Spheros & Variables

Variables to Control Loops

MTH1W – Day 1 (and 2), Part C

MTH1W Curriculum Covered in This Lesson:

☑ C2.1a [Coding] Use coding to demonstrate understanding of **variables**
☐ C2.1b [Coding] Use coding to demonstrate understanding of **parameters**
☑ C2.1c [Coding] Use coding to demonstrate understanding of **equations**
☐ C2.1d [Coding] Use coding to demonstrate understanding of **inequalities**

☑ C2.2a [Coding] Create code by decomposing situations into computational steps in order to **represent mathematical concepts and relationships**
☑ C2.2b [Coding] Create code by decomposing situations into computational steps in order to **solve problems**

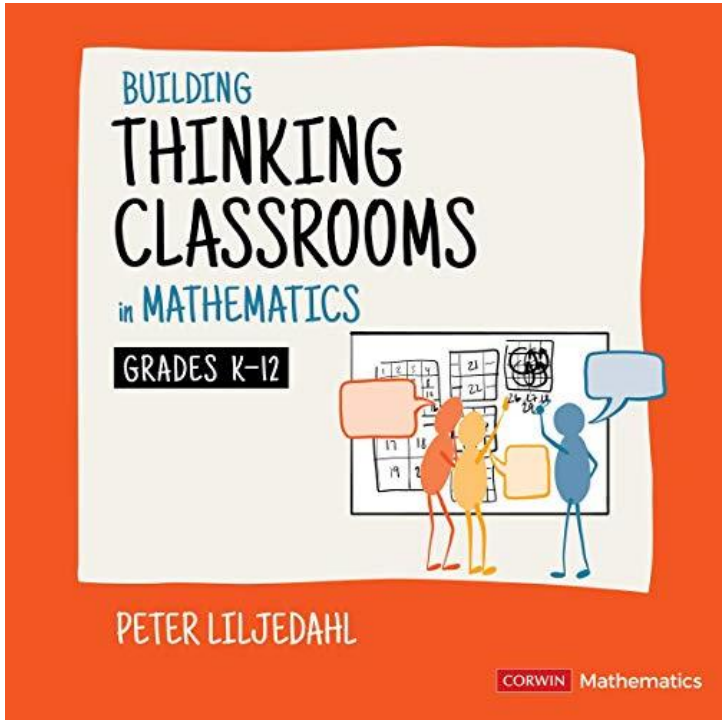
☑ C2.3a [Coding] **Read code to predict** its outcome
☑ C2.3b [Coding] **Alter code** to adjust **constraints, parameters** to represent a similar or new mathematical situation
☑ C2.3c [Coding] **Alter code** to adjust **outcomes** to represent a similar or new mathematical situation

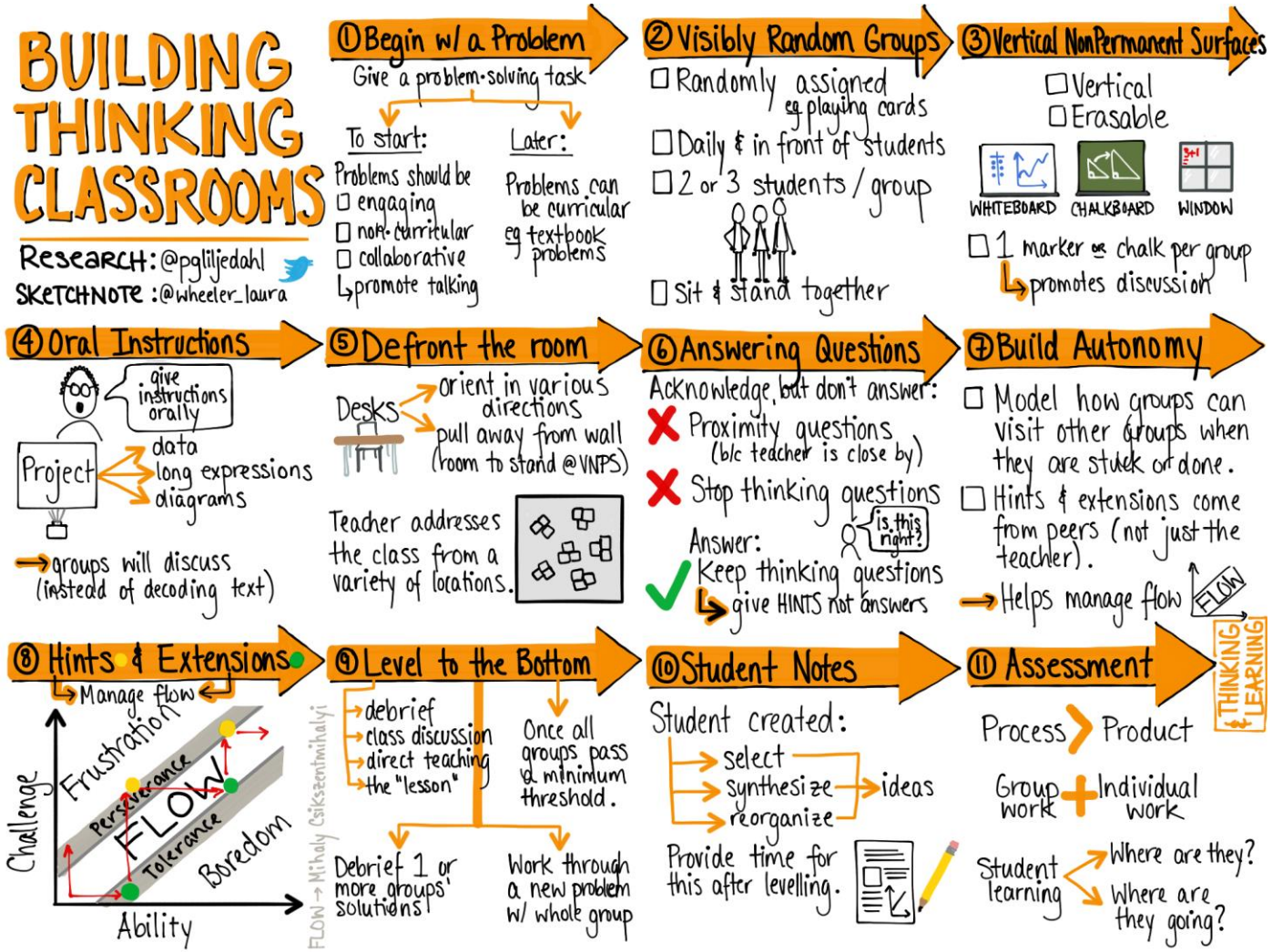
Sequencing and Timing:

- On Day 1, students will cover part A, Part B and some of Part C (this lesson).
- On Day 2, students finish Part C (this lesson) and cover Part D.

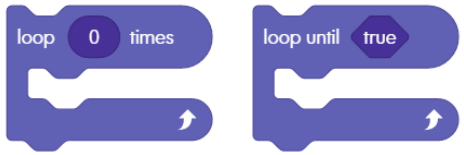
Original Curriculum Language:
C2.1 [Coding] use coding to demonstrate an understanding of algebraic concepts including variables, parameters, equations, and inequalities
C2.2 [Coding] create code by decomposing situations into computational steps in order to represent mathematical concepts and relationships, and to solve problems
C2.3 [Coding] read code to predict its outcome, and alter code to adjust constraints, parameters, and outcomes to represent a similar or new mathematical situation

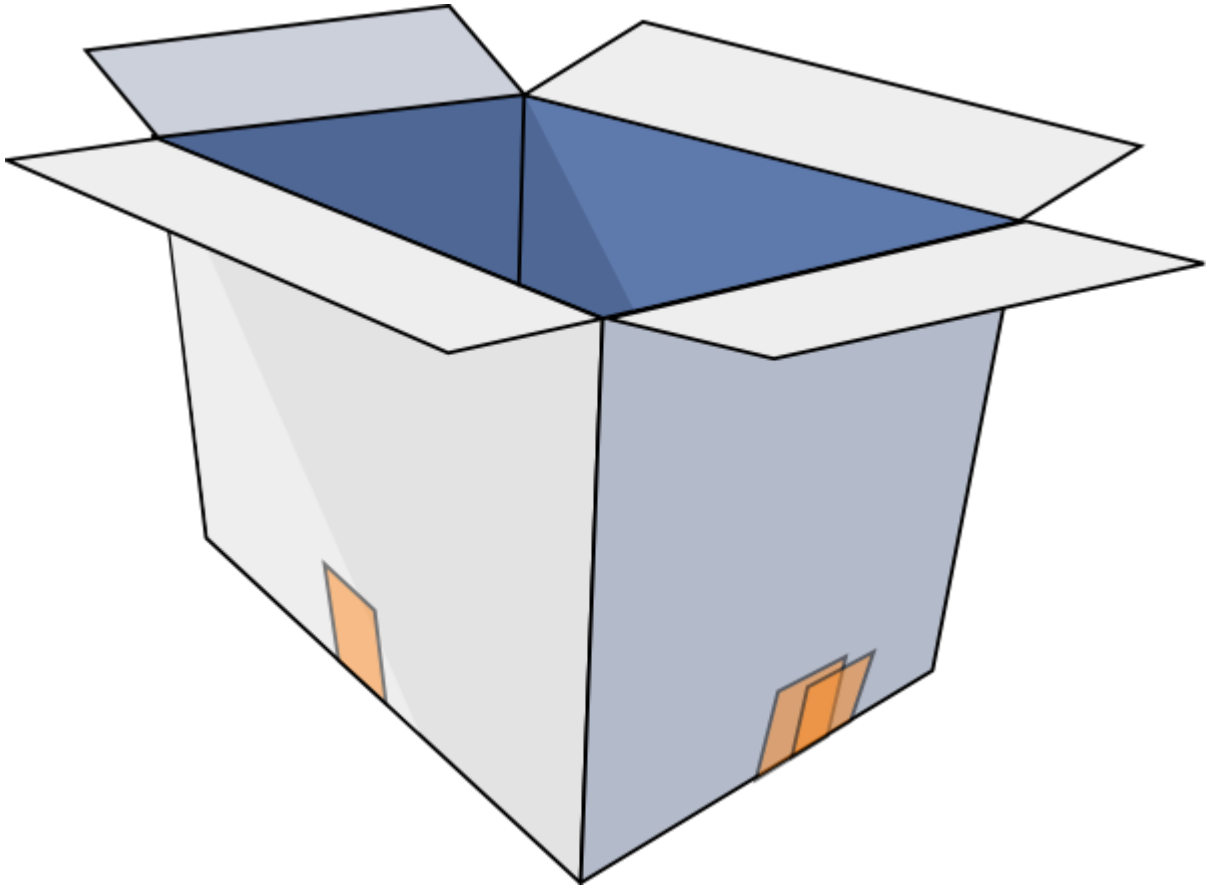
This lesson is set up to follow the format of Building Thinking Classrooms by Peter Liljedahl.





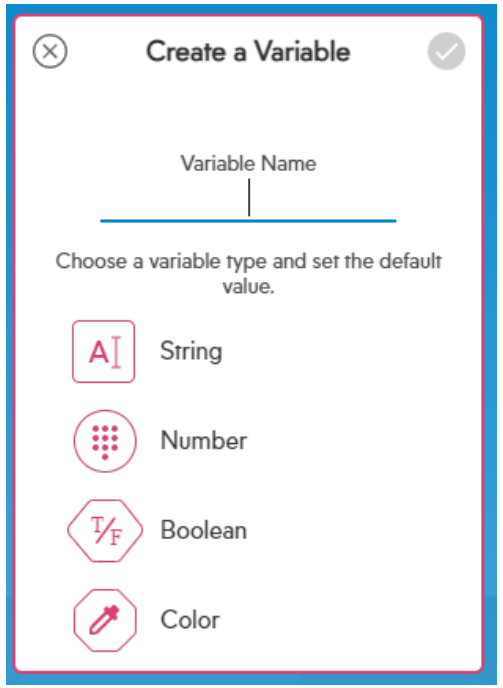
Basic Coding Components

Output	*Changes the robot:* Moves. Spins. Re-directs to a new angle. Makes sound. Speeds up. Lights up.	
Math	*Calculation that results in a number:* +, - , *, /, square root	
Boolean Expressions	*Calculation that results in **true or false**:* >, <, =, >=, <=, and, or, not	
Control	Ifs — ***Decides** which piece of code to run:* Uses a Boolean expression and output or math.	
	Loops — ***Repeats** code:* Uses a Boolean expression and output or math.	
Variables	Named pieces of memory where you can store things to use later OR to store calculation results.	Let's look at variables!
Functions	Named pieces of code where you can group code together to use it later.	

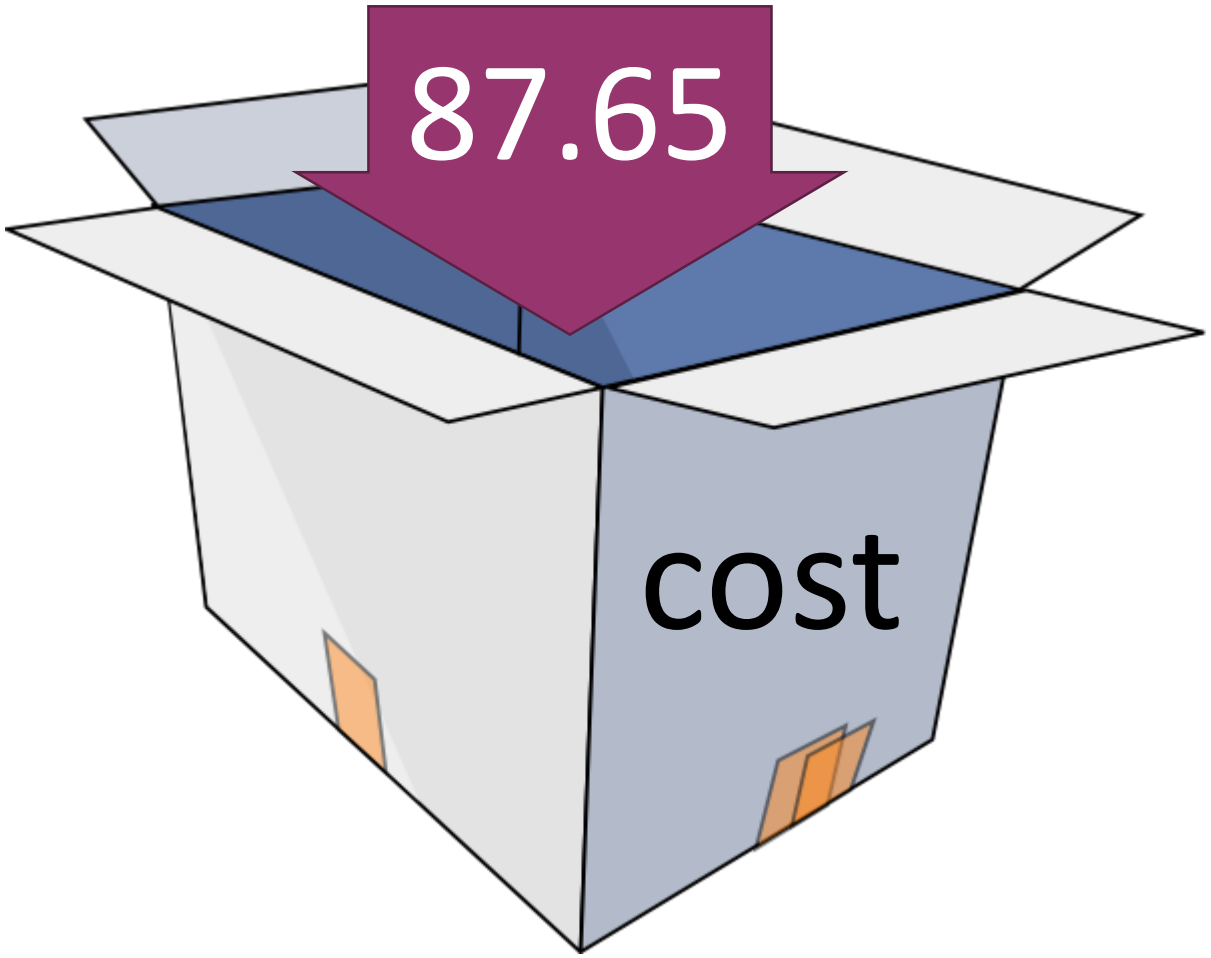


A variable is used in programming to store a value until we need it again.

