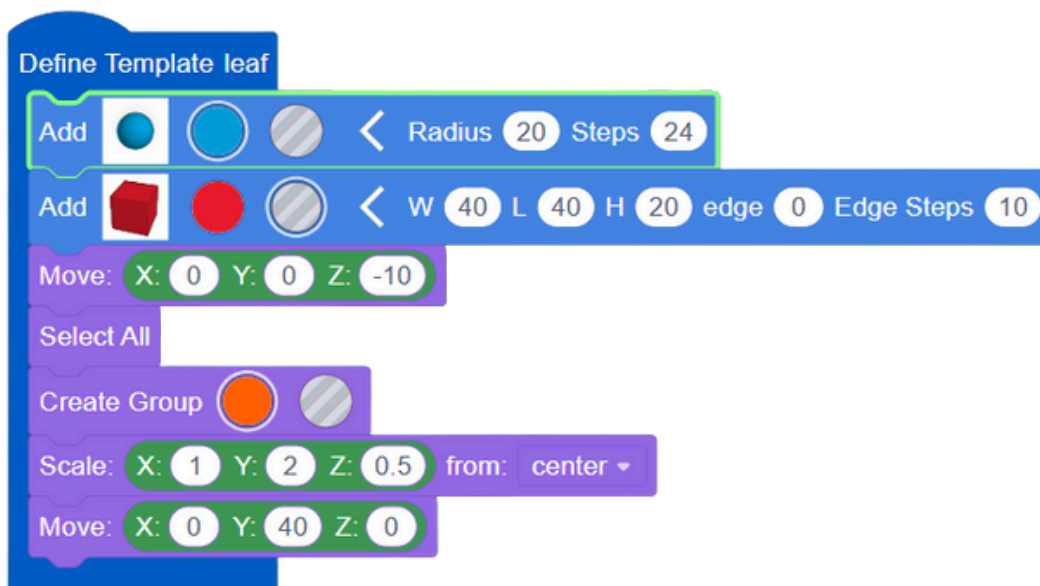
Reasons for using variables in Tinkercad Codeblocks:

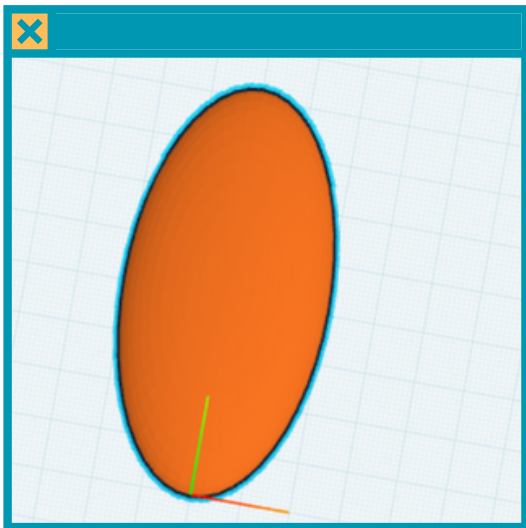
- Easier adjustments: Instead of changing the same number in many places, you change the variable once. (eg. If you use Height equal to 50 in many places of your code and for some reason you want to change Height to 20, by using variable Height you have to change it just once, instead of the number of times it appears at your code. This way you avoid mistakes as well.)
- Simplifies repetition: Variables help manage values that change in loops (e.g., positions, sizes).
- Use in calculations: Variables let you perform math operations to calculate sizes, angles, or positions dynamically
- Better readability: Giving a name to a value (e.g., wheelRadius) makes your code easier to understand and update.

Examples with the use of variables in Tinkercad Codeblocks:

### Example 8

We will shape a simple flower by creating a leaf and using it as template. The flower will be formed by quadrupling the leaf:

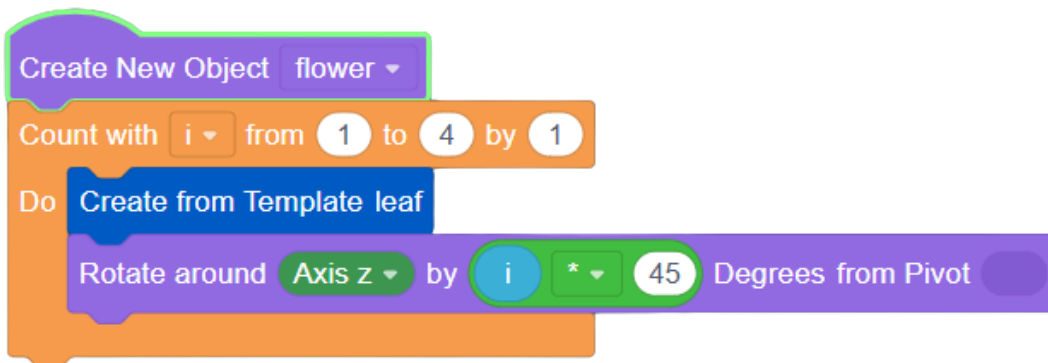
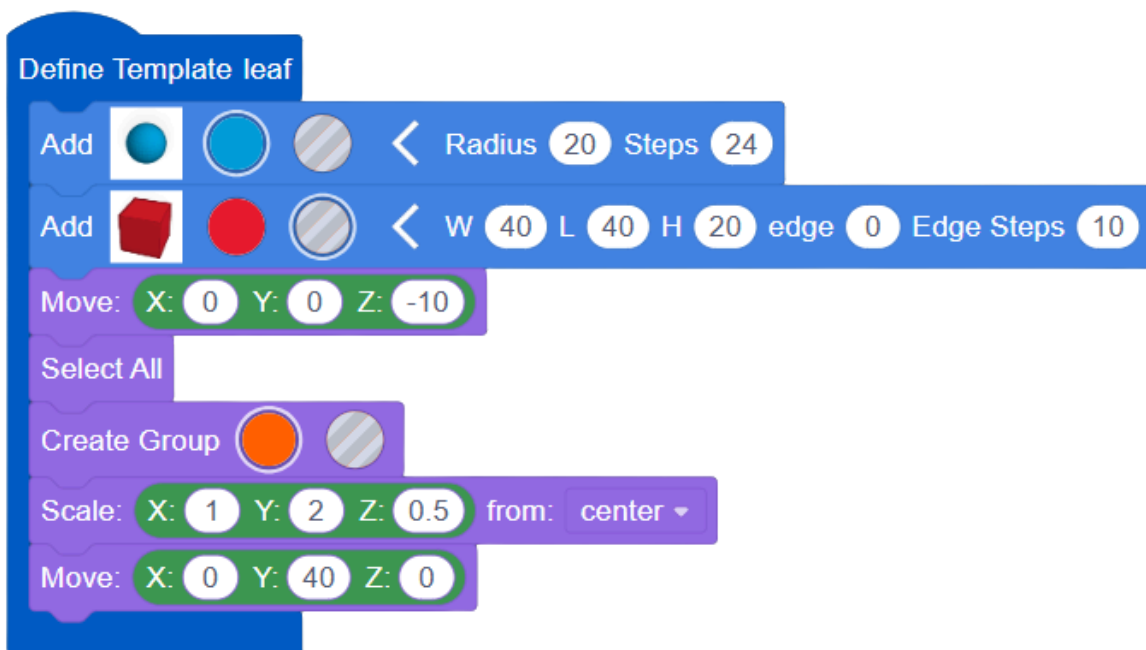




