



Code a Wiggly Snake



by EdgertonCenter

CREATOR(S): Diane Brancazio and Anya Timchenko, MIT Edgerton Center

Create and 3D Print Articulated Snakes

Lesson Overview

Design and 3D print an articulated snake, of any length! “Articulated” means that you print the entire piece at once, and when you remove it from the build plate the pieces are hinged together - no assembly required! Use Codeblocks, the Autodesk coding environment in Tinkercad to create a head, body piece, and tail, then multiply the body piece to change the length of the snake. Customize any piece of the snake to make it look friendly, dangerous, or just unique.

Learn a simple coding platform while creating your snake, and see how basic coding principles can be useful in designing objects with repetitive features.

Essential Question:

How can coding be used in design to make standard parts and repeating shapes?

What are the basic principles of coding and how are they useful in design?

Skills Taught:

- 3D modeling of features in articulated objects
- Basic coding principles
- Coordinate planes with X, Y, Z axes

Time Required: 1 - 2 Hours

Materials Needed:

- Codeblocks workspace in Tinkercad
- 3D printer and filament

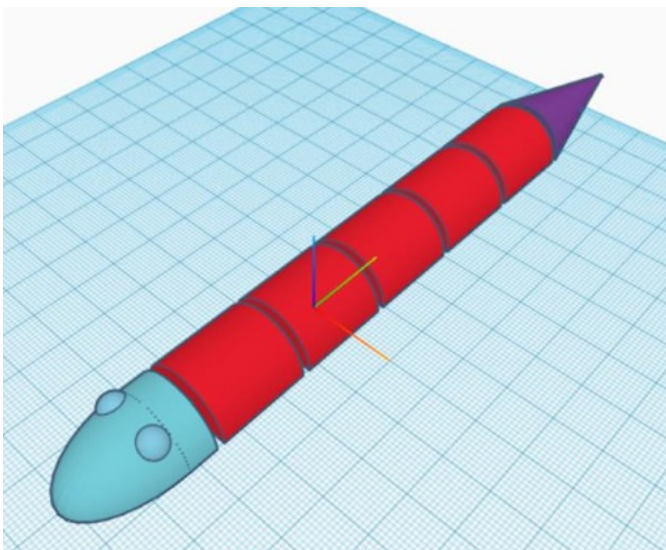


Step 1: Understand the Project and Explore Codeblocks

Context

3D modeling and 3D printing are already fun, engaging, hi-tech activities. One of the coolest features of 3D printing technology is that objects can be created with interior voids that cannot be fabricated using traditional methods. A fascinating application of that capability is the creation of articulated objects that have hinged parts and are moveable right off the build plate.

In this activity, designers will experiment with coding and use geometric and mathematical concepts while creating a fun toy. They will generate code for the articulated snakes based on example code in this tutorial and can customize the pieces as much as they want. Designers choose the number of body pieces in the snake. The only constraint is the size of the print bed on the 3D printer.



As 3D modeling and printing technologies become more reliable, less expensive, and easier for beginners to use, they become useful tools for practicing other skills such as coding, math, and logic. This activity is based on Tinkercad Codeblocks where users “design with code” - making 3D models using draggable blocks of code.

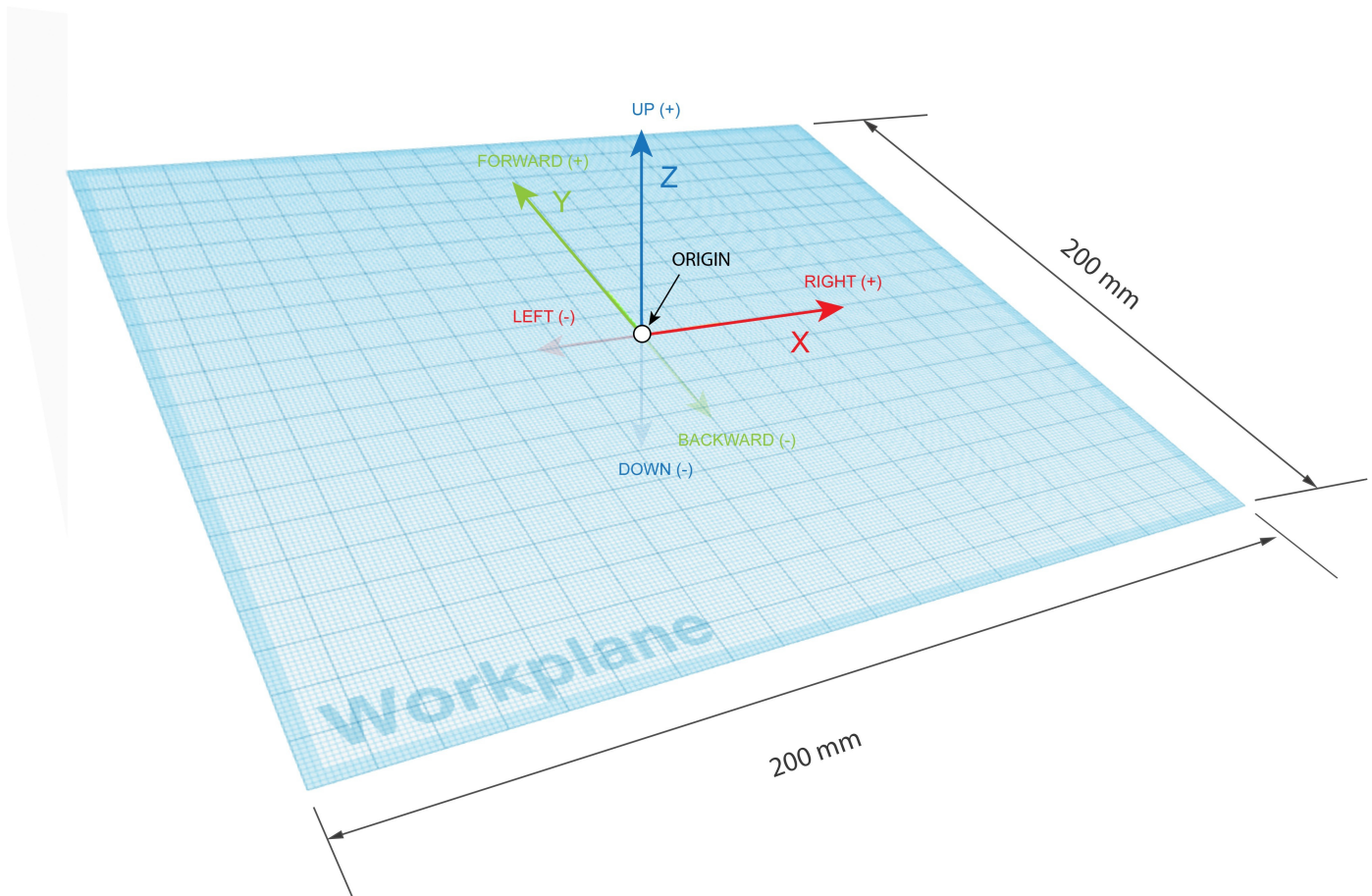
Features of Codeblocks:

- There are computation blocks for common math functions including: () + - * /.
- Objects can be moved around the design space on standard X, Y, Z axes, and can be scaled along those axes as well.
- Designers can create and set variables for objects and for mathematical quantities and can call them in the math computation blocks.
- Count and Repeat loops are available for replicating objects easily
- All the usual Tinkercad 3D modeling capability of building with primitive shapes that can be solid or holes, and grouping to create new shapes.

The first step is to get familiar with the block-based programming environment in Tinkercad Codeblocks. Watch the Quick Start Guide on the Codeblocks site to learn how to snap and unsnap blocks. Explore and practice with these investigations:

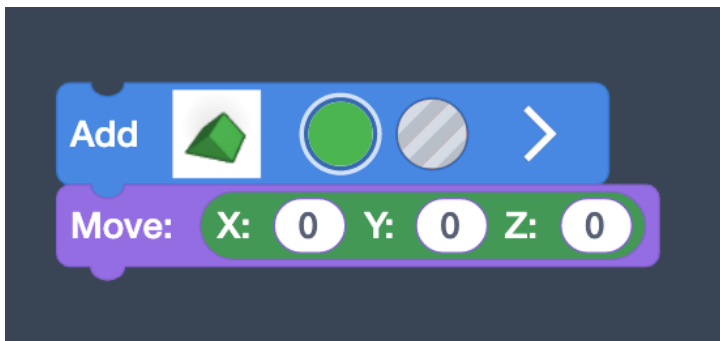
1. The Workplane and the Origin

- a. The Workplane measures 200 mm x 200 mm. The Origin is the center point where $X = 0$, $Y = 0$, and $Z = 0$
- b. Objects come into the Workplane with their center points on the Origin. To set objects to specific positions, it is necessary to shift them in the X, Y, and Z directions. A convenient way to do that is to move the object by half its size in the X, Y, and or Z direction. Some objects allow you to set those Width, Length, and Height dimensions when they are created, others you can determine by moving them around.



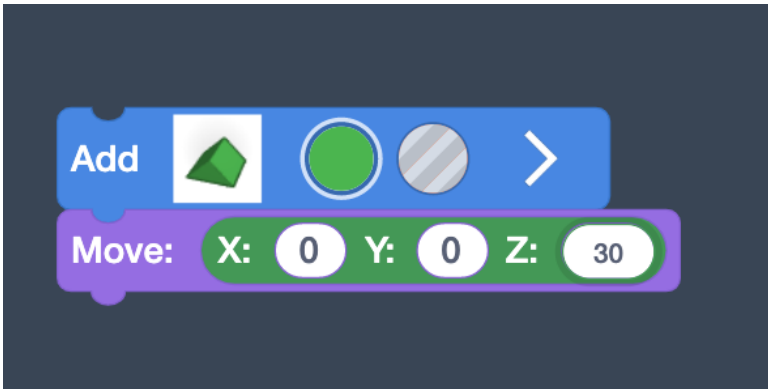
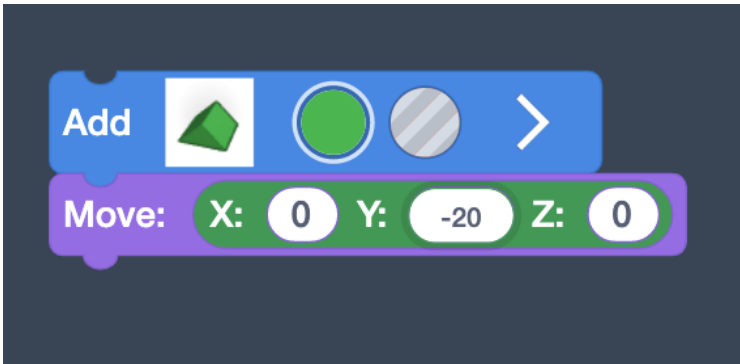
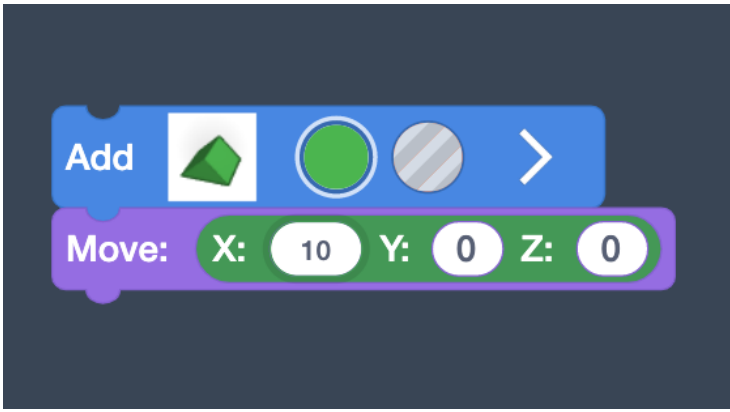
2. Moving

- Pick a Shape. Click and drag it into the black workspace.
- Find the “Move” function. Click and drag it into the code space until it attaches to the Shape:



- Play around with the X, Y and Z values individually.
 - At what position do all shapes start when imported?
 - What happens when you make a value negative?