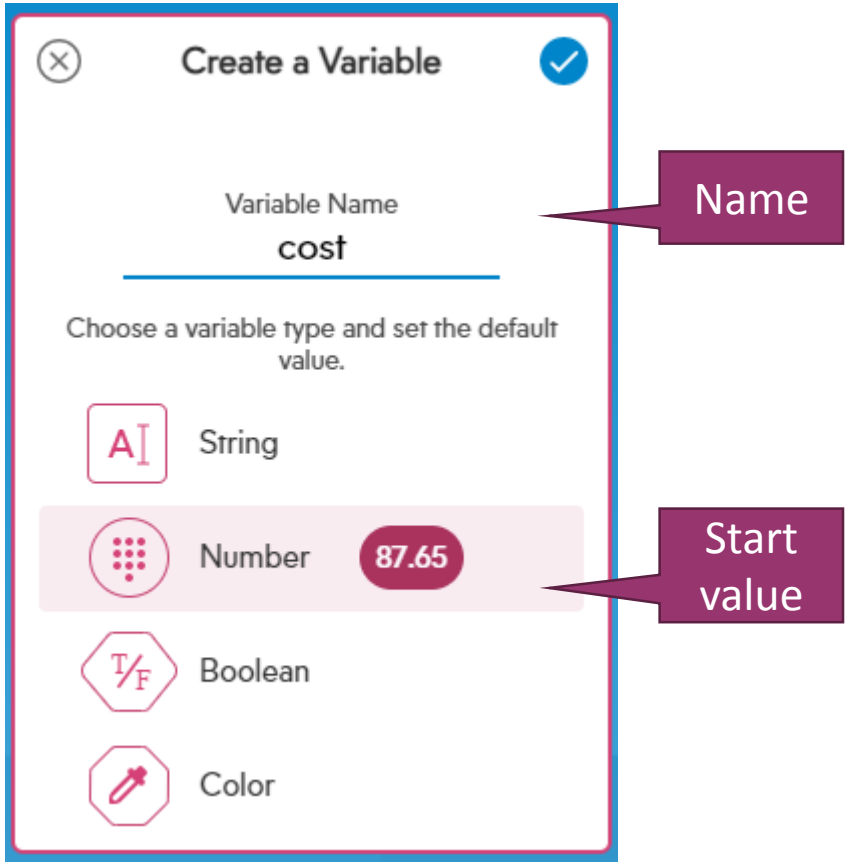
It is sort of like a box.



How sphero creates variables



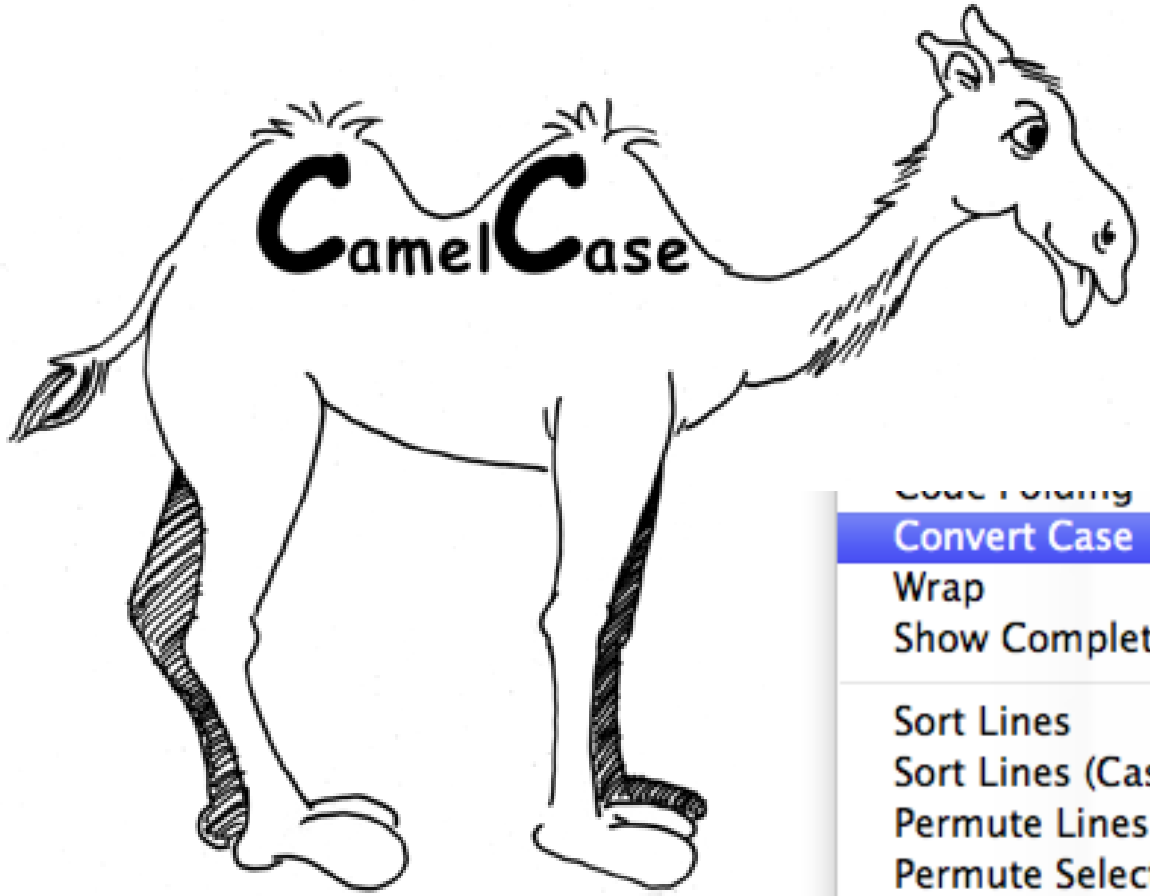
Variables have names so that you can call them again in your code.



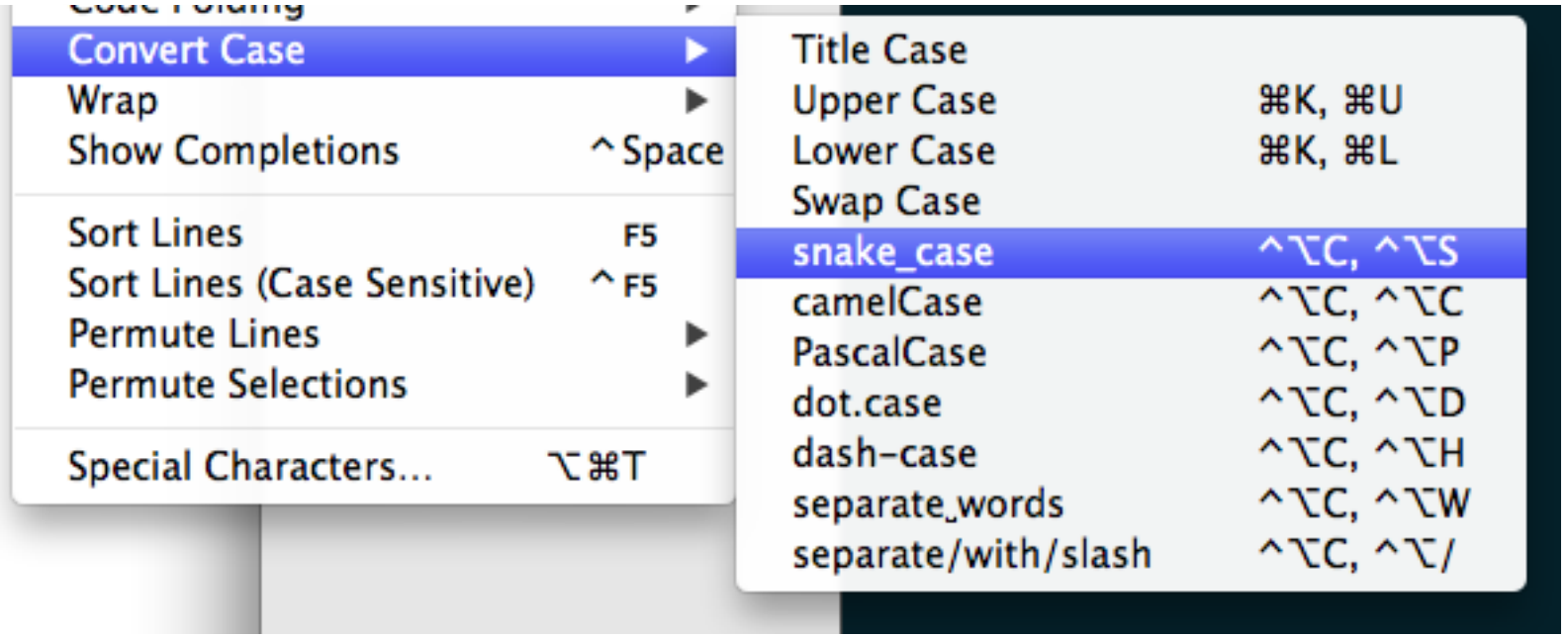
Variable Naming Rules:

Because variables are used in the code, they have be named carefully so the computer can understand.

1. No spaces. Camel case or underscores are fine.
2. Meaningful is easier in the long run.
3. Cannot start with a number. Otherwise, numbers are fine.
4. Cannot contain odd characters.
5. Cannot contain reserved words. (eg. setText or onEvent). These already have a purpose in javascript.



What is camelCase?



Which are valid variable names?

2many	impt*
T or F	tallTree
yOrN?	chicken

Naming Rules

1. No spaces.
2. Meaningful.
3. Cannot start with a number.
4. Cannot contain odd characters.
5. Cannot contain reserved words.

Sphero's variables also types: String and Number.

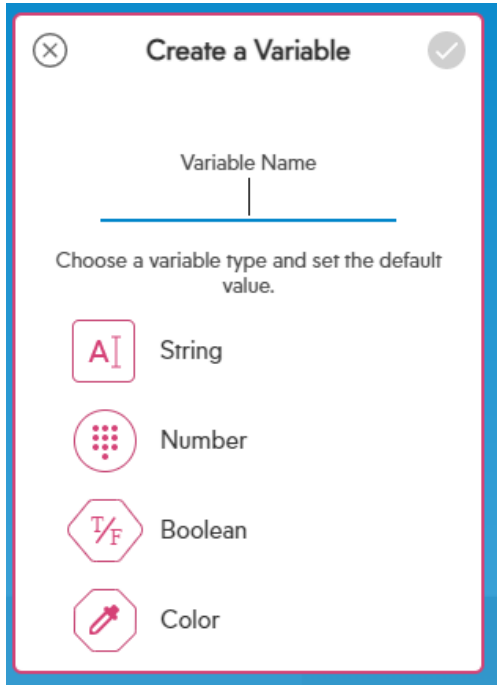
- A type is a kind of data – for example, words are text.
- A type is also an amount of memory allocated to a variable.
- A type is also the kinds of operations that can be done with a variable.

String Data Examples

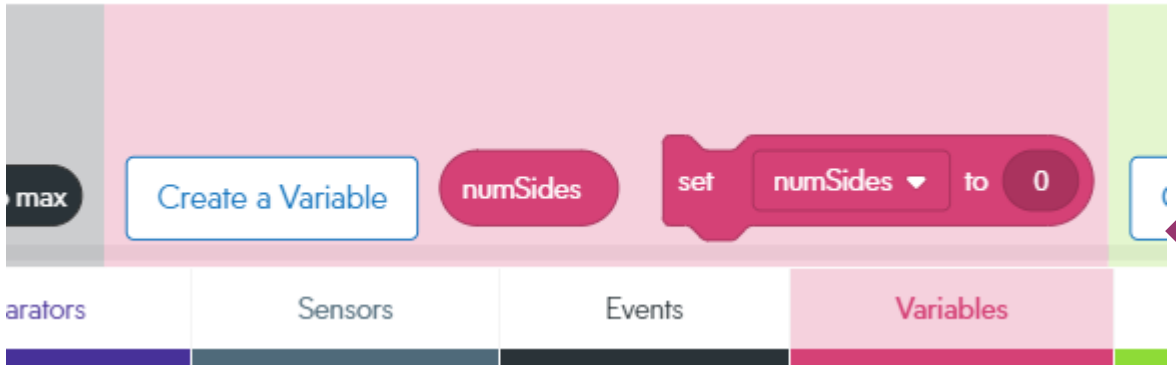
- Gorski
- cat
- Frogs are green
- (905) 567-9345

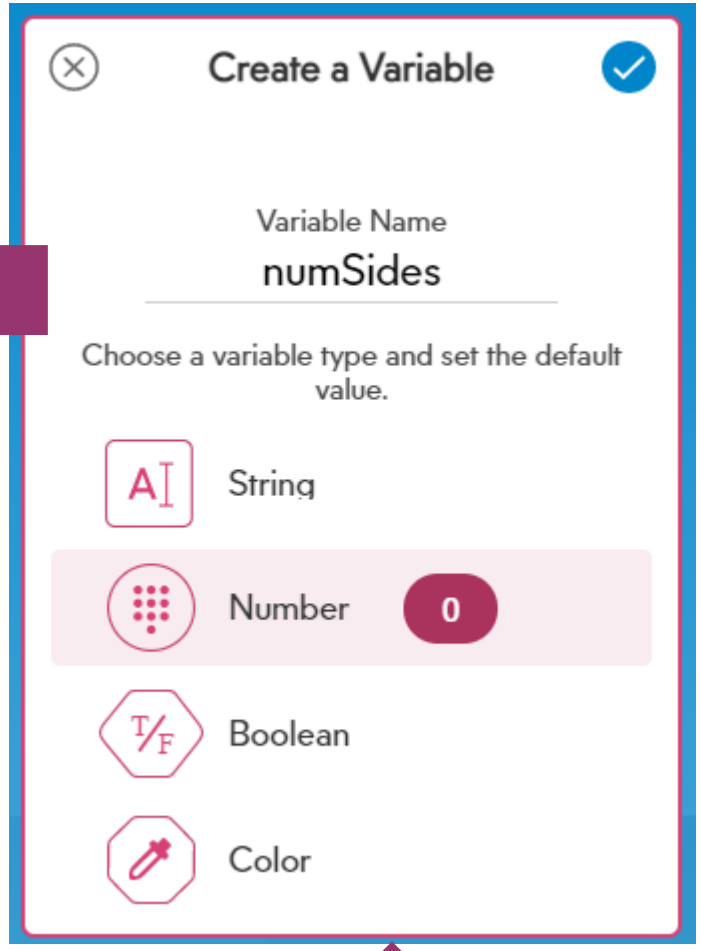
Number Data Examples

- 78.545
- -4
- 98
- 0

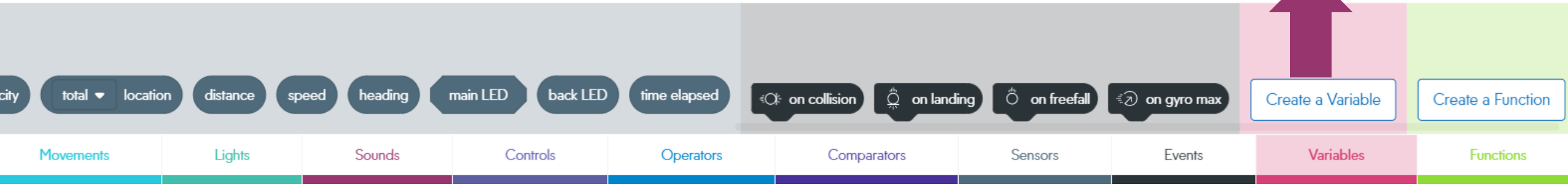


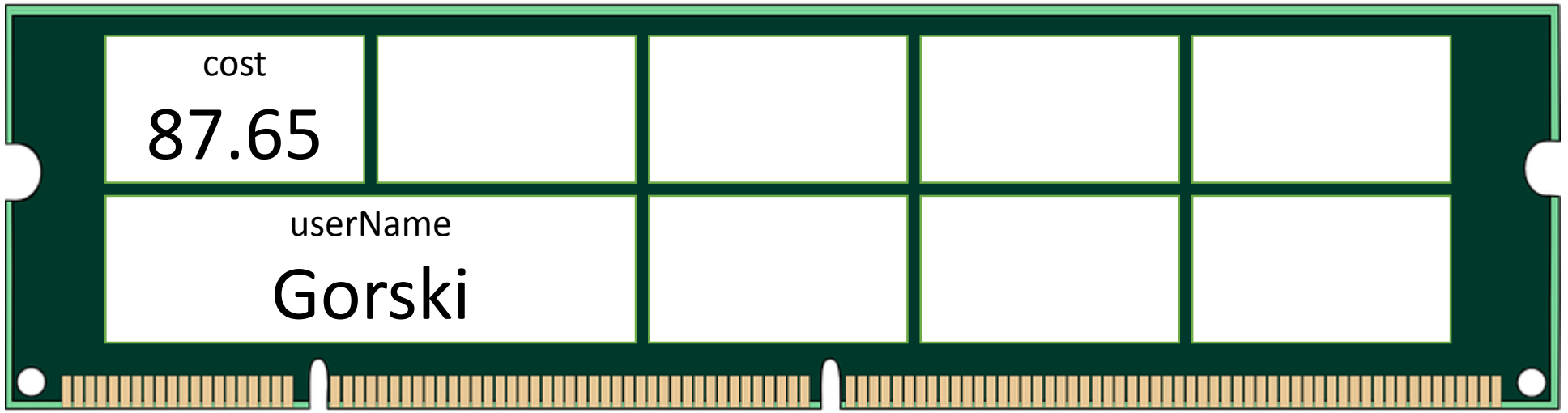
There are two other types, what are they?