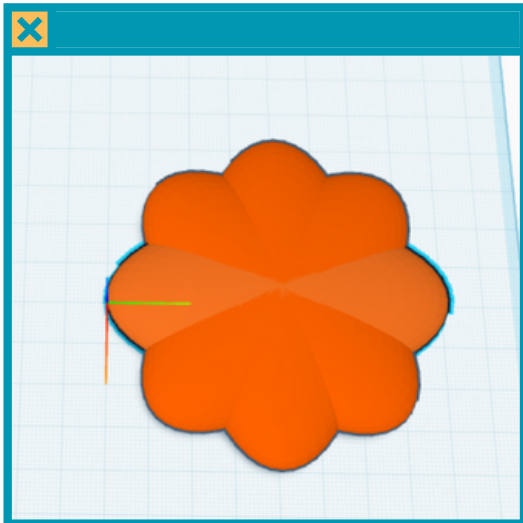
For creating the leaf we use a sphere, then add a hole box to delete the lower half of it, then we change dimensions by scaling it.

This is what we get:

Then we will use the Count with loop to quadruple the leaves:



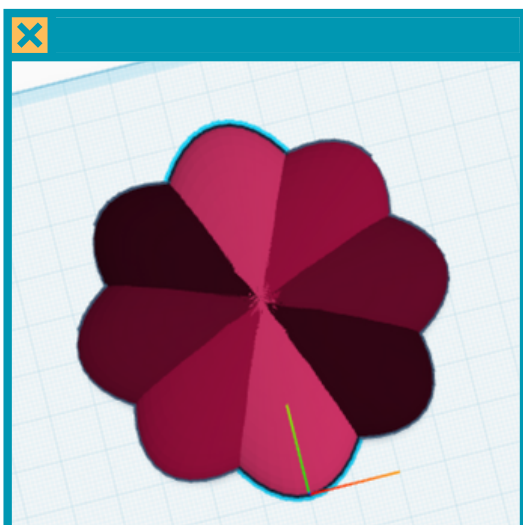
This is what you get:



If you feel like experimenting with colors, here's an example:

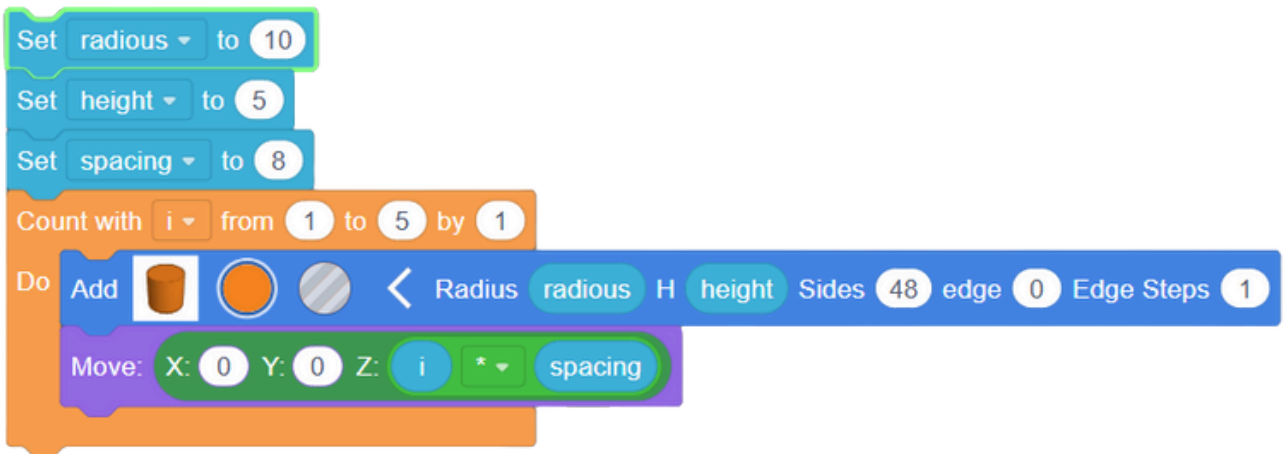
```
Create New Object flower ▾  
Count with i ▾ from 1 to 4 by 1  
Do Create from Template leaf  
  Rotate around Axis z ▾ by i * ▾ 45 Degrees from Pivot  
  Set color Red 50 * ▾ i Green 5 * ▾ i Blue 20 * ▾ i
```

And you will get:

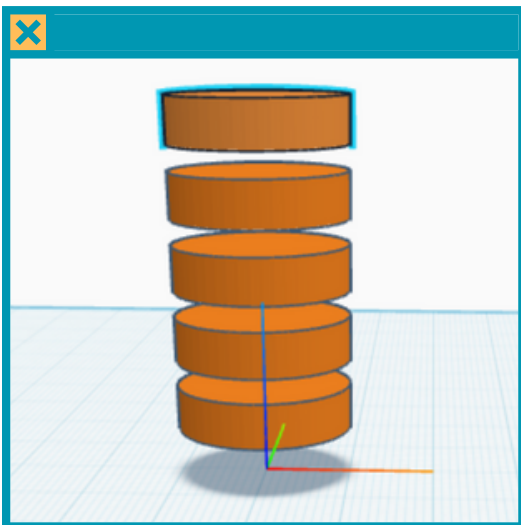


## Example 9

Let's create a stack of 5 cylinders where you can easily adjust the radius, height, and spacing using variables:



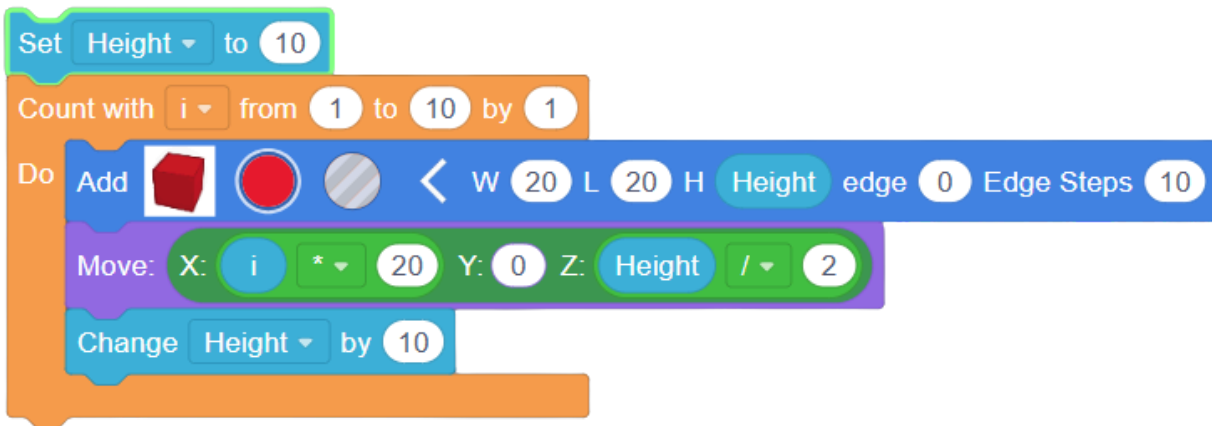
By using the above code you get this:



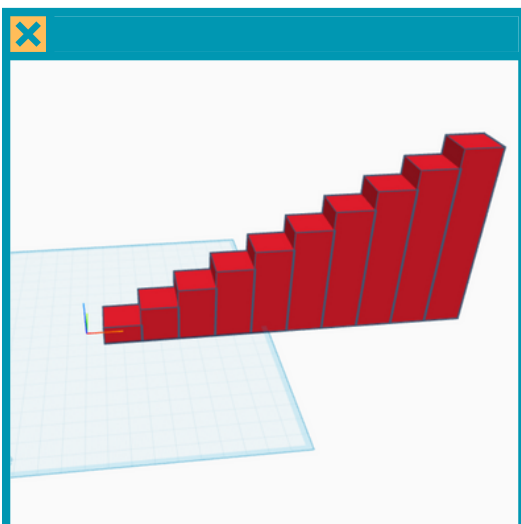
Now you can experiment by changing the values of Height, Radius and Spacing variables.

## Example 10

Let's create a stair with 10 steps:



First you create the variable Height in the Blocks Panel (Create New Variable option). You have 2 variables (Height and i, which is self-declared when you use the Count Loop). You give variable Height an initial value using the Set Variable Block. In the Move block you change the x to  $i \cdot 20$  so that each box is placed next to the other and the z to  $\text{Height}/2$  so that each box stand on the workplane. Above you can see what you will get by using the above code:



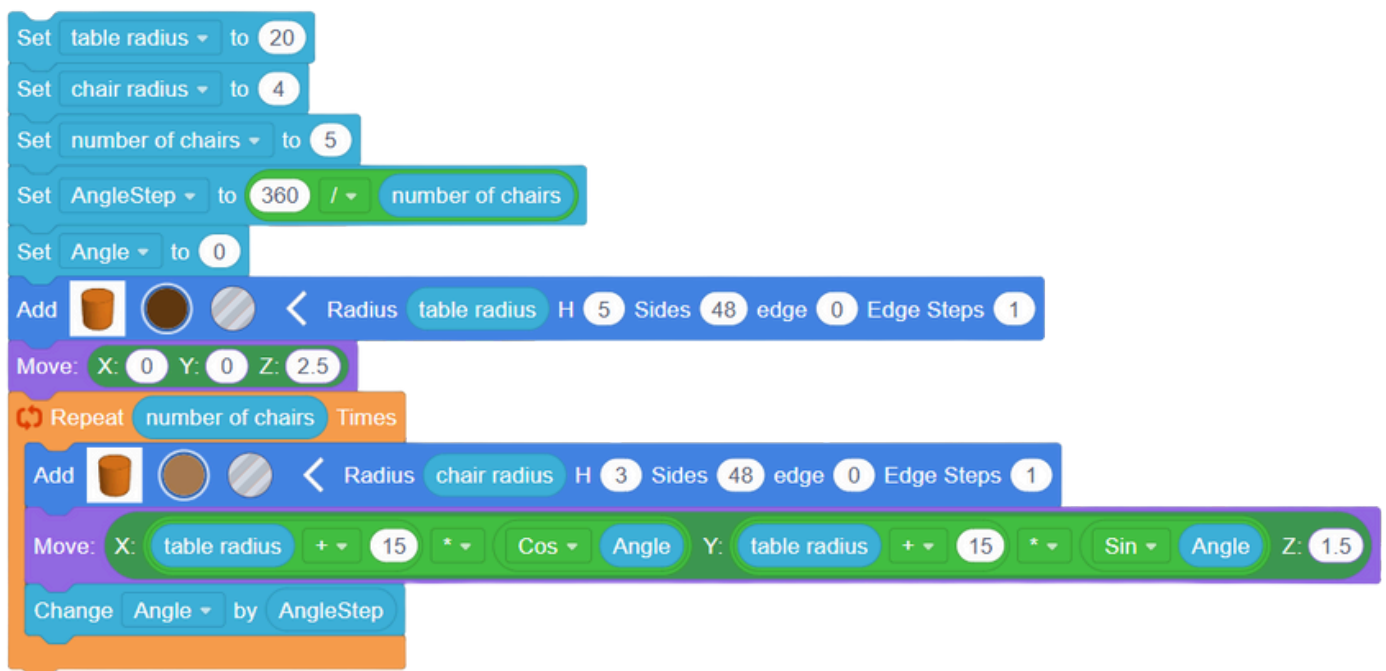
## 2.3 Mathematics for Codeblocks

Using Tinkercad Codeblocks it becomes evident that math isn't abstract. On the contrary, it directly controls shape and symmetry in 3D models.