Examples:



3. Scaling

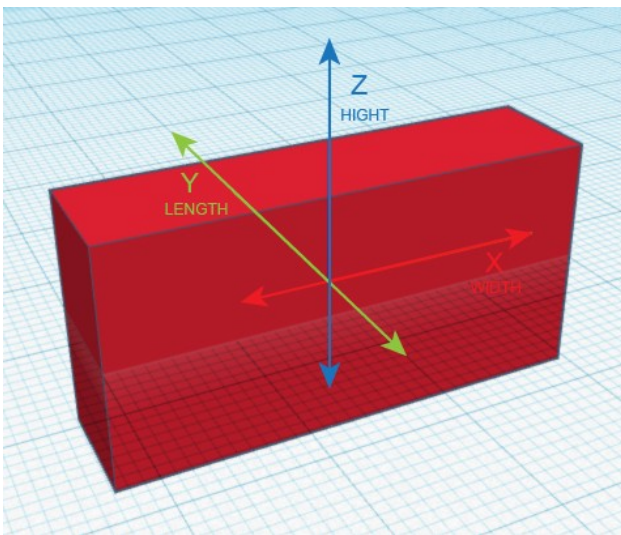
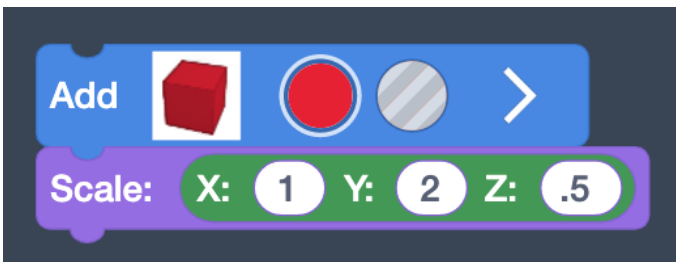
a. Some Shapes will have an extended size arrow that imports the Shape with that specified size. Import and play around with the options to see how the object changes.

Example:



b. Locate and import the “Scale” function into the code space.

c. Attach the “Scale” to the Shape and play around with changing the X, Y and Z variables.



4. Rotating

a. Import a new Shape or use an existing Shape.

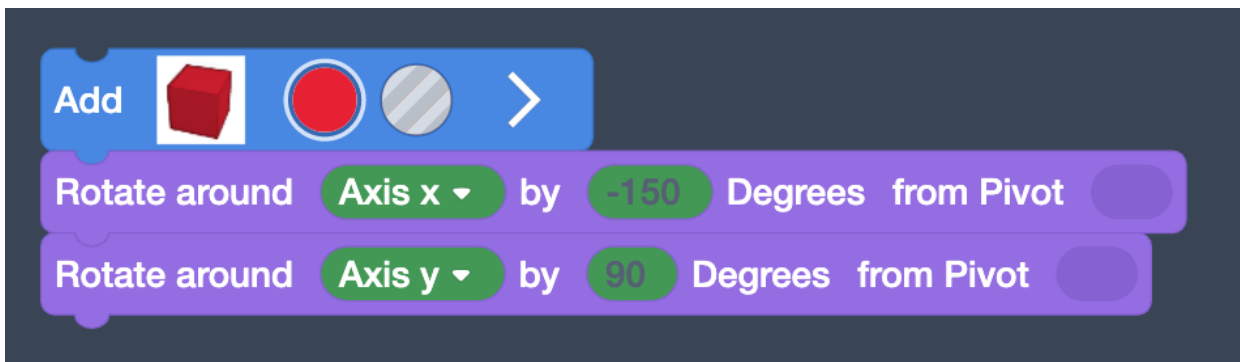
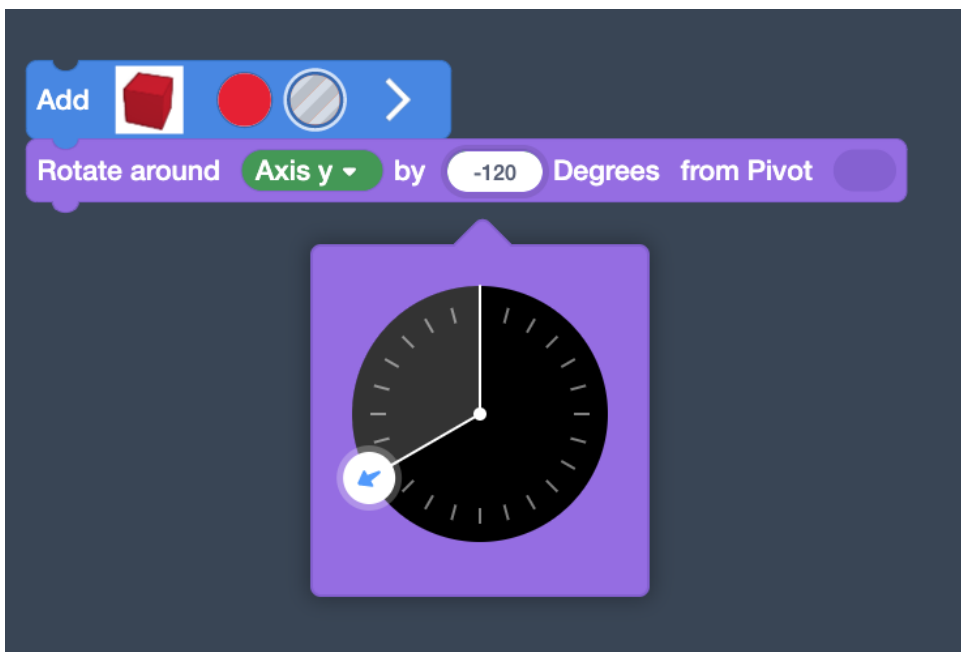
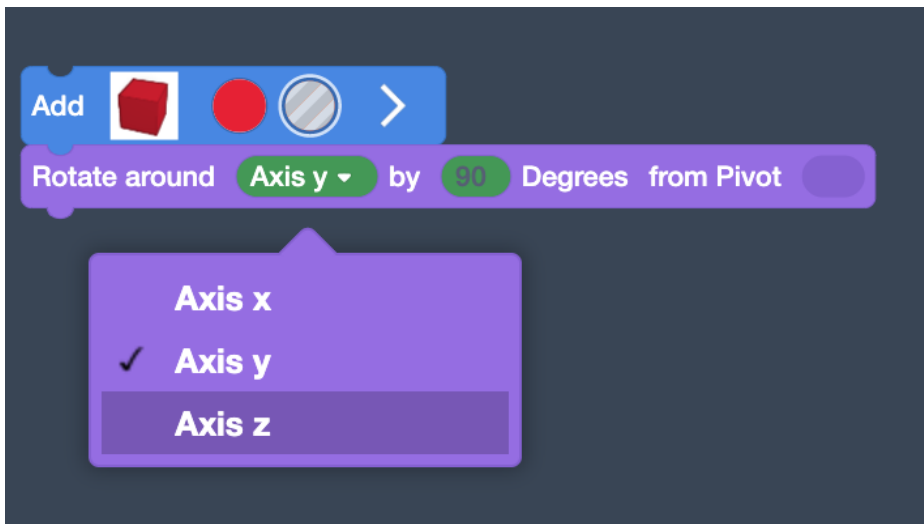
b. Find and add the “Rotate around” function to the Shape.