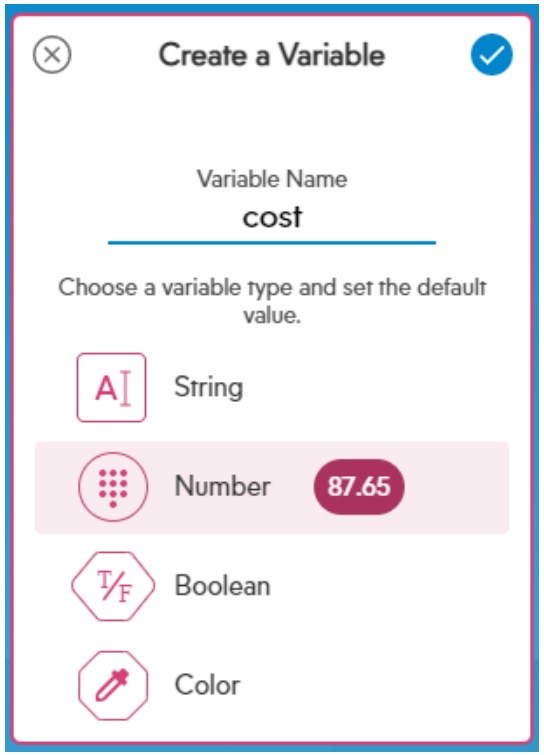


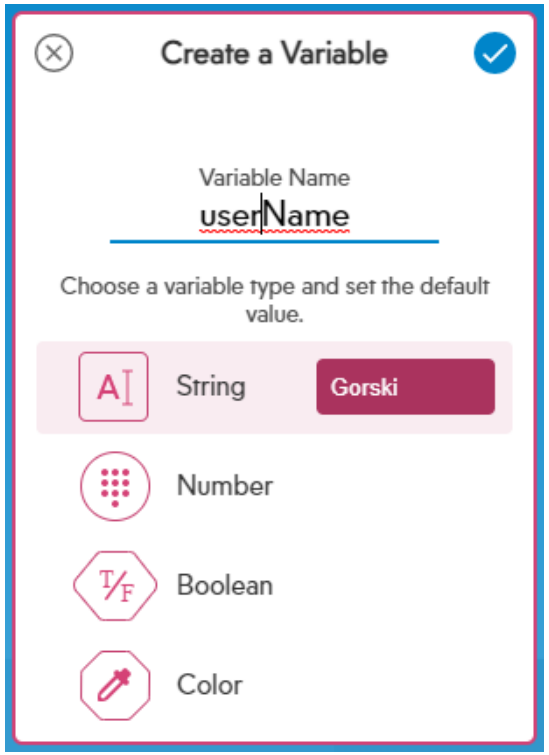


Steps to make a new variable in sphero.edu







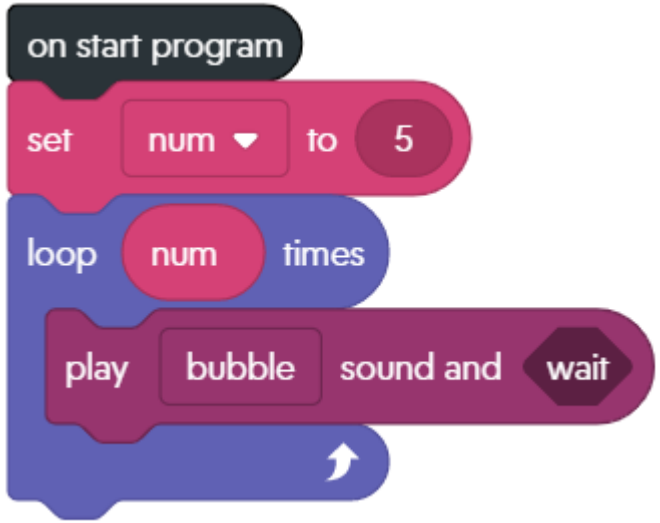


A variable is a space in memory.
It has a name, a value, and a type.

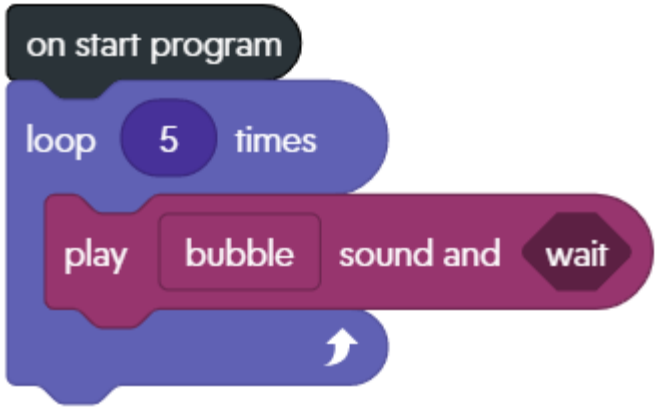
Its value can change whenever you want. Its name and type cannot.

Which type is smaller: String or Number?

1st



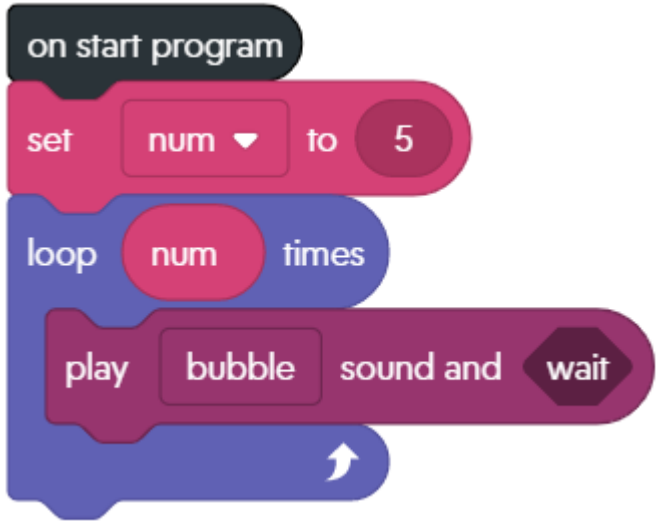
2nd



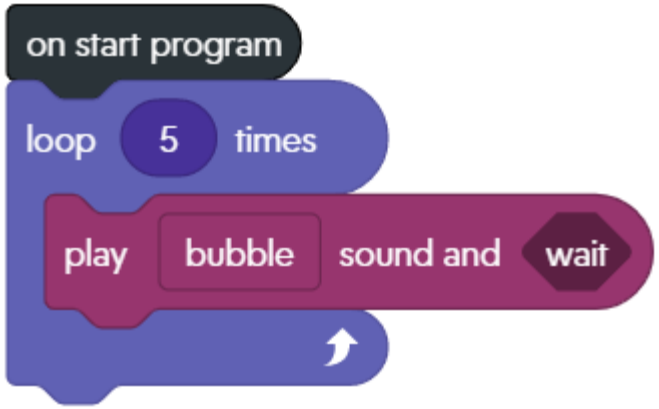
variables

These programs would produce the same result.

1st



2nd

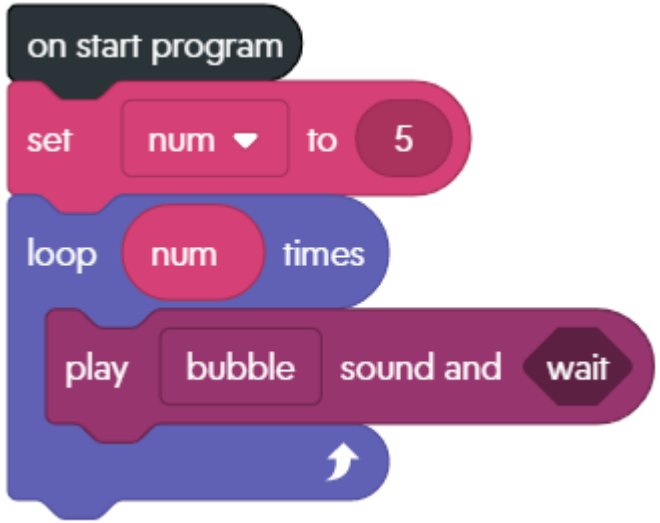


variables

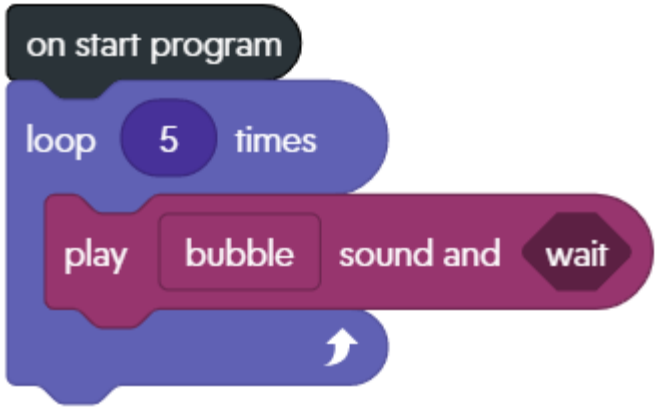
These programs would produce the same result.

What are some reasons that the first is better?

1st



2nd



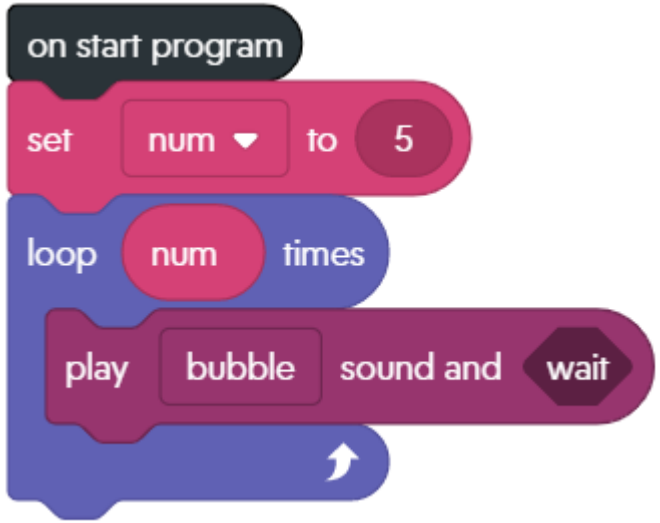
variables

These programs would produce the same result.

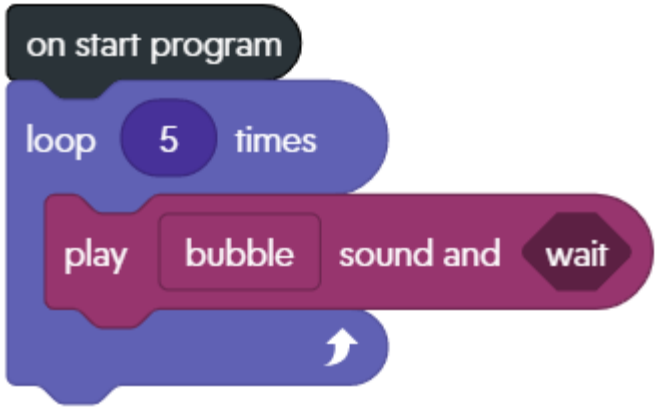
What are some reasons that the first is better?

Everything that needs editing is in a variable at the top.

1st



2nd



variables

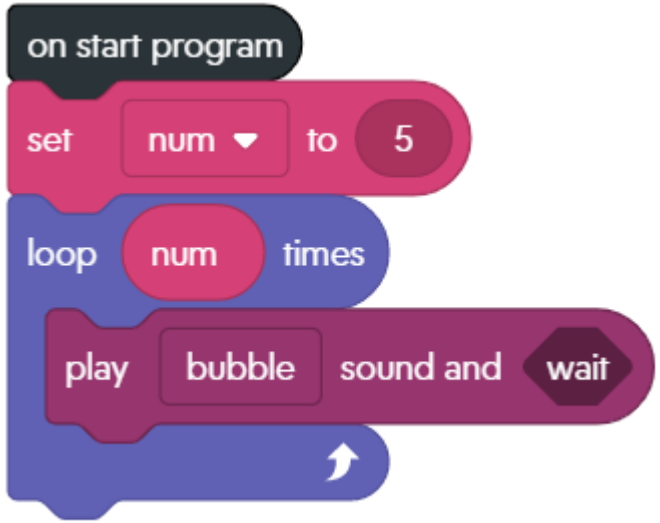
These programs would produce the same result.

What are some reasons that the first is better?

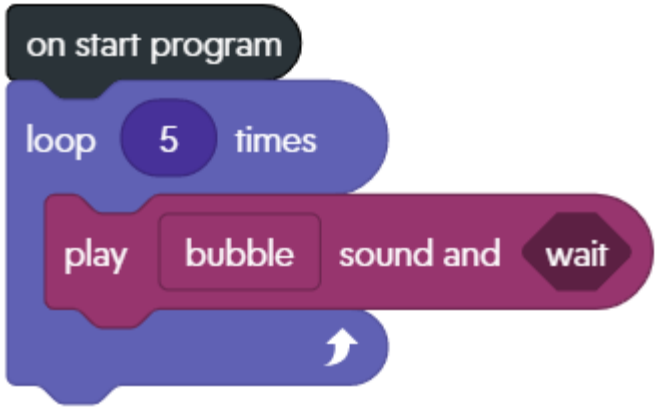
Everything that needs editing is in a variable at the top.

Others using your code don't need to root through it to figure it out.

1st



2nd



variables

These programs would produce the same result.

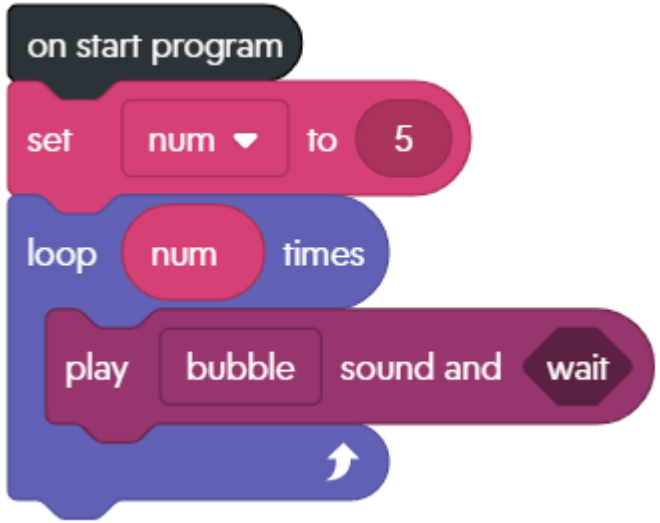
What are some reasons that the first is better?

Everything that needs editing is in a variable at the top.