If you want to create objects placed in circular or angled patterns, trigonometry will help you calculate their positions. In 3D design, this allows you to create symmetry, curves, spirals, gears, and star-like shapes. Sine and cosine ensure mathematical precision in the position of objects.

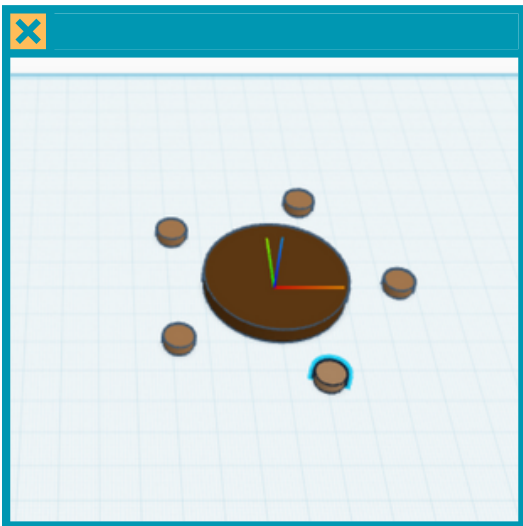
### Example 11

Let's create a table with 5 chairs around it. We will use the following code:

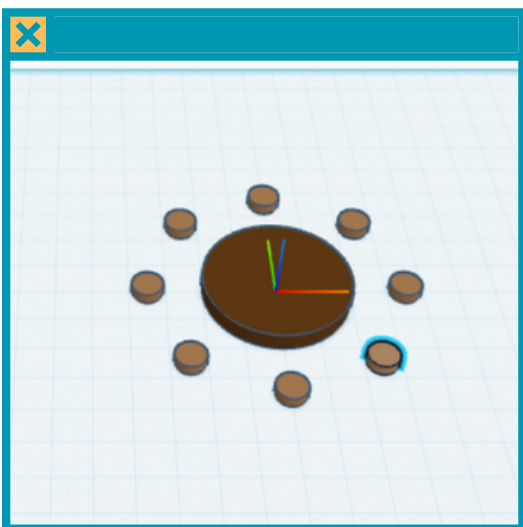


```
Set table radius to 20
Set chair radius to 4
Set number of chairs to 5
Set AngleStep to 360 / number of chairs
Set Angle to 0
Add [Cylinder] Radius table radius H 5 Sides 48 edge 0 Edge Steps 1
Move: X: 0 Y: 0 Z: 2.5
Repeat number of chairs Times
  Add [Cylinder] Radius chair radius H 3 Sides 48 edge 0 Edge Steps 1
  Move: X: table radius + 15 * Cos Angle Y: table radius + 15 * Sin Angle Z: 1.5
  Change Angle by AngleStep
```

And that is what we get:



The use of variables in the code makes any change in the number of chairs very easy. If for example you decided to change the number of chairs to 8, the only thing you have to change in the code is the value of the variable “number of chairs”. And you will get the following:



## PART 3

# STEP-BY-STEP PROJECTS

## 3.1 Project 1 – Creation of a flower

### Objectives

By the end of this activity, students will:

- Understand how to use loops in Tinkercad Codeblocks.
- Apply angles and rotations to create patterns.
- Practice parametric design by changing variables (like number of sides).
- See how mathematics (geometry & symmetry) helps in 3D design.

### Materials / Setup

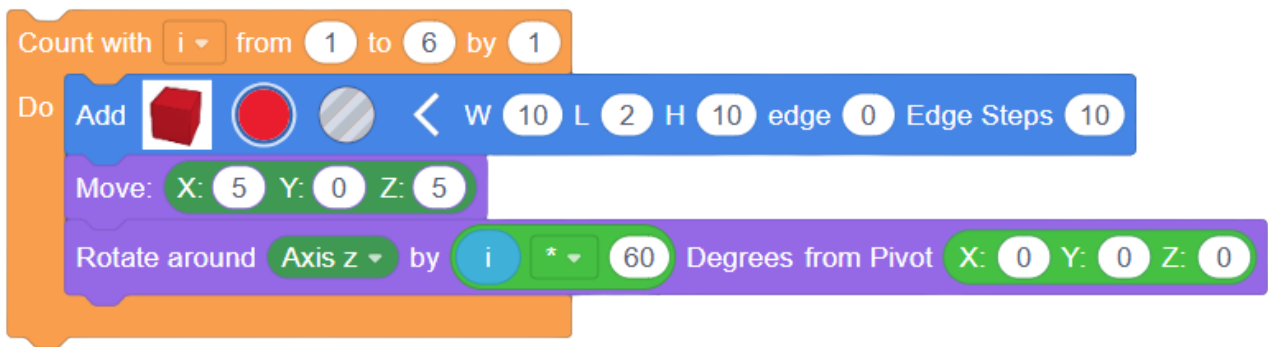
- A computer with internet access.
- A free Tinkercad account.
- (Optional): projector to demonstrate steps.

### Step-by-Step Instructions with Screenshots

- First you add a cube in your design with dimensions (10,2,10)



- From the Control group, drag a count i times block and place the Add Cube inside the loop. Add a Move and a Rotate around axis z by 60 degrees block.