c. Rotate the Shapes around the X, Y and Z directions, switching the degree variable each time.

d. Try adding multiple rotations to the same shape.

Does the direction change the second/third time you rotate?

Example:

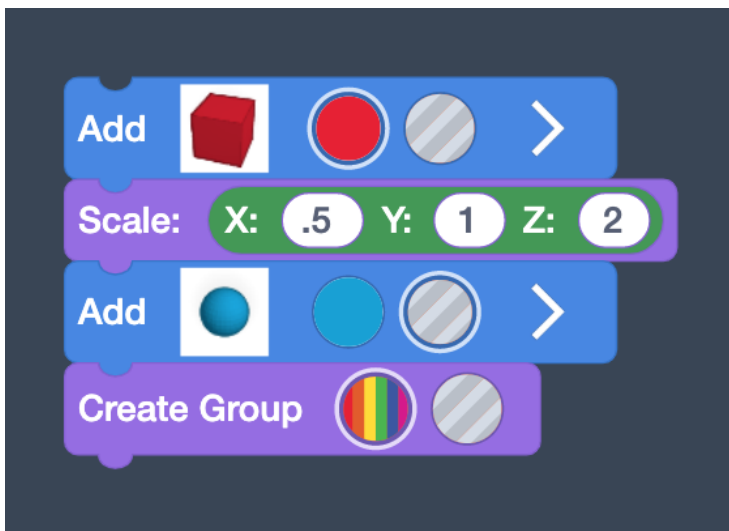
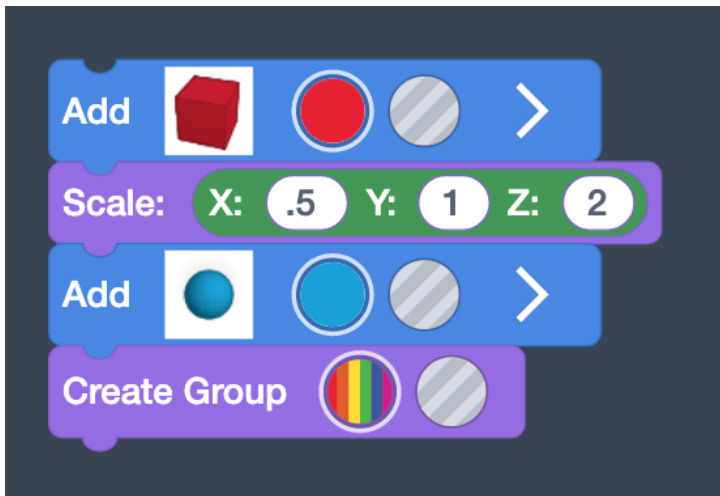


5. Group

a. Import two or more shapes.

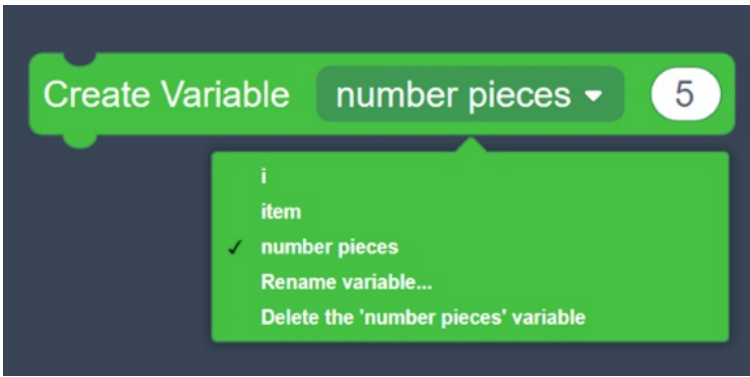
- b. Add a “Move”, “Rotate” or “Scale” one of them.
- c. Turn one of the shapes into a hole.
- d. Find the “Create group” function, add it to the shapes.
- e. Group the shapes as followed:

- Solid grouped to solid. Overall shape: Solid.
- Solid grouped to a hole. Overall shape: Solid.