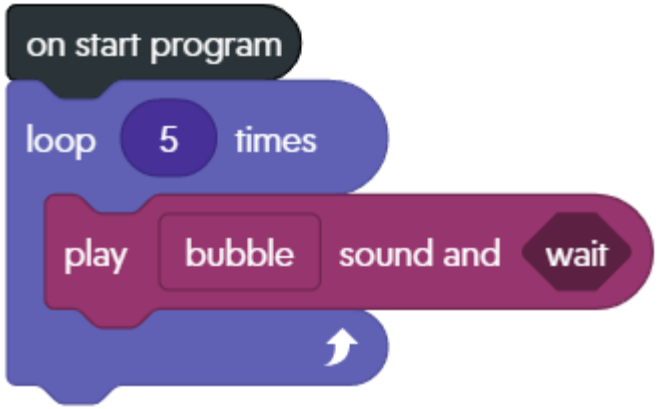
Others using your code don't need to root through it to figure it out.

They edit the variable and they are done!

1st



2nd



variables

These programs would produce the same result.

Depending on what I wanted to do with the code, I might use the first OR the second.

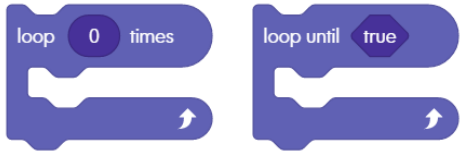
What are some reasons that the first is better?

Everything that needs editing is in a variable at the top.

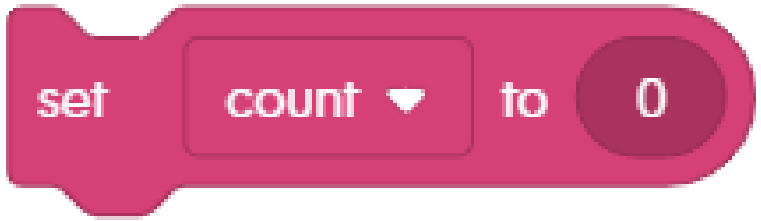
Others using your code don't need to root through it to figure it out.

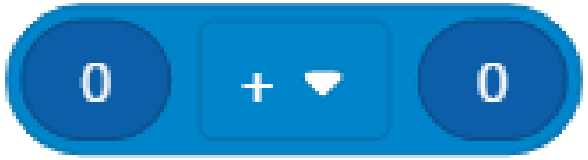
They edit the variable and they are done!

Basic Coding Components

Output	*Changes the robot:* Moves. Spins. Re-directs to a new angle. Makes sound. Speeds up. Lights up.	
Math	*Calculation that results in a number:* +, - , *, /, square root	Let's look at math!
Boolean Expressions	*Calculation that results in **true or false**:* >, <, =, >=, <=, and, or, not	
Control	Ifs — ***Decides** which piece of code to run:* Uses a Boolean expression and output or math.	
	Loops — ***Repeats** code:* Uses a Boolean expression and output or math.	
Variables	Named pieces of memory where you can store things to use later OR to store calculation results.	
Functions	Named pieces of code where you can group code together to use it later.	

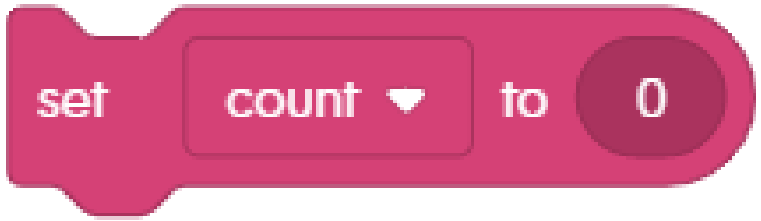
To build up math statements, you set with different pieces….

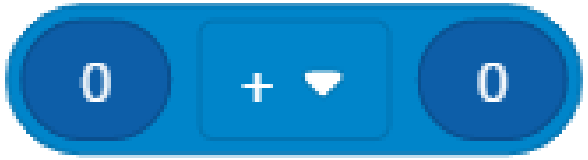




That you combine to create a mathematical expression:

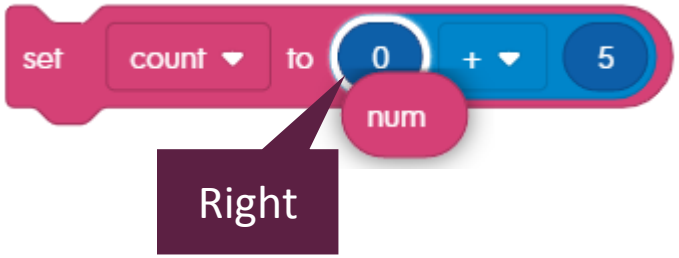
To build up math statements, you set with different pieces….



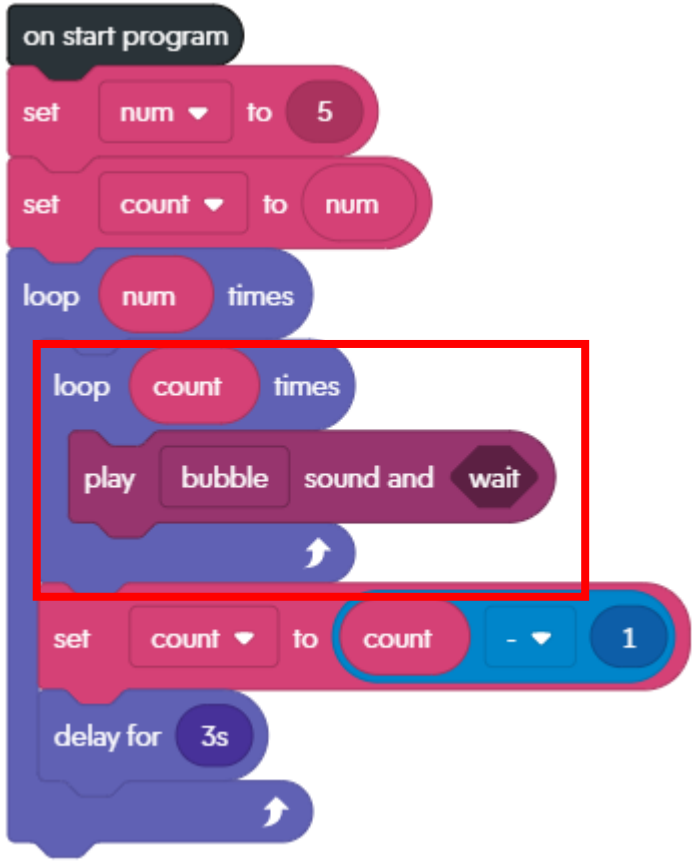


That you combine to create a mathematical expression:

Be careful that the right thing is selected before you place the block:



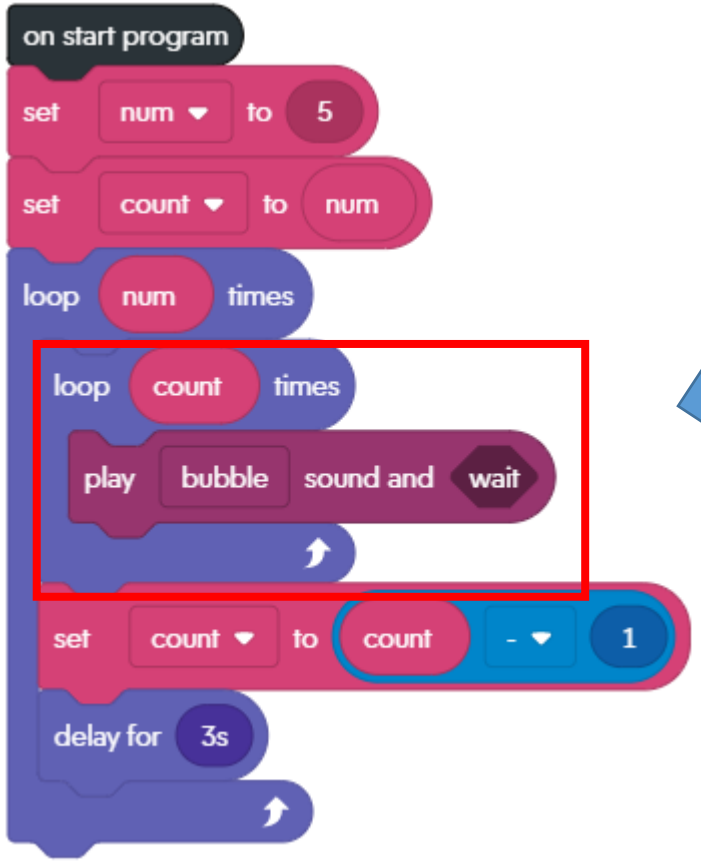
The Bubble Count Down



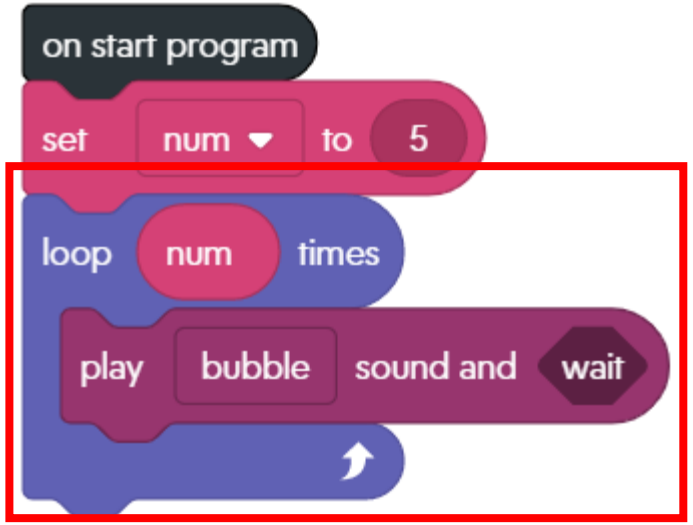
variable

The code on the right puts the loop that does the counting INSIDE another loop.

The Bubble Count Down



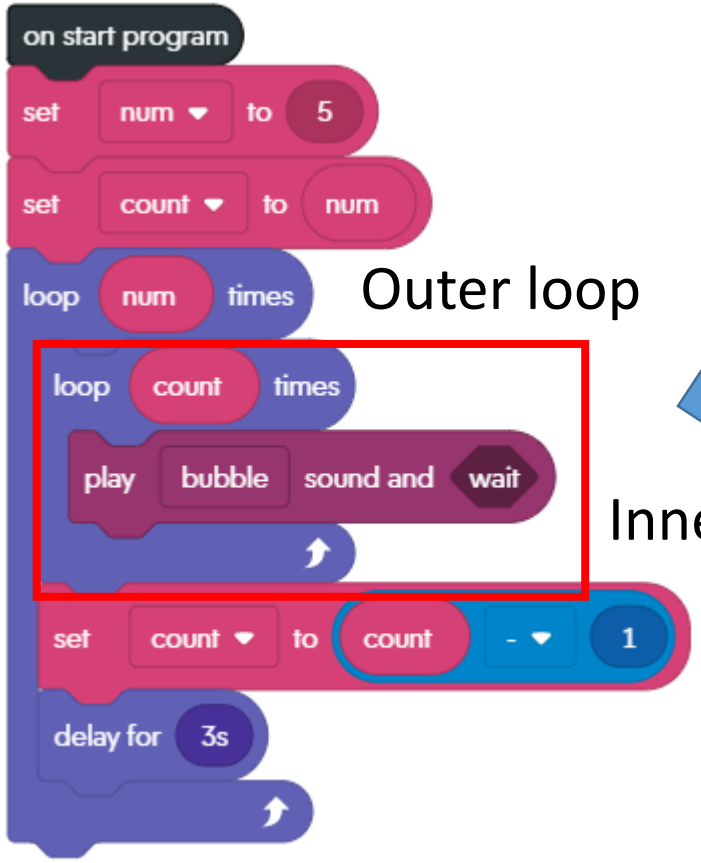
variable



Previous loop inside other loop.

The code on the right puts the loop that does the counting INSIDE another loop.

The Bubble Count Down

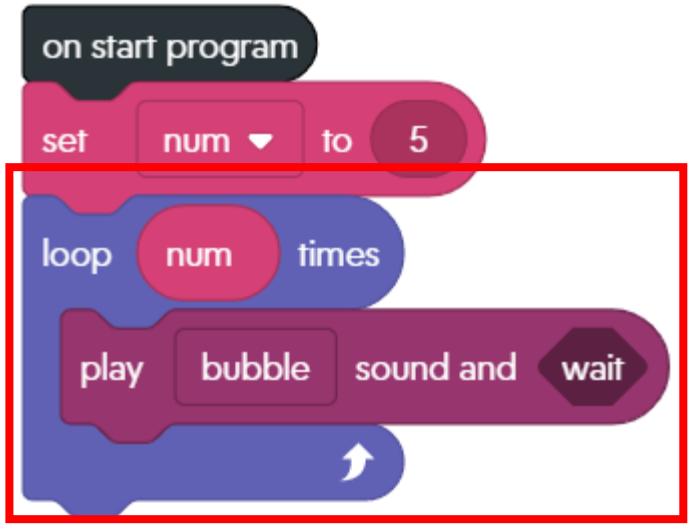


Outer loop

Inner loop

More Outer loop

variable



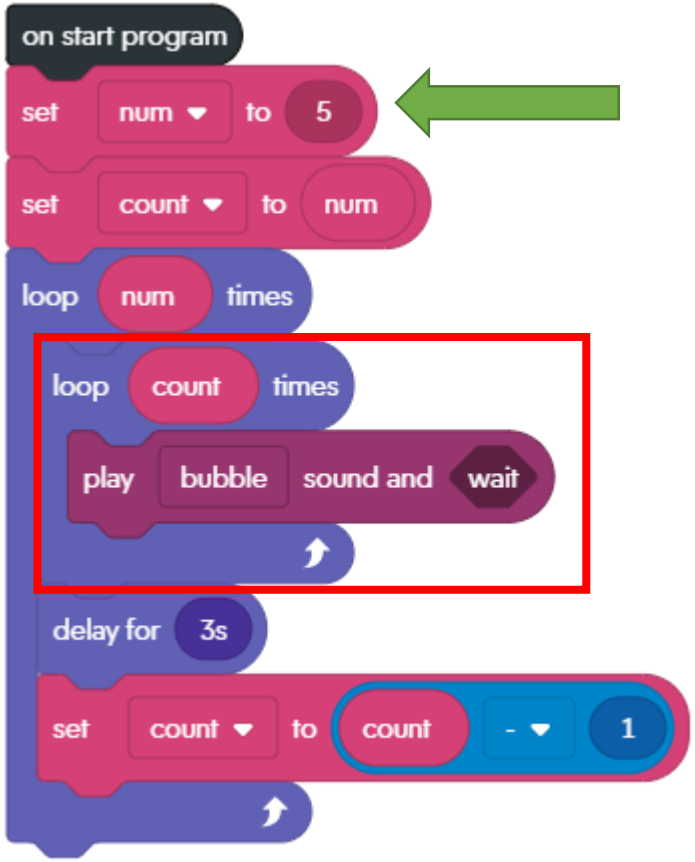
Previous loop inside other loop.

The code on the right puts the loop that does the counting INSIDE another loop.

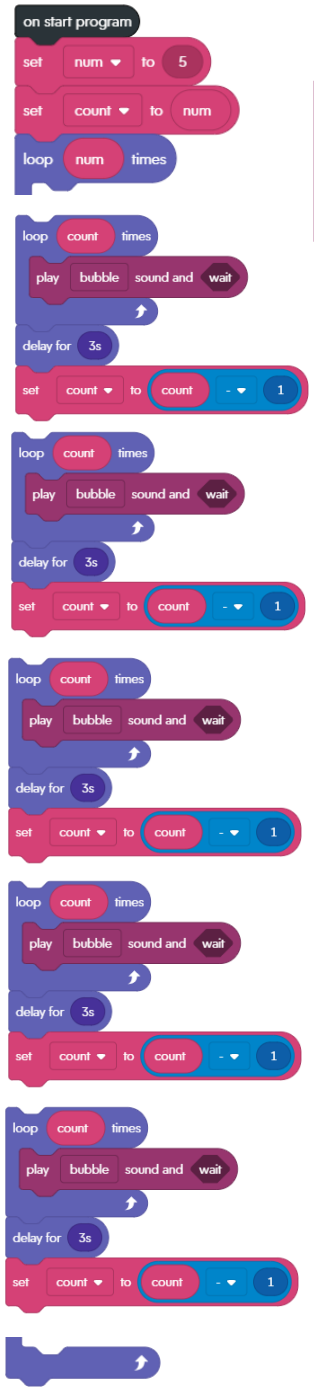
The outer loop counts down from 5 to 1.