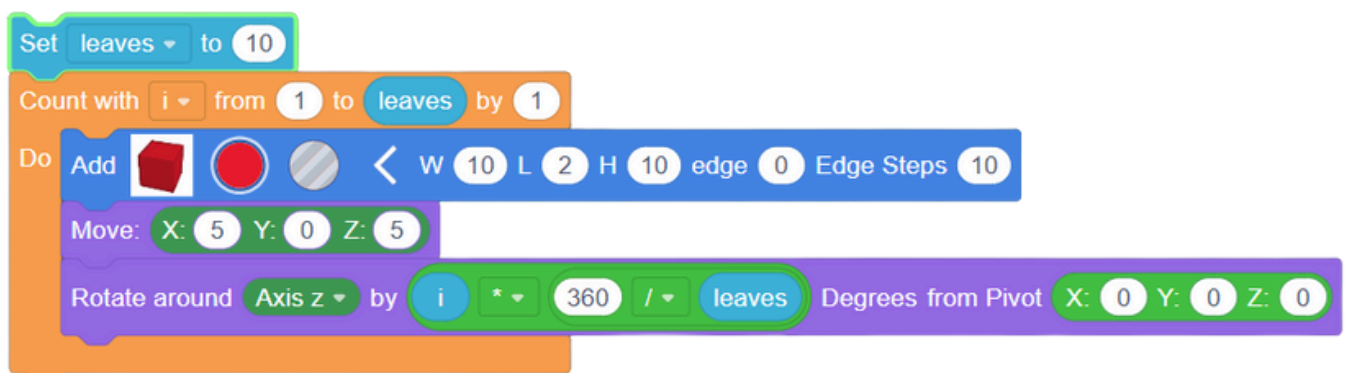
The Add Cube adds your cube in the workplane.

Since the center of your cube will be placed in the (0,0,0) point, you add the Move block so that it lifts it up by half of its Height (z:5 in this example) so that it “sits” on the workplane and you also move it in the x axis so that its left side touches the starting point (x:5 in this example).

You want to create a flower with 6 leaves, so each one will have an angle of  $360/6=60$  with the next one. During the first loop ( $i=1$ ) the angle will be 60, during the 2nd loop ( $i=2$ ) the angle will be 120 etc.

Tinkercad will rotate your cube from its center point unless you state otherwise using Pivot. In this example we instruct Tinkercad to rotate the cube from the starting point.

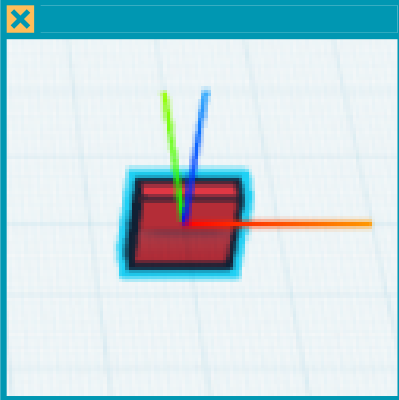
- Let’s explore the way to make flowers with various leaves numbers, just by changing one parameter: I create a number Variable named “leaves” and set it to 10. Then we change the Rotate around z axis to  $360/\text{leaves}$  instead of 60. This way you will get a flower with 10 leaves. You also change the count loop from  $i$  to ‘leaves’ .



Now, whenever you want to change the leaves of your flower, you only need to set the variable “leaves” to the number you wish and the magic will happen!

## Codeblock Visual Flow

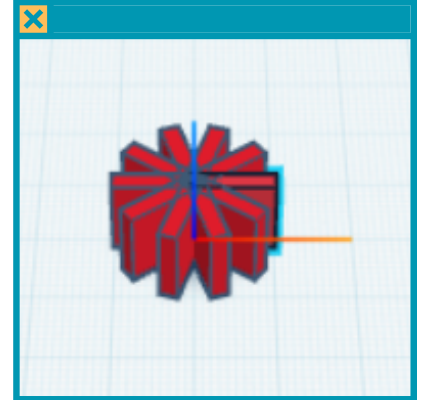
Placement of the first cube.



Flower with 6 leaves



Flower with 10 leaves



## Extensions / Challenges

If you wish to practice more, here are some challenges for you:

1. **Shape Swap:** Replace the cube with a cylinder, pyramid, or star.
2. **Color Patterns:** Use set color red green blue block inside the loop to make each tile a different color. Change the values of red green blue depending on the value of  $i$
3. **3D Layering:** After each repeat, also move up by 2mm → creates a spiral tower.

## 3.2 Project 2 – Creation of a name tag

### Objectives

By the end of this activity, students will:

- Learn to use the Text block in Tinkercad Codeblocks.
- Apply variables to create customizable name tags.
- Practice alignment and grouping in code-based design.
- Export their designs for 3D printing.

### Materials / Setup

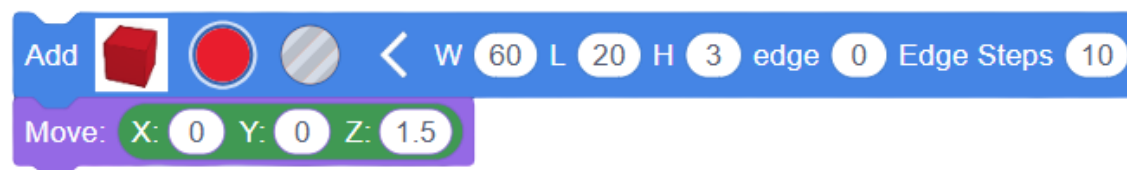
- A computer with internet access.
- A free Tinkercad account.
- (Optional): 3D printer and filament.

### Step-by-Step Instructions with Screenshots

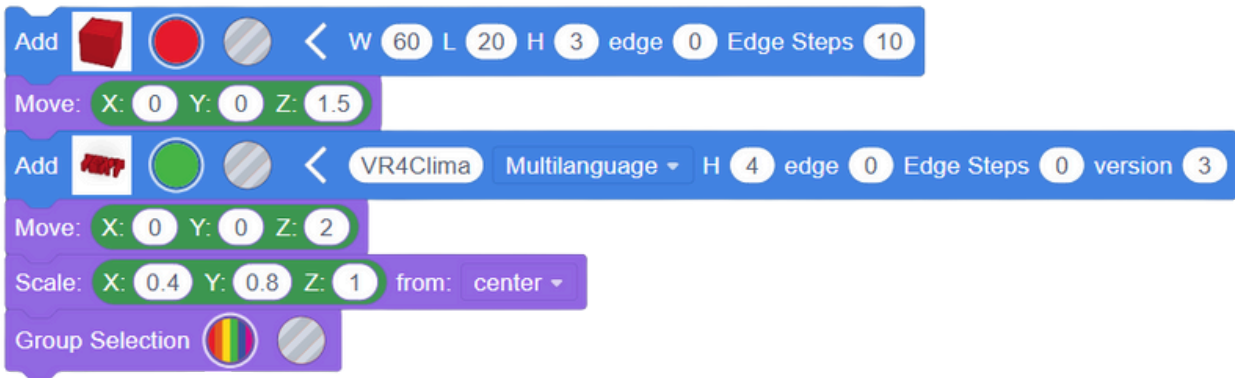
- First you add a cube in your design with dimensions (60, 20, 3) to create a flat rectangle. This will be the base of the name tag.