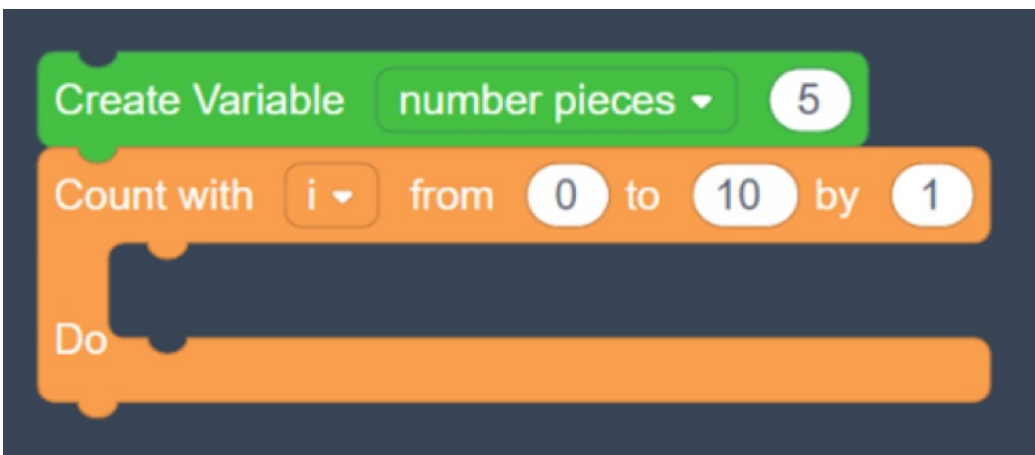
6. Define variable

- a. Import the “Create Variable” into the code space.
- b. In the drop-down options, select “Rename Variable...” and rename as “Number pieces”. In the number box on the right, change the variable to anything between 1 and 10. (This number will be a reference for the repeat variable to refer to.)



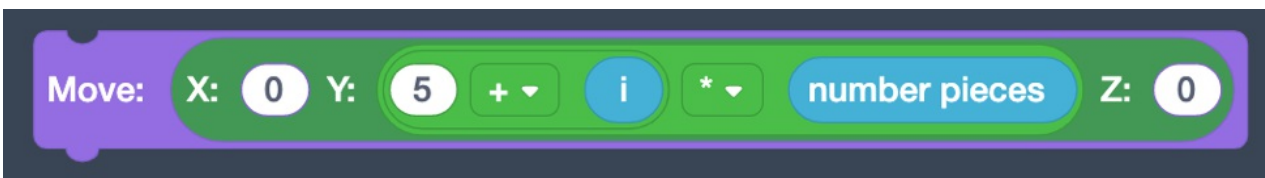
c. Find and attach a “Count with” to the bottom of the “Create Variable”. The variable “i” is the index number used in the Count loop

d. Under “Data” find the “Number pieces”, and drag the code into the “10” box.



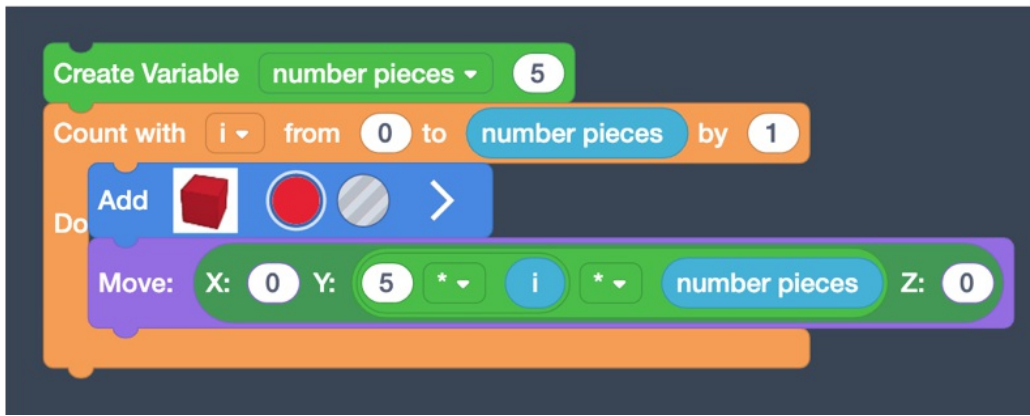
e. Import a “Shape”, a “Move”, and a computation block “0 + 0” found under “Math” into the code space. Change the + to any of the drop-down selection (+,-,*,/) and choose a number to move the shape by in the second number variable box. You can play the “0+0” blocks inside each other and play around with the math to see how the equation affects the X, Y, and Z movement.

Here is an example of a math equation placed in the Y variable:



$((5 + i) * \text{number pieces})$ The i stands for the “Count with” number variable and the number pieces references the number found in the “Count Variable”.

f. Add the “Shape” and “Move” code into “Count with” and play the code.

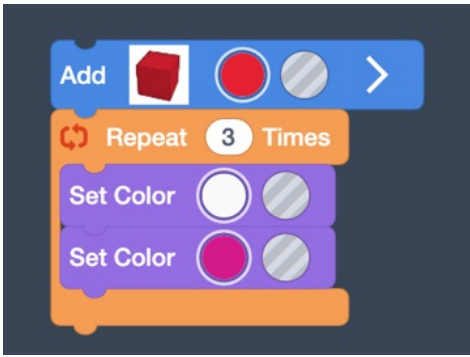


g. Play around with the numbers under “Move”, and “Create Variable”. Try changing the 1 in “count with”, to see how it affects the code.