The inner loop makes the “current count” of bubble noises

The Bubble Count Down



What does this line of code change?

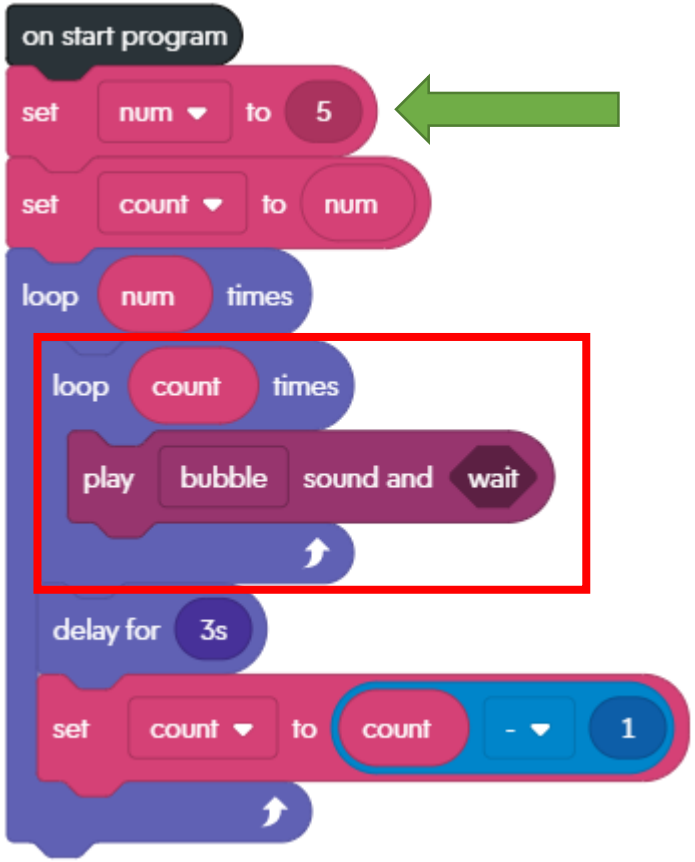


variables

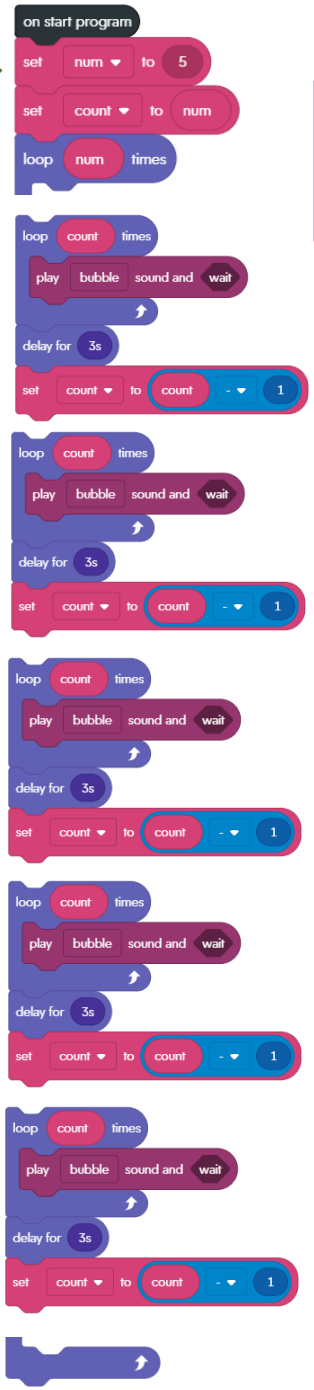
num ?

count ?

The Bubble Count Down

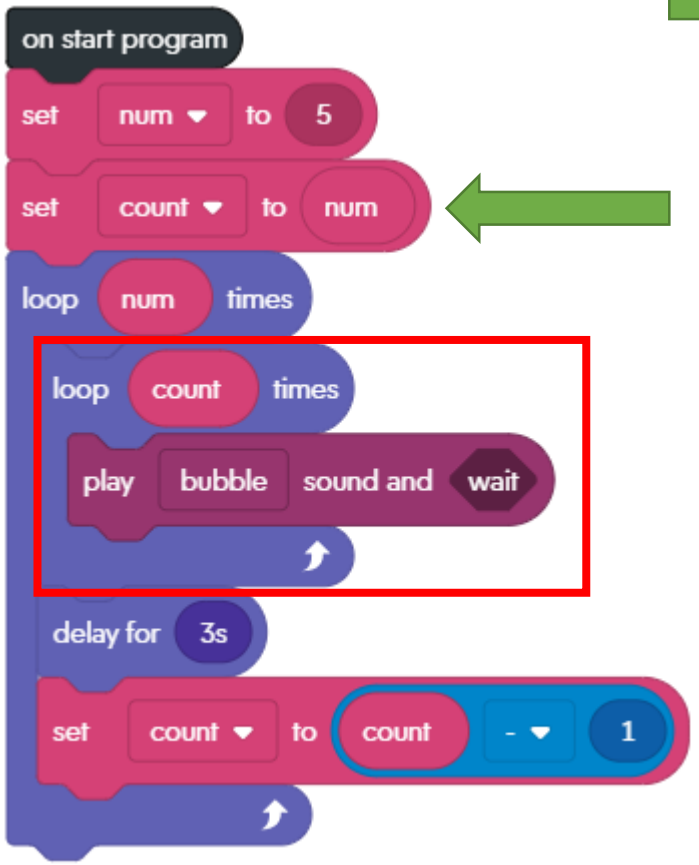


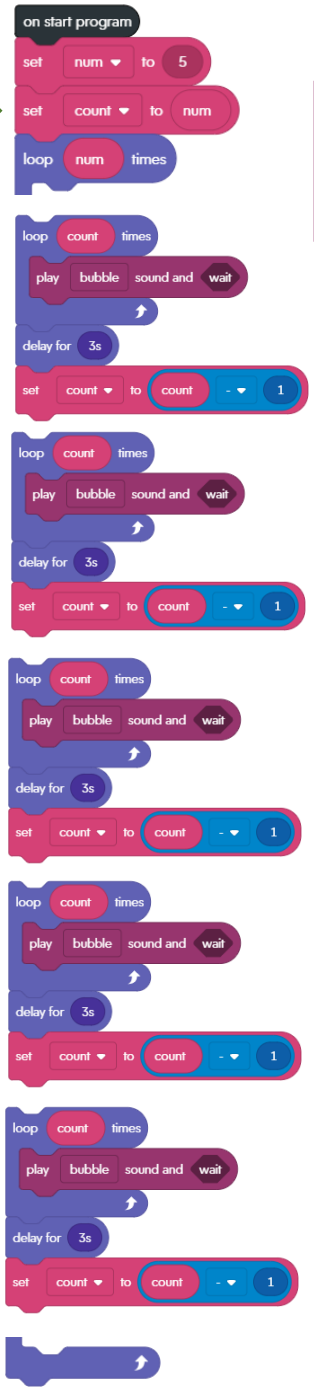
The num variable gets the value 5.



variables

The Bubble Count Down

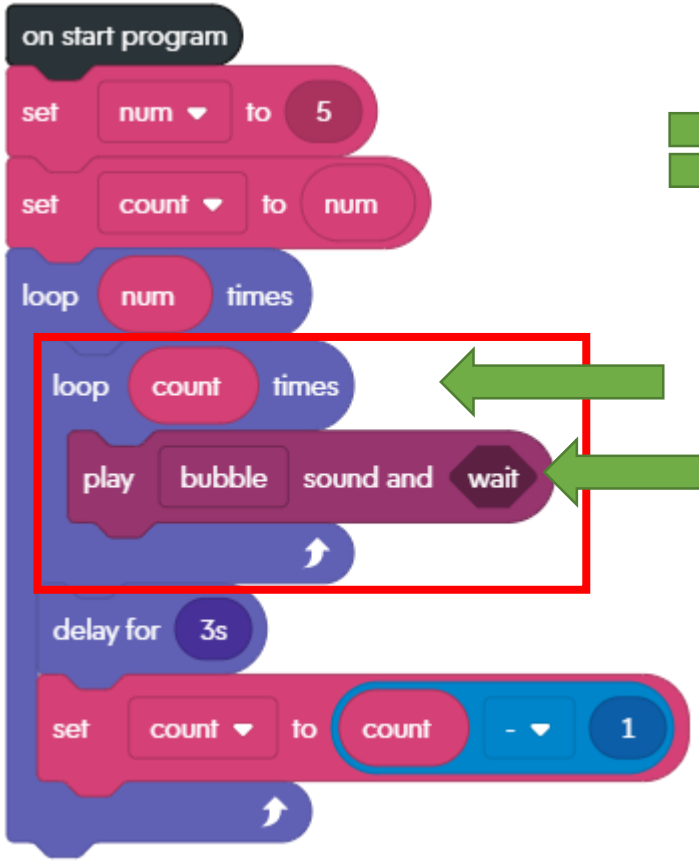


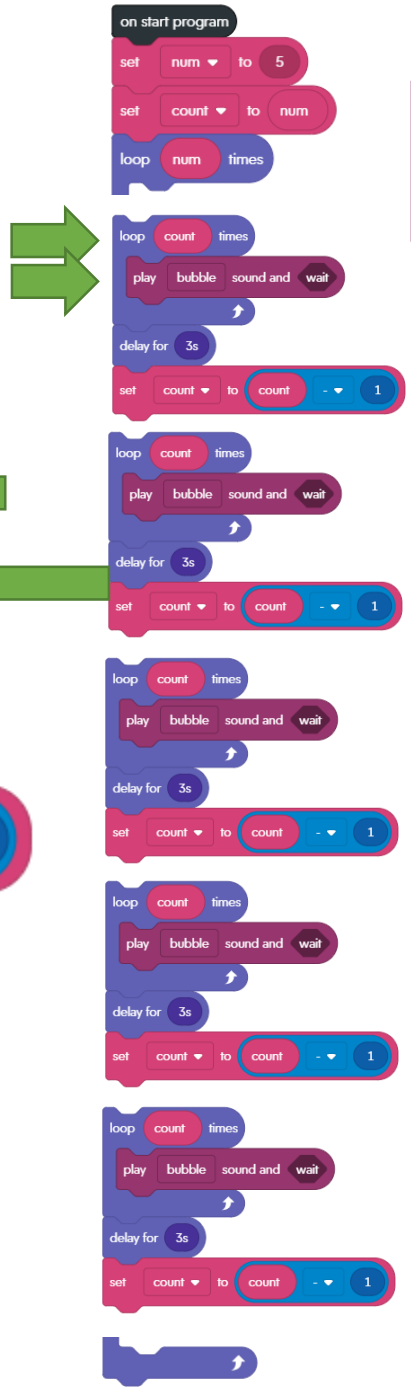


What does this line of code change?

variables

The Bubble Count Down

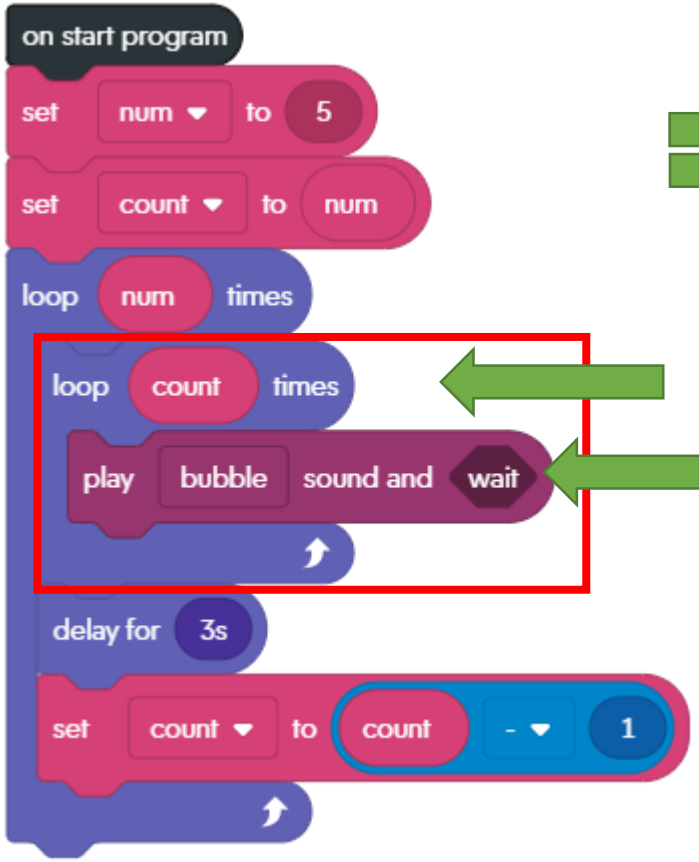


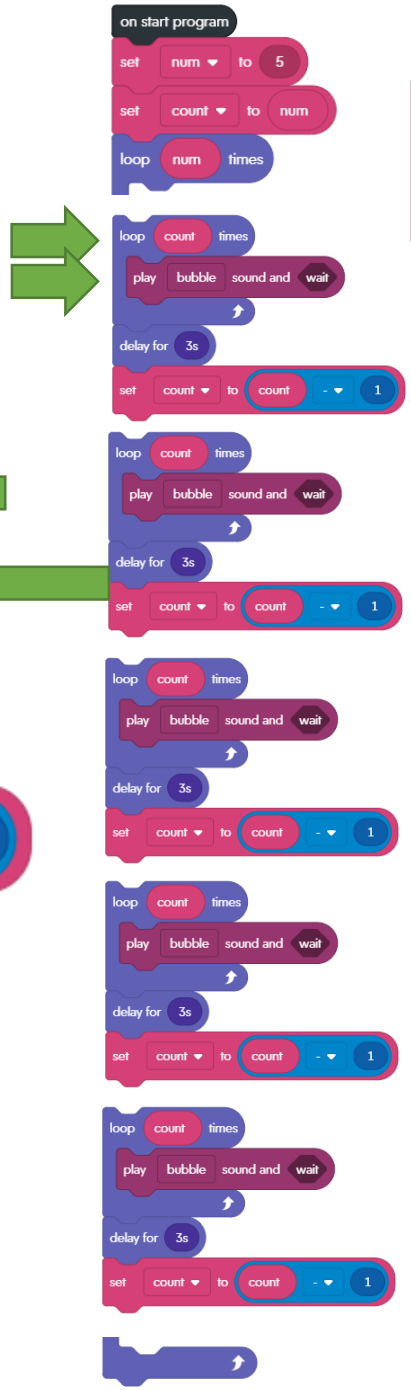


variables

What do these lines of code change?