- From the Modify group, drag a Move block to lift the cube, so that it “sits” on the Workplane.



- Add a Text box with your name and Height 4 or 5 (it has to be a bit higher than the base of the name tag). Then, from the Modify group drag a Move box to move your text up by half its height. Then, use the Scale box to scale the text so that it fits in the base of the name tag. You can experiment with different scalings in the x and y axis until the result satisfies you. Finally, use the Group selection box to group your design.



- If you wish to 3d print your name tag, you will need to export your design from Tinkercad. Click on the Share button on top right of your screen.



Then you will have to choose the format you wish. For most 3D printing, an STL (STereoLithography) file is the best format, as it's widely compatible with printers and software, especially for simple, single-color models.



### Codeblock Visual Flow

Placement of the text box on top of the base of the name tag.



Name tag after scaling.



Name tag after grouping.



## Extensions / Challenges

If you wish to practice more, here are some challenges for you:

1. **Change Colors:** Use set color block (e.g., base = blue, text = white).
2. **Add Hole for Keychain:** Place a cylinder, resize (diameter = 5mm), and subtract it from one corner.
3. **Shape Variations:** Instead of a rectangle, use a rounded box, star, or oval for the base.

# 1.3.3 Project 3 – Creation of a geometric spiral

## Objectives

By the end of this activity, students will:

- Understand how loops and incremental changes create complex patterns.
- Apply rotation and movement together.
- Experiment with variables to control the number of steps and spiral size.
- See how coding can generate mathematical art.
- Use rotation and repeat loops to create patterns.

## Materials / Setup

- A computer with internet access.
- A free Tinkercad account.

## Step-by-Step Instructions with Screenshots

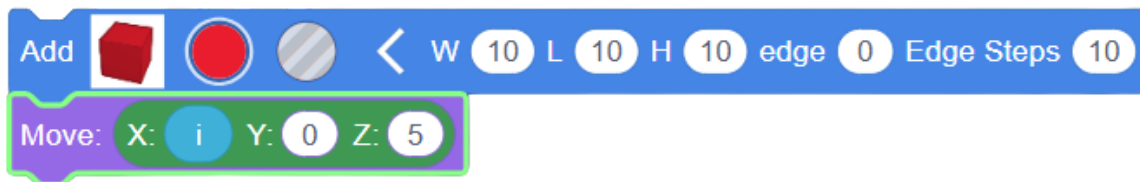
- First you add a cube in your design with dimensions (10, 10, 10) to create the first part of your star.



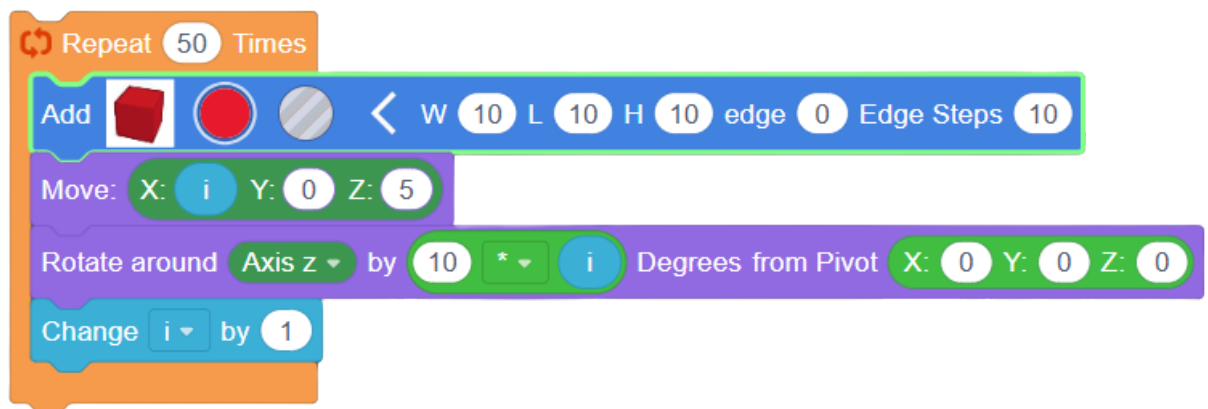
- From the Variables group, create a variable named i to use with a loop.



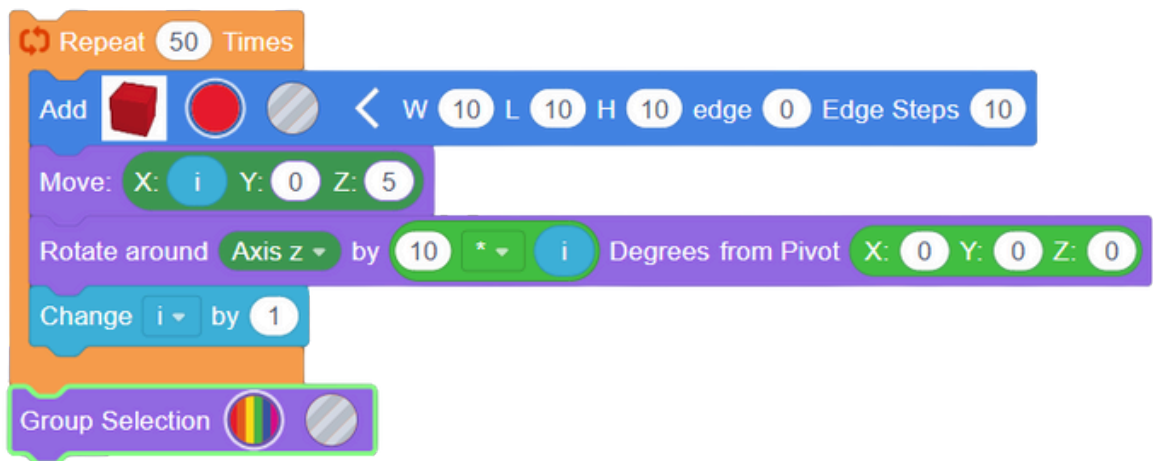
- From the Modify group add a Move block, so that your cube “sits” on the Workplane and each time it is moves in the x axis a bit. (This is going to be placed inside a Repeat loop)



- Add the Repeat Loop. Each time a cube is inserted, it will be moved in the z axis to create the spiral effect.