7. Repeat variable.

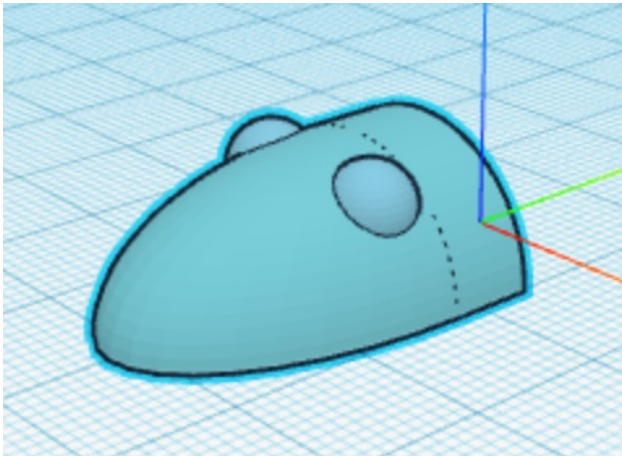
- Find and import the “Repeat” function.
- Add two or more “Set Color”.
- Play around with the colors and see if you can make a glowing pattern.



You should now be comfortable enough with Codeblocks to easily follow the activity instructions. Have fun!

Step 2: Design the Head

The over-all head size is 30cm in length, 20cm in width, and a 22 cm height. Use this code or create your own design.

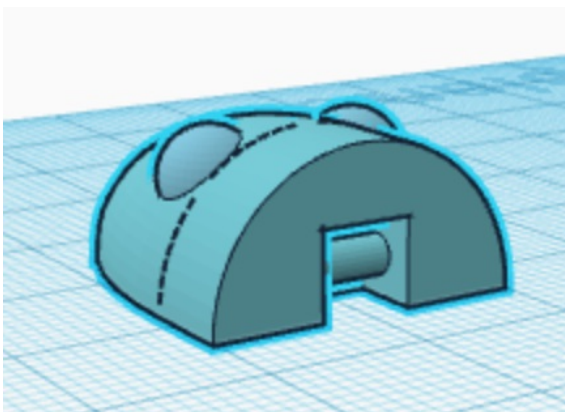


A. Create this code in your workspace to make a basic snakehead. Use the Create New Object block to give it a name. The head is made up of several shapes that are moved, grouped, and scaled. Step through the code to see how these commands are used to create a new shape.

```

Create New Object head
Add [Red Sphere]
Move: X: 0 Y: -5 Z: 0
Add [Green Sphere]
Scale: X: 1 Y: .5 Z: 1
Create Group
Move: X: 0 Y: 0 Z: 5
Scale: X: 1 Y: 2 Z: 1
Add [Red Cube] W: 20 L: 15 H: 20 edge: 0 Edge Steps: 10
Move: X: 0 Y: 8.5 Z: 0
Create Group
  
```

The back of the head should now look like this:



B. Create eyes using Spheres. Attach the following code to the bottom of the previous code.

```

// Eyes
Add [Blue Sphere] Radius: 3 Steps: 18
Move: X: 5 Y: -8 Z: 7
Add [Blue Sphere] Radius: 3 Steps: 18
Move: X: -5 Y: -8 Z: 7
Create Group
Move: X: 0 Y: -7 Z: 0
  
```

C. The head will have a pin at the back as part of a hinge with the body piece. Create the cutout with the following code, attached to the previous code. Step through the code to see what each step does.

```

// Create cut out for pin
Add [Orange Cylinder] Radius: 3 H: 6 Sides: 20 edge: 0 Edge Steps: 1
Rotate around Axis y by 90 Degrees from Pivot
Move: X: 0 Y: -9 Z: 2.5
Add [Red Cube] W: 6 L: 6 H: 6 edge: 0 Edge Steps: 10
Move: X: 0 Y: -6 Z: 2.5
Create Group
  
```

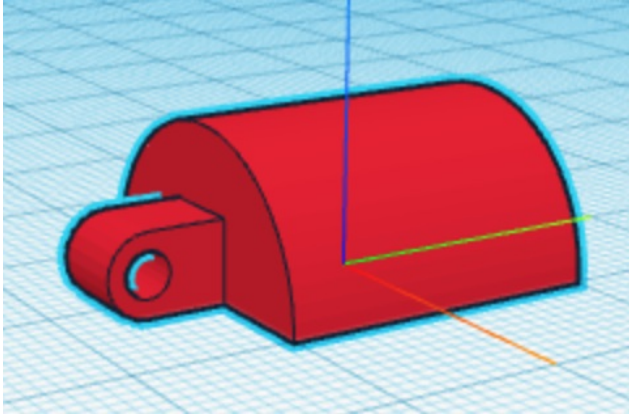
D. Finally the hinge pin is created in the cutout.

```

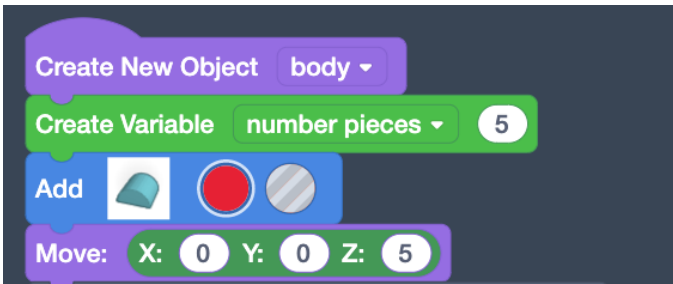
// Pin
Add [Orange Cylinder] Radius: 1.125 H: 7 Sides: 20 edge: 0 Edge Steps: 1
Rotate around Axis y by 90 Degrees from Pivot
Move: X: 0 Y: -9 Z: 2.5
Create Group
Move: X: 0 Y: -30 Z: 0
  
```

Step 3: Create the Body

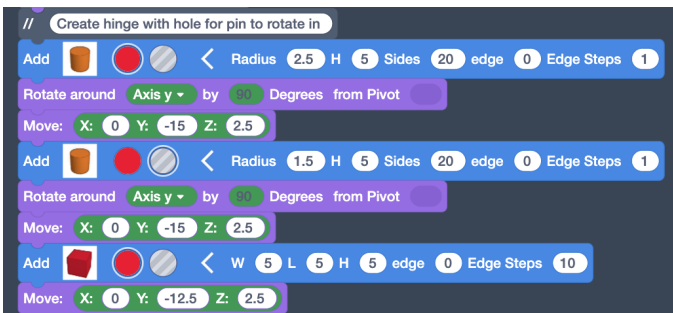
The overall dimension of the body is 20 cm long, 20 cm wide, and 10 cm tall, not including the hinge. Use this code or create your own design.