The Bubble Count Down





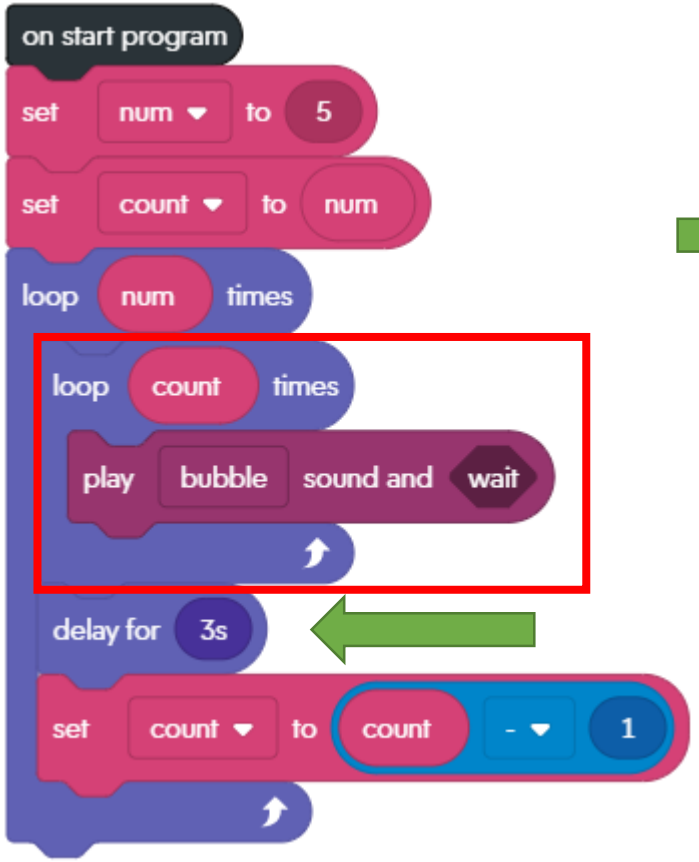
The loop makes 5 bubble sounds.

variables

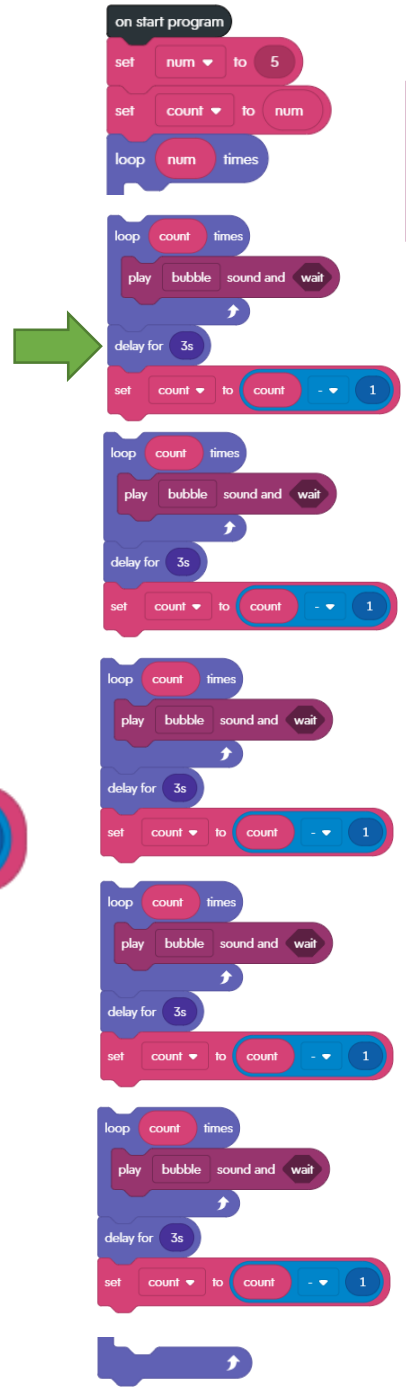
num 5

count 5

The Bubble Count Down



What does this line of code change?

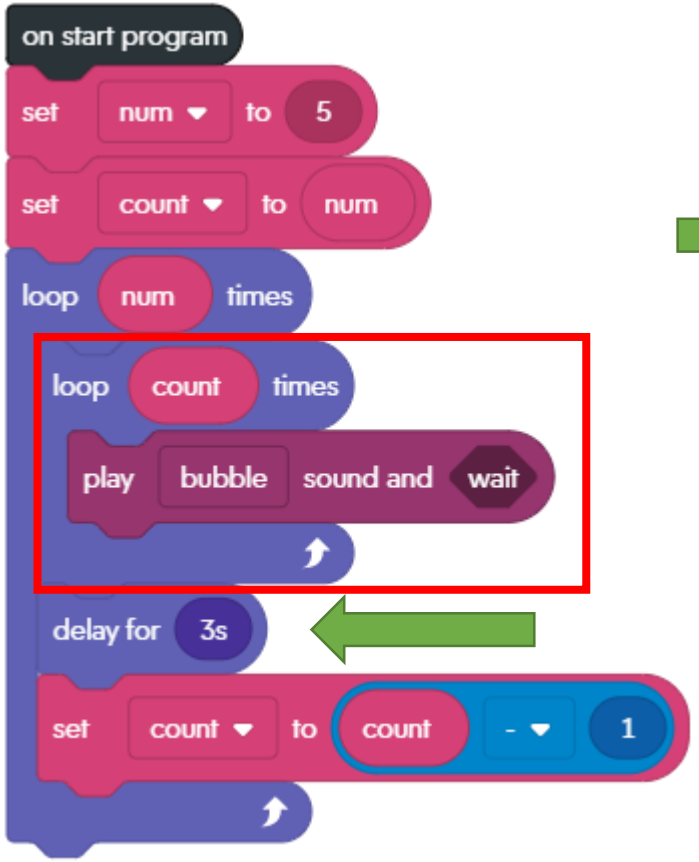


variables

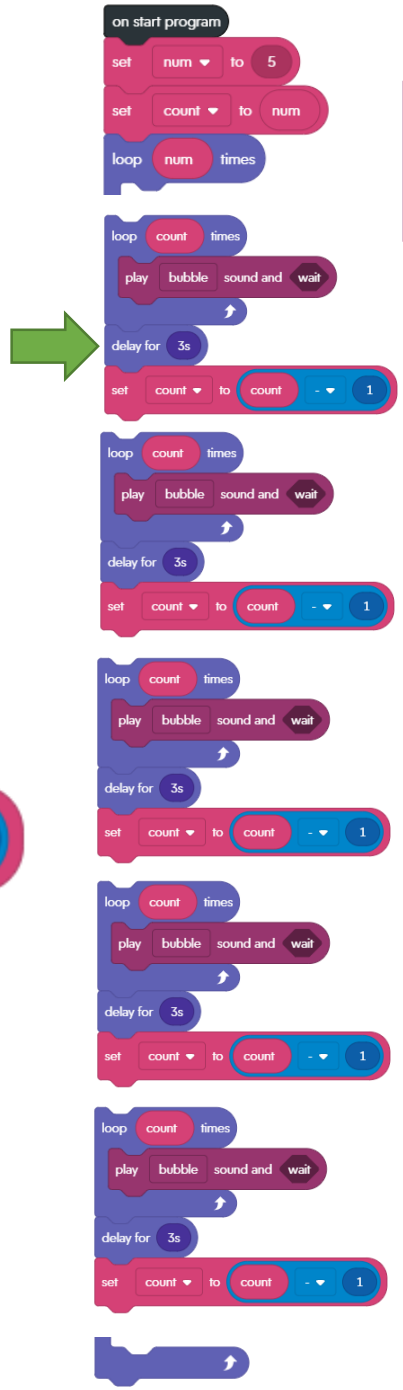
num 5

count 5

The Bubble Count Down



It causes a 3s delay to separate the bubble sounds.



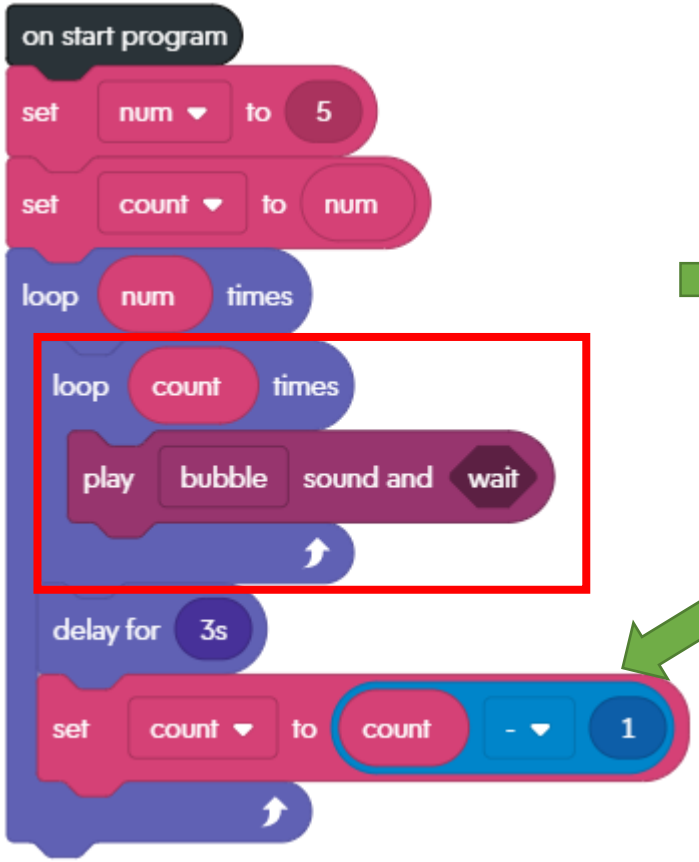
variables

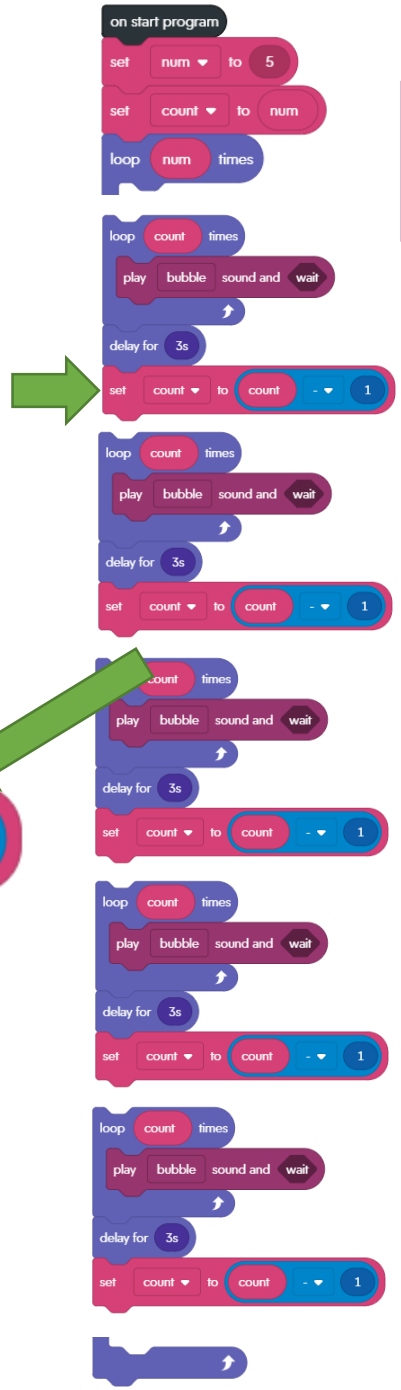
num 5

count 5

3s delay

The Bubble Count Down



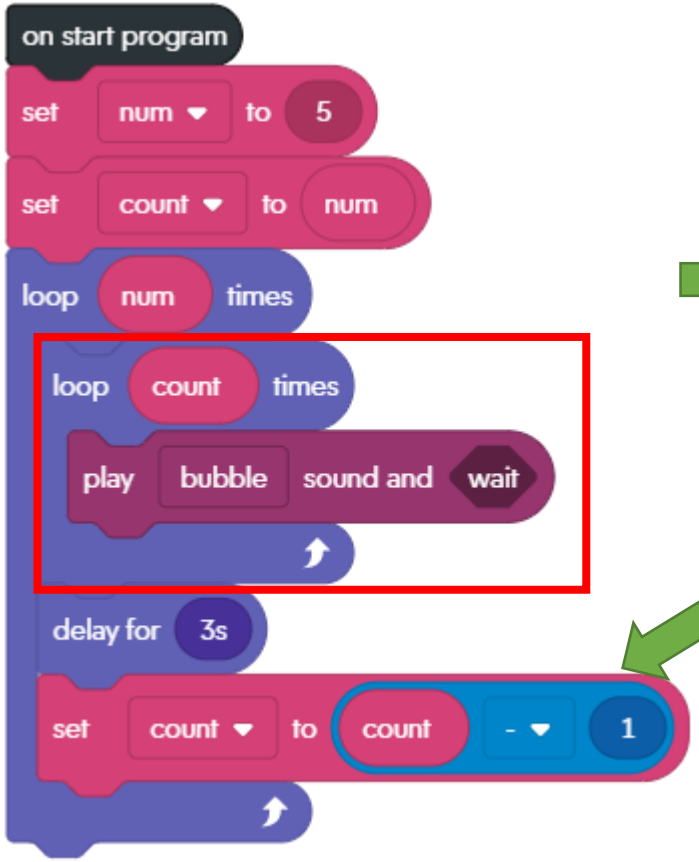


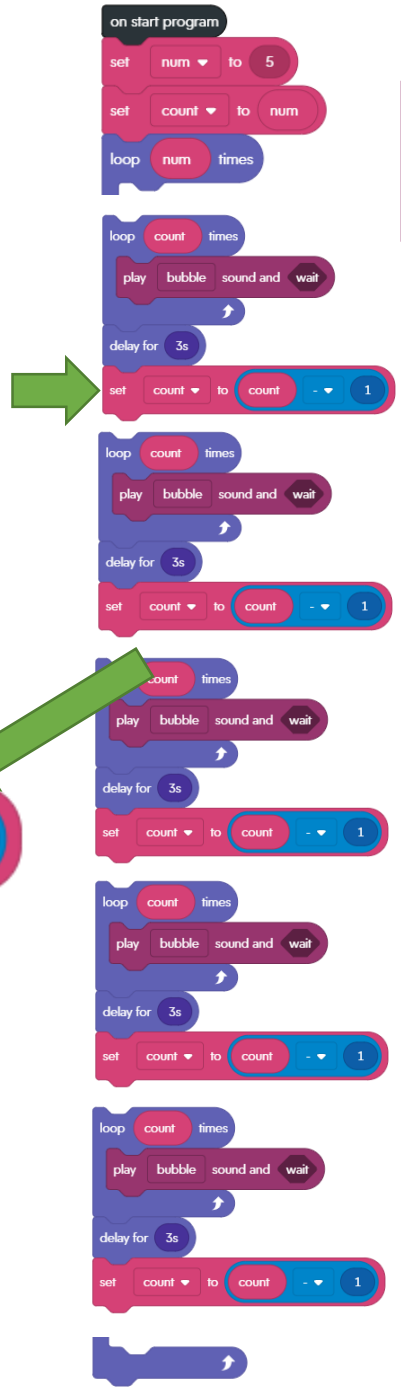
What does this line of code change?

variables

3s delay

The Bubble Count Down



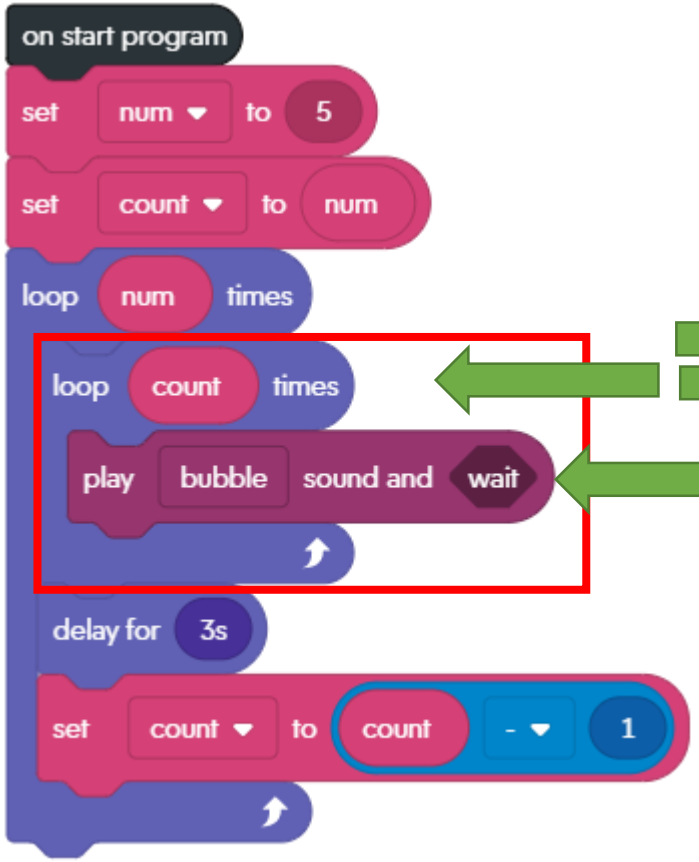


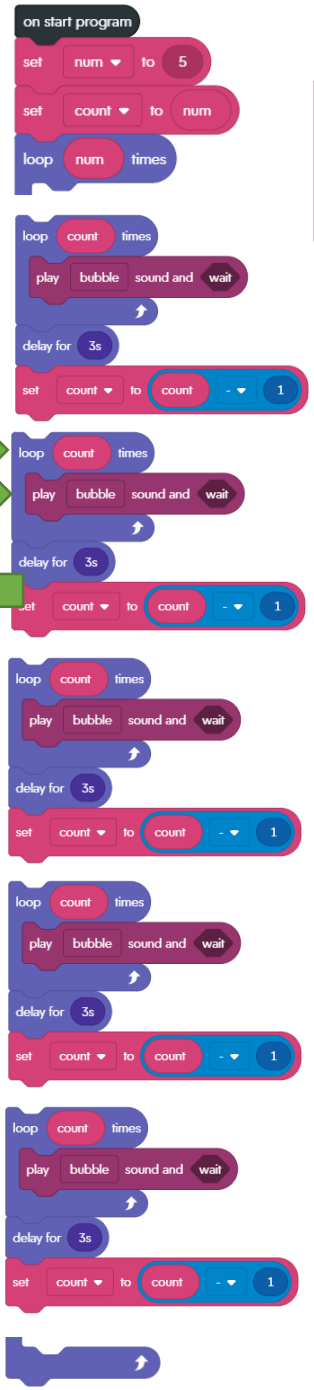
Count becomes 4.