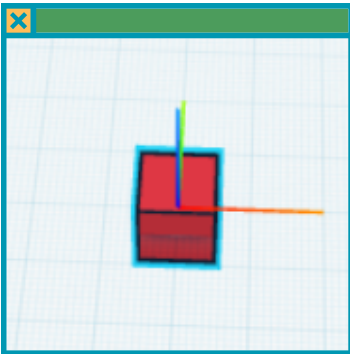


- Add the Repeat Loop. Each time a cube is inserted, it will be moved in the z axis to create the spiral effect.

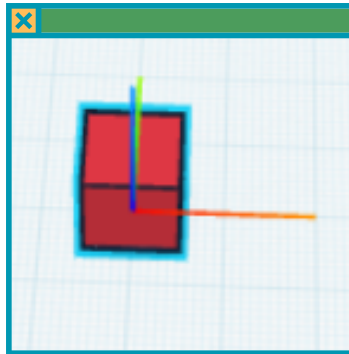


## Codeblock Visual Flow

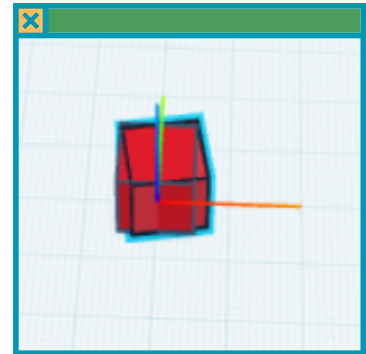
Placement of the first box on the Workplane.



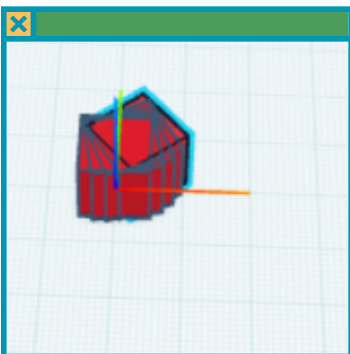
Reposition of the box so that it “sits” on the Workplane and its base is at the starting point.



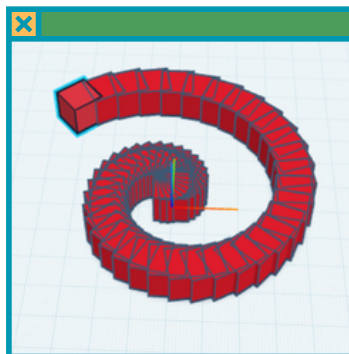
Second cube.



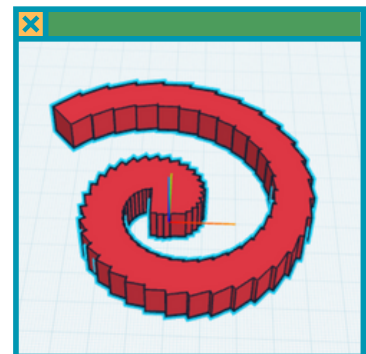
After adding a couple of cubes.



After adding 50 cubes.



End result.



## Extensions / Challenges

If you wish to practice more, here are some challenges for you:

1. **Change Rays:** Modify number of rays
2. **Colorful Star:** Set random color for each ray.
3. **Layered Ornament:** Add multiple stars stacked at different heights.
4. **Snowflake Design:** Replace boxes with thin rectangles or pyramids for variety.
5. **Change Shapes:** Try spheres, pyramids, or text instead of cubes.
6. **Color Gradients:** Add a set color block inside the loop, changing hue each step.
7. **Double Spiral:** Run a second spiral rotated 180° for symmetry.
8. **3D Spiral Tower:** Add a move up by 1 in each step to create a vertical helix.

## PART 4

# EXPORTING AND TROUBLESHOOTING

## 4.1 Exporting and Printing 3D Models

Next step after creating a 3D design with Tinkercad Codeblocks is to bring your model into the real world. Exporting allows you to save your digital design in a format that can be used by:

- 3D printers (to make physical objects)
- Other CAD software (to refine or combine designs)
- Game engines or AR/VR environments (for digital experiences).

Before exporting your project:

- Make sure your code is complete and click Run so that the final shape appears in the Codeblocks workspace.
- Check that shapes are grouped and aligned.

### Export Options:

In Codeblocks, click the Export button. You'll see several file formats:

- **.STL (Stereolithography)**: The most common format for 3D printing. It stores geometry but no color.
- **.OBJ (Object File)**: Supports both geometry and color; useful if you want multi-material printing or import into design software.
- **.GLTF (GL Transmission Format)**: Web-friendly 3D model format, supports color and textures.
- **.SVG (Scalable Vector Graphics)**: For 2D laser cutting or CNC machines (only for flat designs).

## 4.2 Common Troubleshooting Tips

- *My Shape Is Too Small / Invisible*

Check scaling values: Shapes smaller than 1 unit can be hard to see. Reset the Set Size or Scale block to larger numbers (e.g., 10, 20, 30). Zoom in/out with the mouse wheel to make sure it's not hidden.

- *Objects Overlap or Don't Align*

Check position values: Use the Move block carefully (positive = right/up/forward, negative = left/down/back). For precision, use variables and math (e.g., move by `boxWidth + 2`). Turn on the grid and zoom out to see the layout clearly.

- *Rotation Looks Wrong*

Rotation happens around a pivot point (usually the object's center). Check if you rotated around the correct axis (X, Y, or Z). Try 90° increments first to understand orientation.

- *My Code Runs Too Fast*

Add a Pause block from the Control group to slow down the execution and observe steps.