E. To create the body, you will make a Create New Object and give it a name. Add a Create Variable, and rename it as “number pieces”. This feature will come in handy later when you decide how many links in the chain you wish to print.

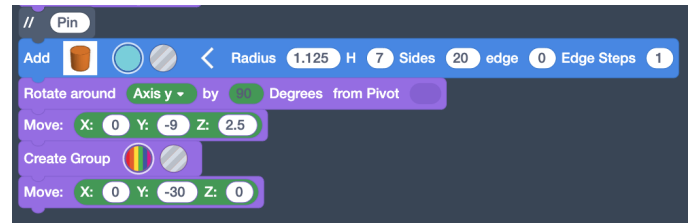


F. The hinge with a hole in the middle is added and later moved to attach to the head. Follow the code below:

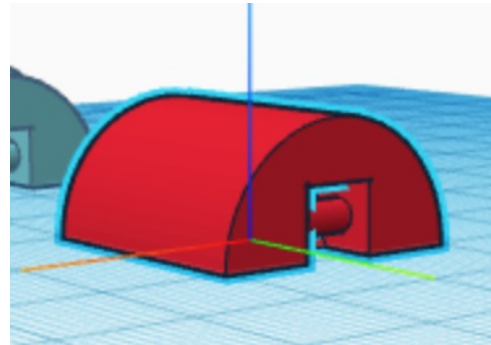


G. You'll notice the cutout and shapes in these following two steps are the same as in the head. You can copy and paste (Ctrl-C, Ctrl-V) them from the

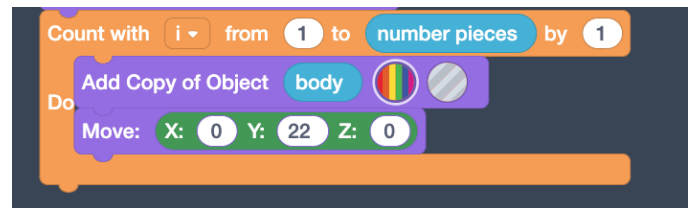
H. Add the pin to the cutout with the following code



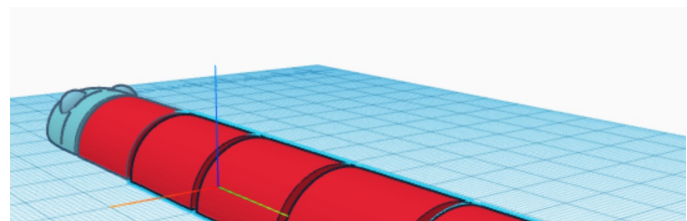
The body should look as follows from the back.



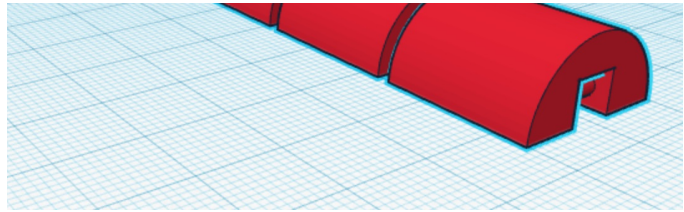
I. The fun part of the snake code is the Count loop to add body pieces. Use the Count loop to multiply the body pieces by any number. Create a variable (as explained in the Step 1 exploratory exercises) called “number charms”. Set the variable to your chosen number. You will add additional body pieces to the original one you created, so the count loop goes from 1 to “number charms-1”. The Move applies to the entire body since the new section is added to it.



After the code is run the body would look as follows. If you want more or fewer links, go back to step E. and change the value in the Create Variable block.



head code and attach the code to the snake body. Be sure to change the Move parameters.

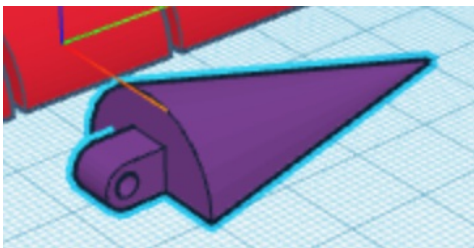


```
// Create cut out for pin
Add [Cylinder] < Radius 3 H 6 Sides 20 edge 0 Edge Steps 1
Rotate around Axis y by 180 Degrees from Pivot
Move: X: 0 Y: 7 Z: 2.5
Add [Cylinder] < W 6 L 6 H 6 edge 0 Edge Steps 10
Move: X: 0 Y: 10 Z: 2.5
Create Group
```

Step 4: Add a Tail

The overall size of the tail is 47.5 cm long, 20 cm wide and 10 cm tall. Use this code or create your own design.

The last Move block positions the tail at the end of the body. It requires two Math blocks nested in the Y variable of the Move block and codes this calculation: $((\text{number piece} - 1) * 22)$. The code should be as follows:

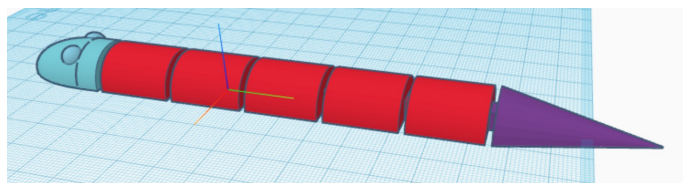


J. The last shape is the tail which is a half cone. Create a New Object and name it. Add a Cone and follow the code below.

```
// Create hinge
Add [Cylinder] < W 5 L 5 H 5 edge 0 Edge Steps 10
Move: X: 30 Y: -12.5 Z: 2.5
Add [Cylinder] < Radius 2.5 H 5 Sides 20 edge 0 Edge Steps 1
Rotate around Axis y by 180 Degrees from Pivot
Move: X: 30 Y: -15 Z: 2.5
Add [Cylinder] < Radius 1.5 H 5 Sides 20 edge 0 Edge Steps 1
Rotate around Axis y by 180 Degrees from Pivot
Move: X: 30 Y: -15 Z: 2.5
Create Group
Move: X: 0 Y: -2 Z: 0
Move: X: -30 Y: (number pieces - 1) * 22 Z: 0
```

```
Create New Object Tail
// Tail
Add [Cone] < Top Radius 0 Bottom Radius 10 H 20 Sides 24
Scale: X: 1 Y: 1 Z: 2
Move: X: 30 Y: 0 Z: 0
Rotate around Axis x by 180 Degrees from Pivot
Add [Cylinder] < W 20 L 40 H 20 edge 0 Edge Steps 10
Move: X: 30 Y: 0 Z: -10
Create Group
Move: X: 0 Y: 10 Z: 0
```

Final shape snake code will vary with the chain count:



K. The hinge will be the same as the one on the body. Feel free to copy and paste the shapes to the end of the tail, but be mindful to change the Move variables.

The code described above can be found at <https://www.tinkercad.com/codeblocks/a1WpkADG6eQ>

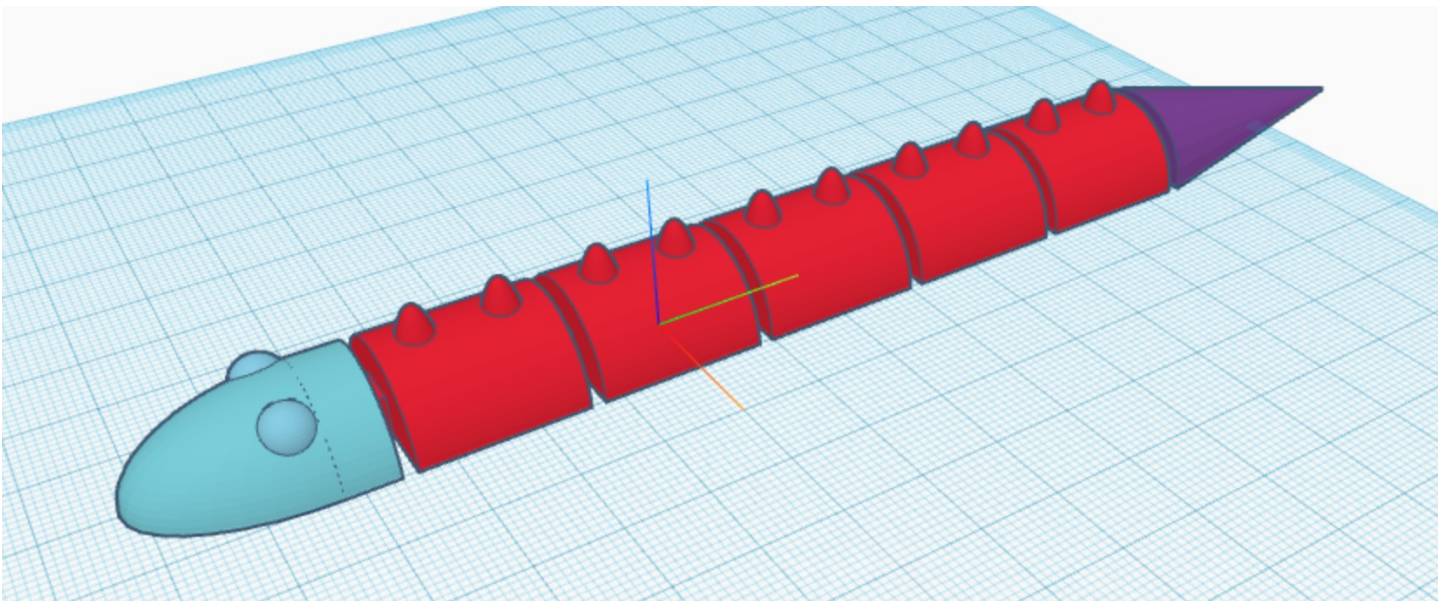
Step 5: Export the Part and 3D Print It

Export the part and use the slicer program to prepare it for the 3D printer. Make sure not to print with a "brim" as it will fill in the gaps in the hinges.



Step 6: Project Extensions

Add shapes to the body! As long as the cutouts, pins and overall width of the bodies are unaltered, you can add new shapes or designs to the snake. Note the use of scale blocks you add to the body. This is done by trial and error.



Create New Object **body** ▾

Create Variable **number pieces** ▾ 5

Add



Move: X: 0 Y: 0 Z: 5

// Add spikes

Add



< Radius 1.5 H 5 Sides 24

Scale: X: .5 Y: .5 Z: 1

Move: X: 0 Y: 0 Z: 11

Move: X: 0 Y: 5 Z: 0

Add



< Radius 1.5 H 5 Sides 24

Scale: X: .5 Y: .5 Z: 1

Move: X: 0 Y: 0 Z: 11

Move: X: 0 Y: -5 Z: 0

Create Group





Thanks for trying our Instructable!

We hope you had fun designing and sharing your charm chains. What did you make? We want to see! If you did this in a K-12 classroom, what subject was it in?

Send us an email or leave us a comment so we can see what you're making.

Visit our website k12maker.mit.edu to get resources for K-12 teachers:

- [Maker skills workshops for K-12 educators](#) - Spring, Summer, and Fall
- [Makerspace Design Guide](#) for K-12 Makerspaces
- [Lists of Tools and Materials and illustrated charts](#) to print and post
- [Supervision and safety guidelines](#) for shop administrators
- [Training guides for common tools](#) (including student checklists and refresher guides)
- Our [Maker Methodology for designing Maker Projects for the core curriculum](#), including sample projects