variables

3s delay

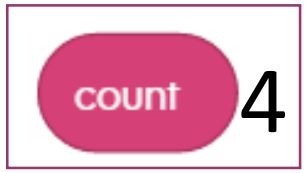
The Bubble Count Down





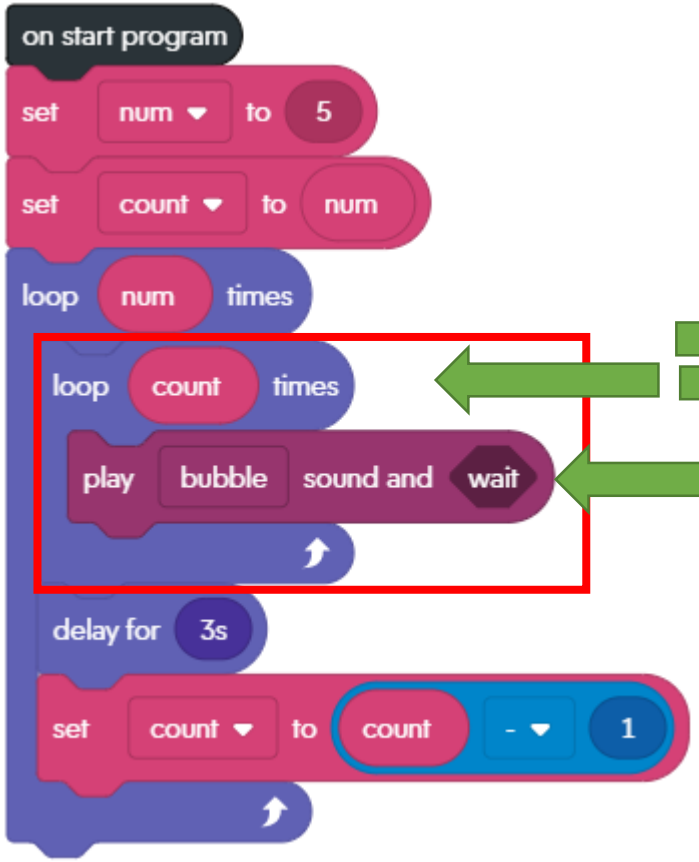
What does this line of code change?

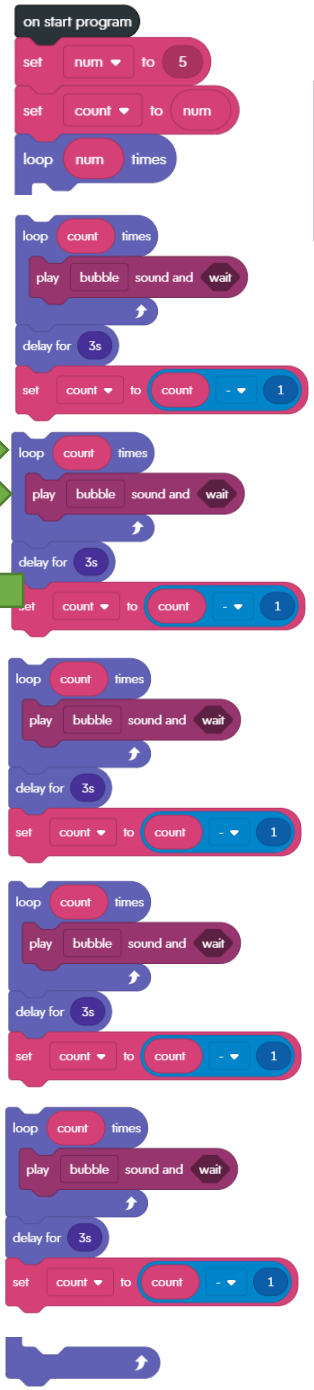
variables



3s delay

The Bubble Count Down





Now, only 4 bubbles are displayed.

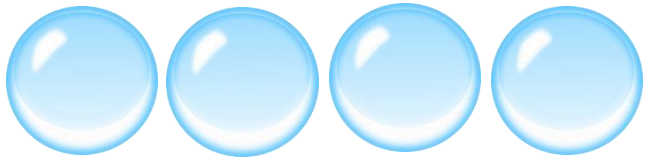
variables

num 5

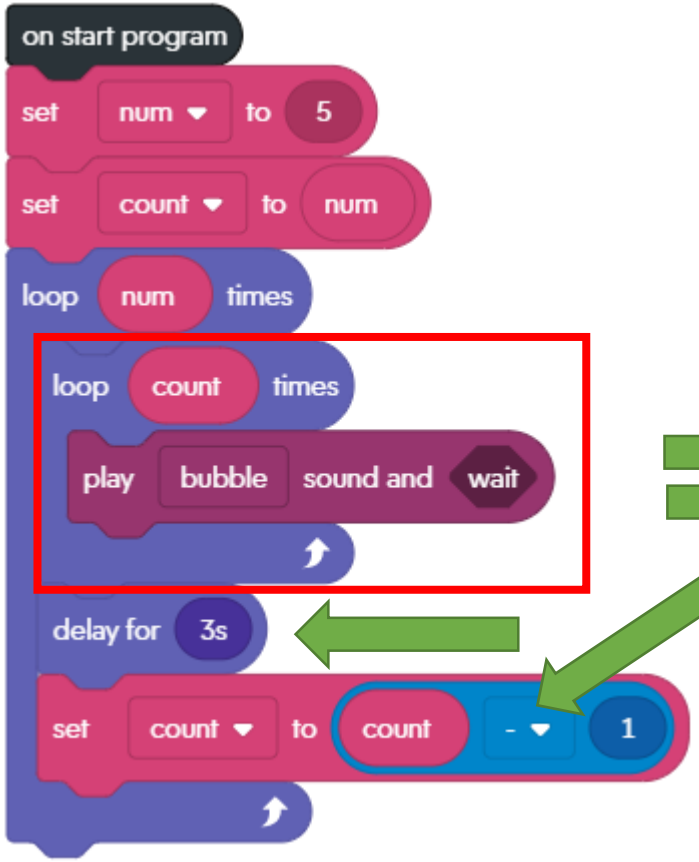
count 5

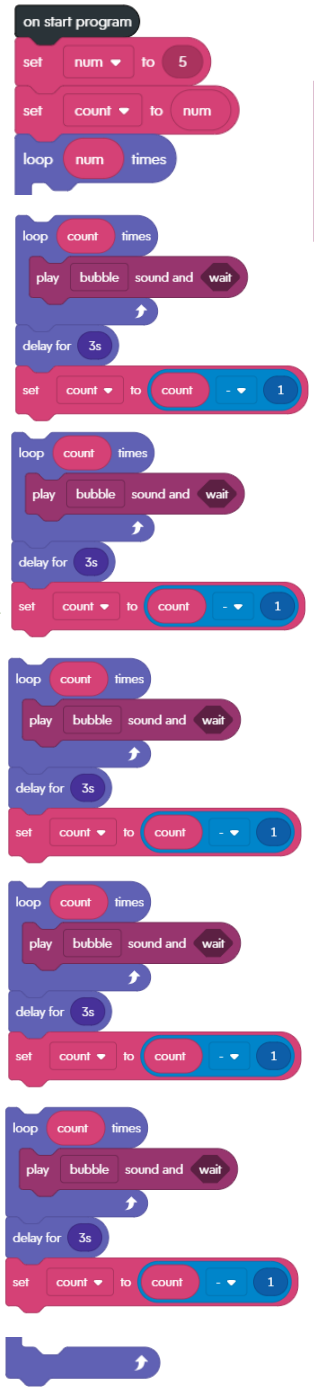
3s delay

count 4



The Bubble Count Down





What does this line of code change?

variables

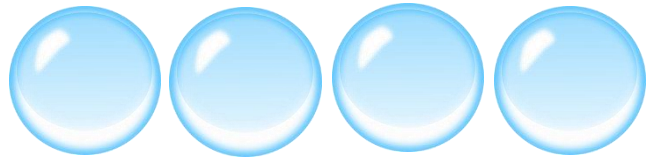
num 5

count 5

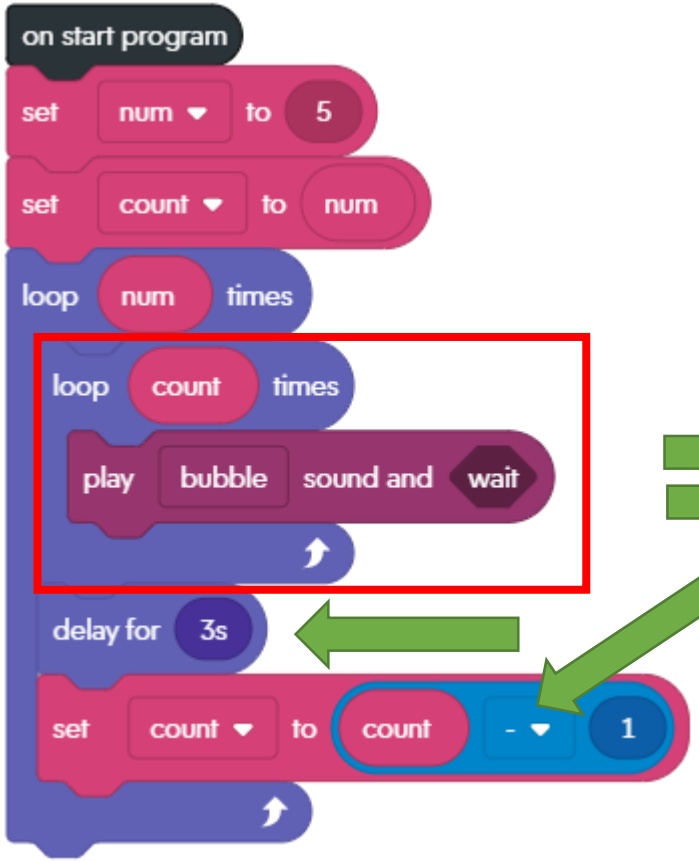
count 4

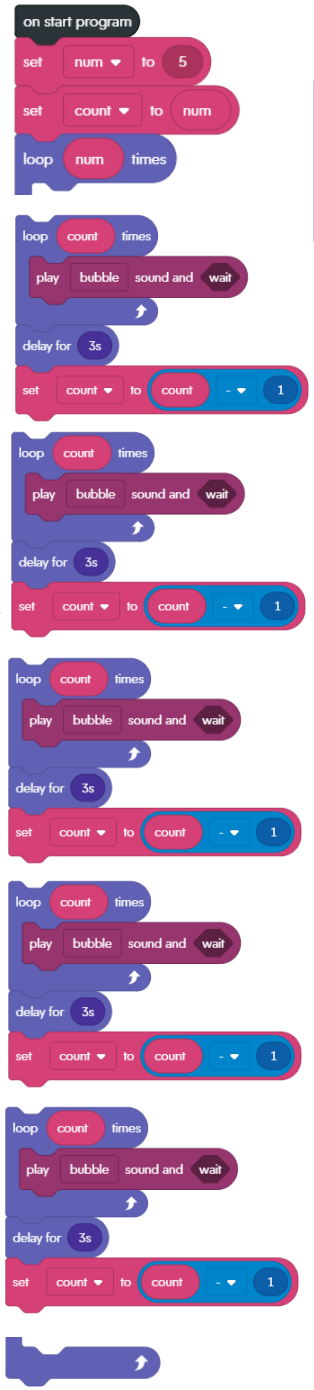
count ?

3s delay



The Bubble Count Down





3s delay and the count decreases.

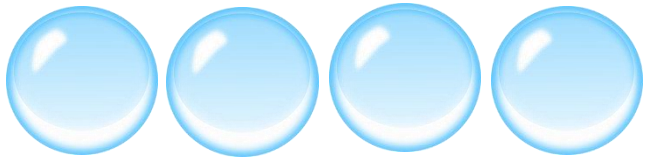
variables

num 5

count 5

3s delay

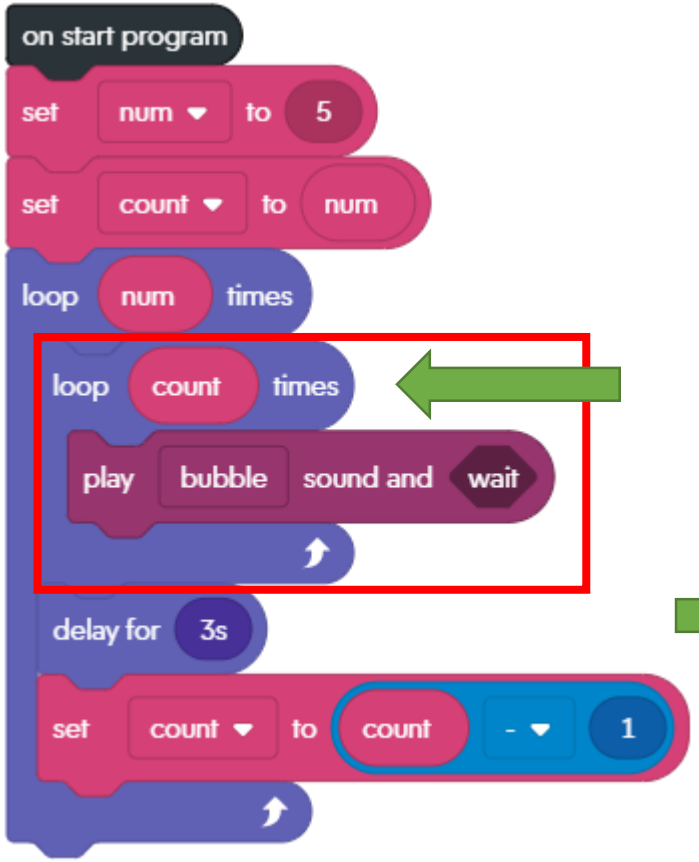
count 4

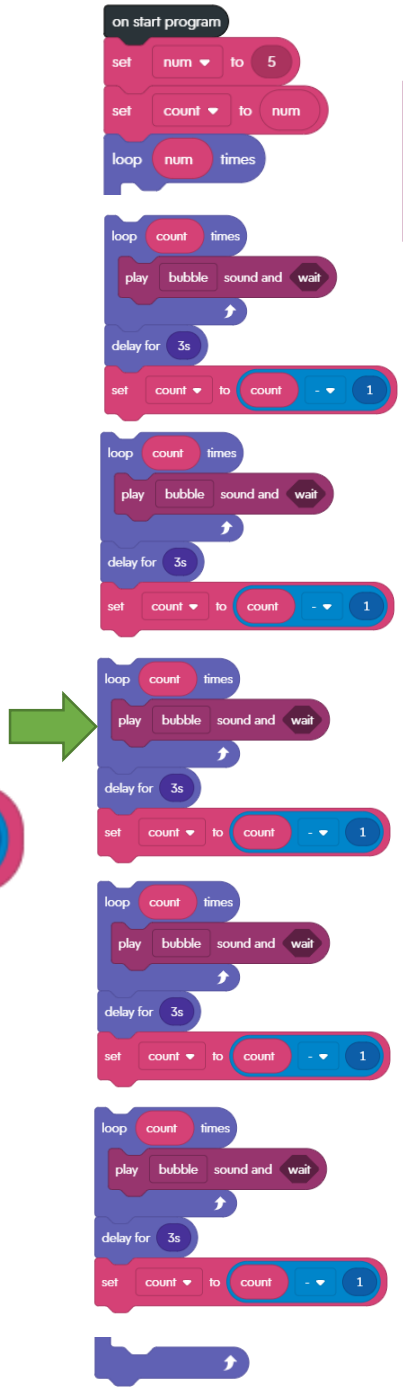


3s delay

count 3

The Bubble Count Down





What does this line of code change?