- *Repetition Creates Too Many Objects*

Check loops: If you used a Repeat block, make sure the number is not too high (e.g., "100" can freeze your computer). Test with a small repeat (e.g., 3–5) before increasing.

- *Color Didn't Change*

Use Set Color before running the program. Make sure grouping doesn't override colors (grouped objects may take a single color).

- *Shapes Disappear After Grouping*

If you group a hole with a shape, it cuts material. Check if you accidentally set an object to Hole instead of Solid.

- *My Design Looks Messy*

Organize code blocks in a logical order (indent if needed). Rename variables with clear names (e.g., boxHeight instead of var1). Use comments (the Comments block) to explain what sections do.

- *Computer Becomes Slow or Freezes*

Too many shapes or repeats can overload the browser. Reduce the number of objects. Close other browser tabs while working. If stuck, refresh the page – Tinkercad usually auto-saves.

## PART 5

# APPENDICES

## 5.1 Tinkercad Keyboard Shortcuts

View and Navigation	
Right Mouse Drag	Orbit (rotate the view)
Shift + Right Mouse Drag	Pan (move view)
Scroll Wheel	Zoom in/out
F	Focus view on selected object
Object Movement & Placement	
Arrow Keys	Move object on X/Y plane (1 step at a time)
Shift + Arrow Keys	Move object in larger steps (10 units)
Ctrl + Arrow Keys (Cmd on Mac)	Move object vertically (up/down)
Rotation	
Rotate Handles (with mouse)	Free rotation
Shift + Rotate	Snap rotation to 45° increments
Scaling and Size	
Shift + Drag Corner Handle	Scale object proportionally
Alt (Option) + Drag Corner Handle	Scale from center
Copy, Paste, Duplicate	
Ctrl + C	Copy
Ctrl + V	Paste
Ctrl + D	Duplicate and repeat last move/rotate action

Delete & Undo	
<b>Delete</b>	Delete selected object
<b>Ctrl + Z</b>	Undo
<b>Ctrl + Y</b>	Redo
Align & Group	
<b>L</b>	Align selected objects
<b>Ctrl + G</b>	Group
<b>Ctrl + Shift + G</b>	Ungroup
Helpers	
<b>R</b>	Ruler tool
<b>W</b>	Workplane tool
<b>M</b>	Mirror selected object
Snap Grid	
<b>Ctrl + Shift + Up/Down Arrow</b>	Adjust snap grid step size (finer or coarser moves)

## 5.2 Glossary of Terms

### **Align**

A function that positions objects relative to each other along the X, Y, or Z axis.

### **Axis (X, Y, Z)**

Reference lines in 3D space.

X-axis = left–right

Y-axis = front–back

Z-axis = up–down

### **Block**

A visual coding element that represents an instruction (e.g., “Add Box,” “Move,” “Rotate”).

**Boolean Operations**

Combining objects using grouping or subtraction (holes) to create new shapes.

**Camera**

The point of view used to orbit, pan, and zoom around the 3D workspace.

**Codeblock Visual Flow**

The sequence of blocks connected from Start to create a program.

**Control Blocks**

Blocks that manage logic and repetition (e.g., “Repeat,” “If–Then”).

**Duplicate**

Making an exact copy of an object or code sequence.

**Edge Steps**

A setting in some shapes (e.g., cube, cylinder) that adjusts the smoothness of rounded edges.

**Group / Ungroup**

Combining multiple objects into one, or separating them again.

**Hole**

An object property that subtracts its shape from other solids when grouped.

**Loop**

A control structure that repeats code (e.g., “Repeat 5 times”).

**Move**

Shifts an object along the X, Y, or Z axis.

**Object**

Any 3D shape created in the workspace (box, sphere, cylinder, etc.).

**Parameter**

A value that defines shape properties, such as width, height, radius, number of sides.

**Pause**

A control block that stops execution for a set time before continuing.

**Pivot Point**

The center around which an object rotates.

**Repeat**

A block that repeats instructions multiple times to save effort.

**Rotate**

Turns an object around an axis (X, Y, or Z) by a set number of degrees.

**Scale**

Resizes an object by multiplying its dimensions (relative resizing).

**Set Size**

Directly sets an object's width, height, and depth (absolute resizing).

**Shape Generator**

Any block that creates a 3D primitive (box, cylinder, sphere, etc.).

**Snap Grid**

The step size used when moving objects with the mouse or keyboard.

**Solid**

An object property that makes a shape visible and printable.

**Start Block**

The first block in a Codeblocks program; all sequences begin here.

**Variable**

A named container that stores values (e.g., "radius = 10") to make designs dynamic and reusable.

**Workplane**

The 3D workspace surface where shapes are placed and manipulated.

## 5.3 Resources for Further Learning (Websites, Videos, Communities)

### **Official Tinkercad Resources**

- [Tinkercad Codeblocks Learning Page](#) – Official tutorials, beginner projects, and guides from Autodesk.
- [Tinkercad Blog](#)

### **Video Tutorials**

- [Autodesk Tinkercad YouTube Channel](#) – Short videos covering basics, classroom tips, and design ideas.
- [Teachers & Makers on YouTube](#) – Maker’s Muse and PrintLab often showcase creative 3D projects; Search for “Tinkercad Codeblocks Tutorial” to find classroom walkthroughs.

### **Teacher & Classroom Guides**

- [Autodesk Design Academy](#) – Free lesson plans and educator resources.
- [Instructables Tinkercad Projects](#) – Step-by-step projects created by the community.

### **Online Communities & Forums**

- [Tinkercad Community Forum](#) – Ask questions, share projects, and get help.
- [Reddit – r/Tinkercad](#) – A place where students, hobbyists, and teachers post projects and tips.
- [Facebook Groups \(e.g., Tinkercad Teachers, 3D Printing for Education\)](#) – Active spaces to exchange classroom practices and codeblocks projects.

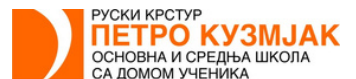
This project has been funded with support from the European Commission. This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.  
Project Number: 2023-1-EL01-KA220-SCH-000155463



Funded by  
the European Union



CENTER FOR  
THE  
PROMOTION  
OF SCIENCE





Here you can find [Official Guide to Tinkercad Codeblocks](#) for further questions.