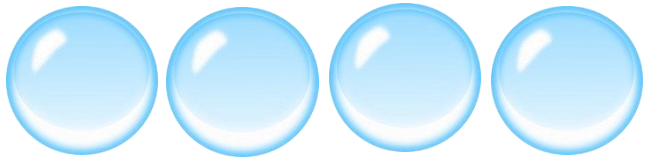
variables

num 5

count 5

3s delay

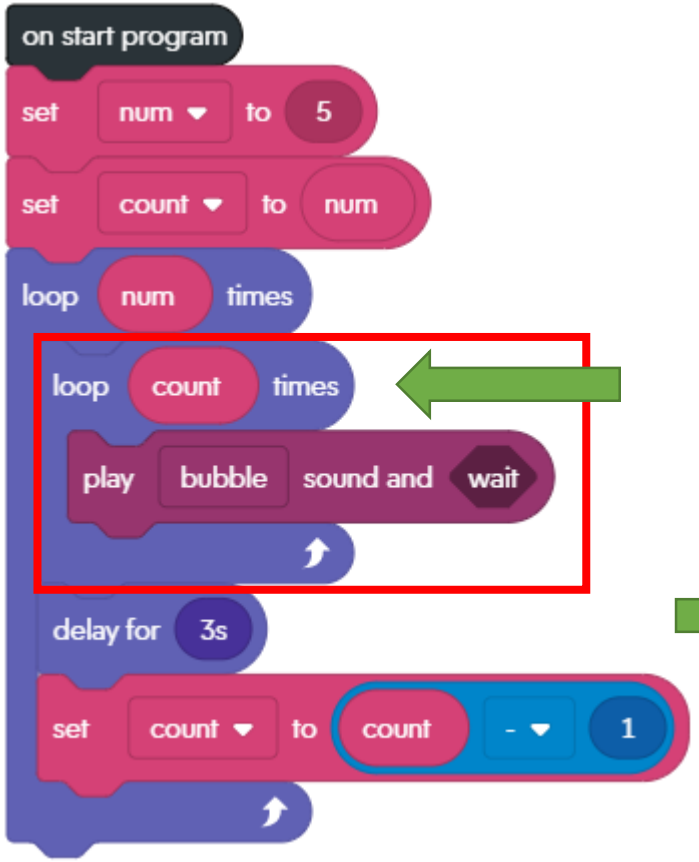
count 4



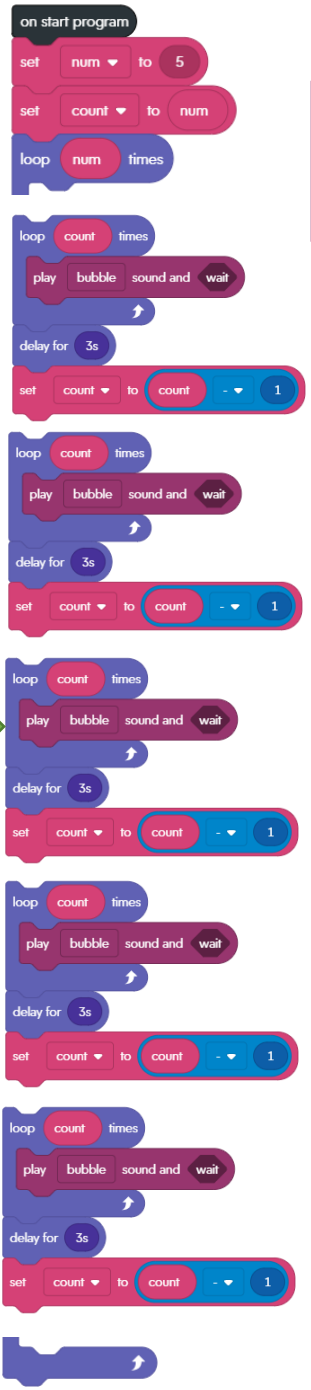
3s delay

count 3

The Bubble Count Down



3 bubble sounds this time in the loop.



variables

num 5

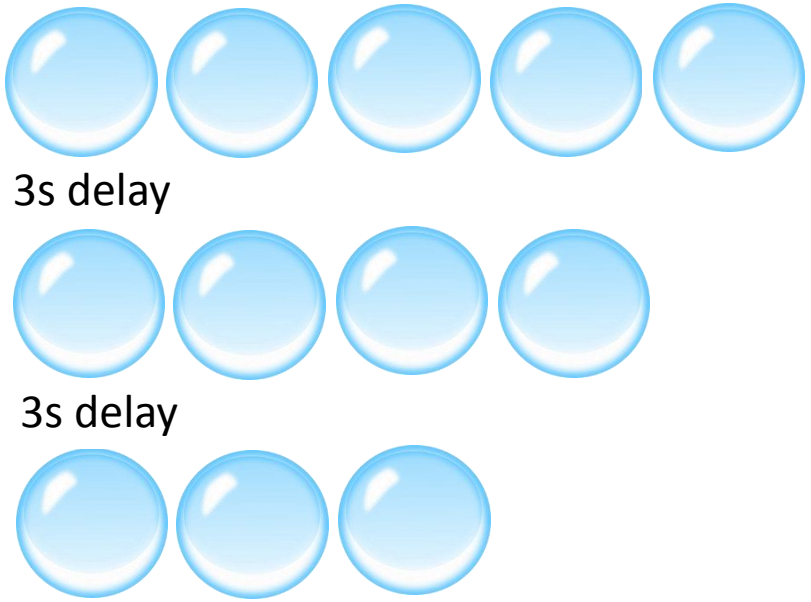
count 5

3s delay

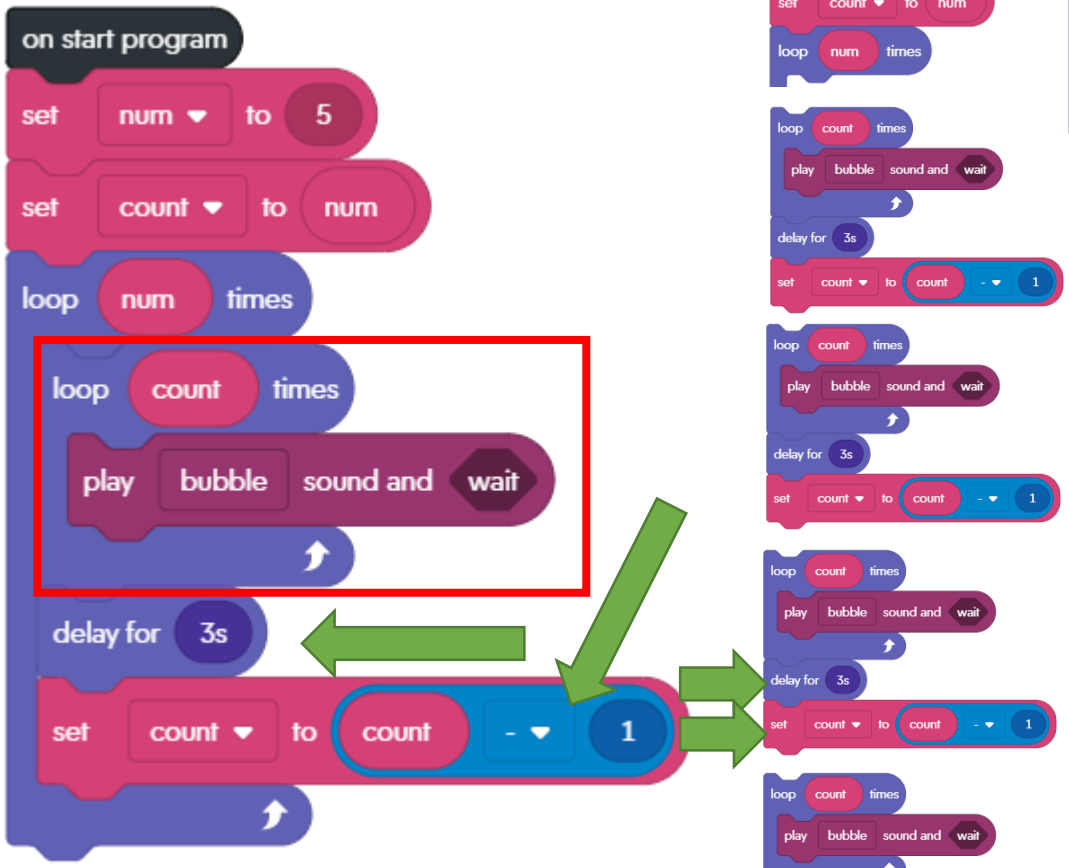
count 4

3s delay

count 3



The Bubble Count Down



What do these lines of code change?

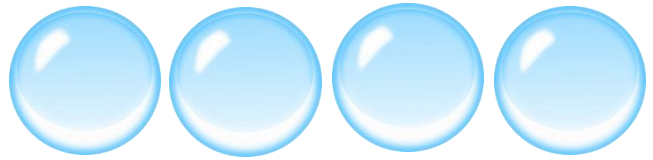
variables

num 5

count 5

3s delay

count 4

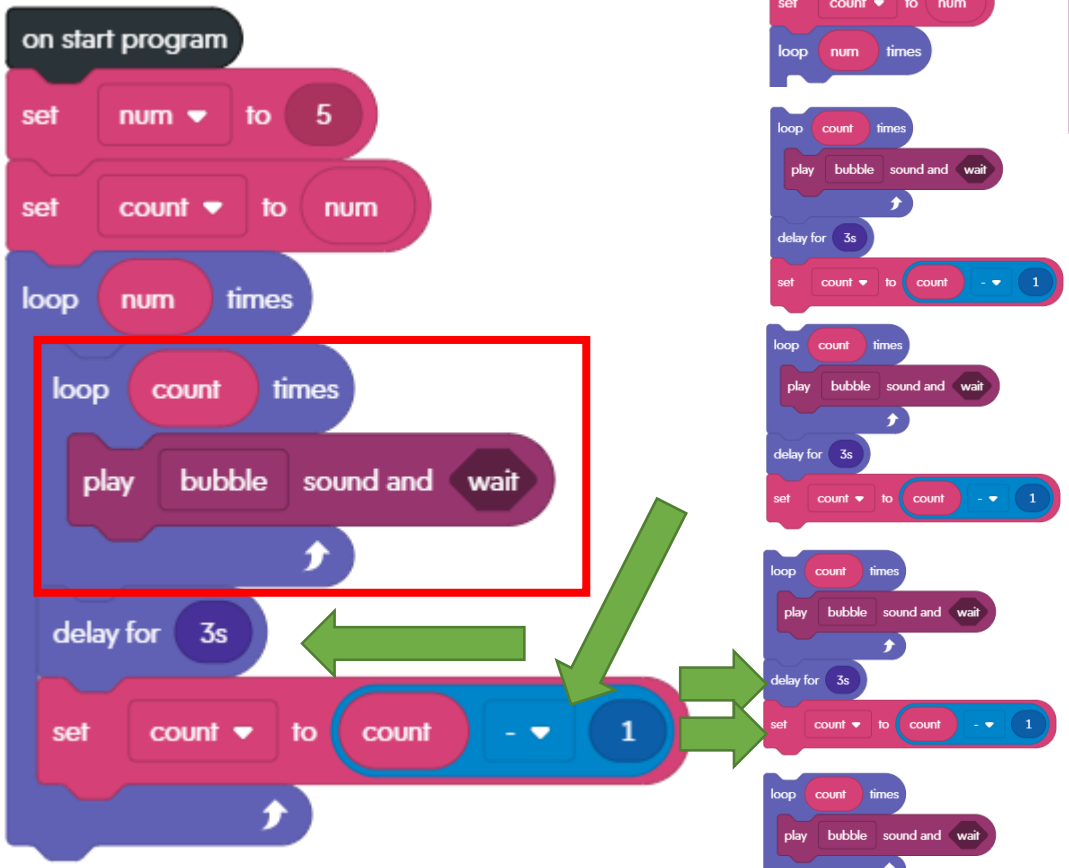


3s delay

count 3

count ?

The Bubble Count Down



3s delay and count decreases.

variables

num 5

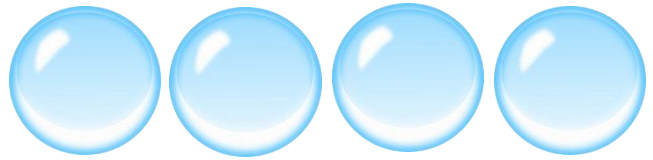
count 5

count 4

count 3

count 2

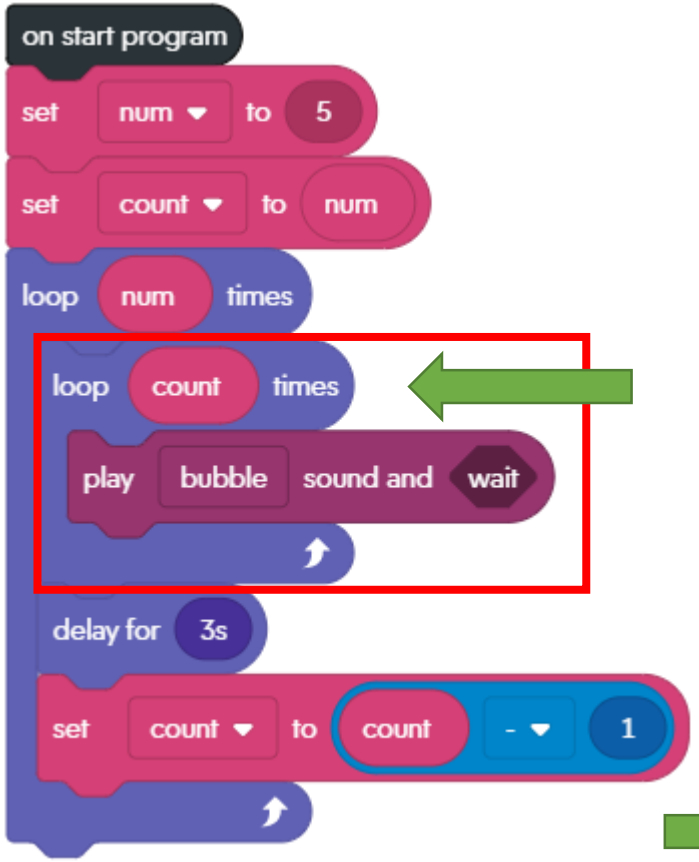
3s delay

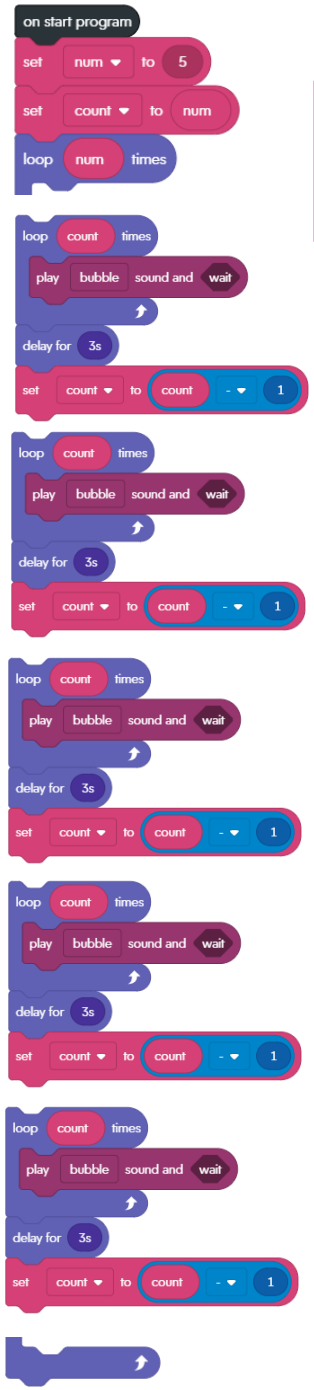


3s delay

3s delay

The Bubble Count Down





The 2 Bubbles are displayed.

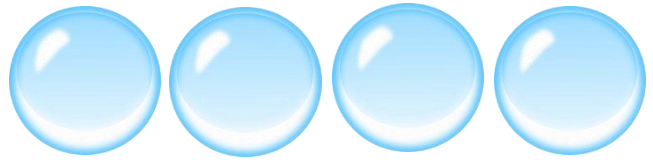
variables

num 5

count 5

3s delay

count 4

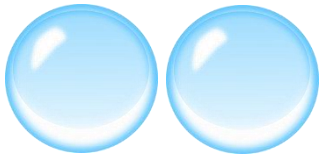


3s delay

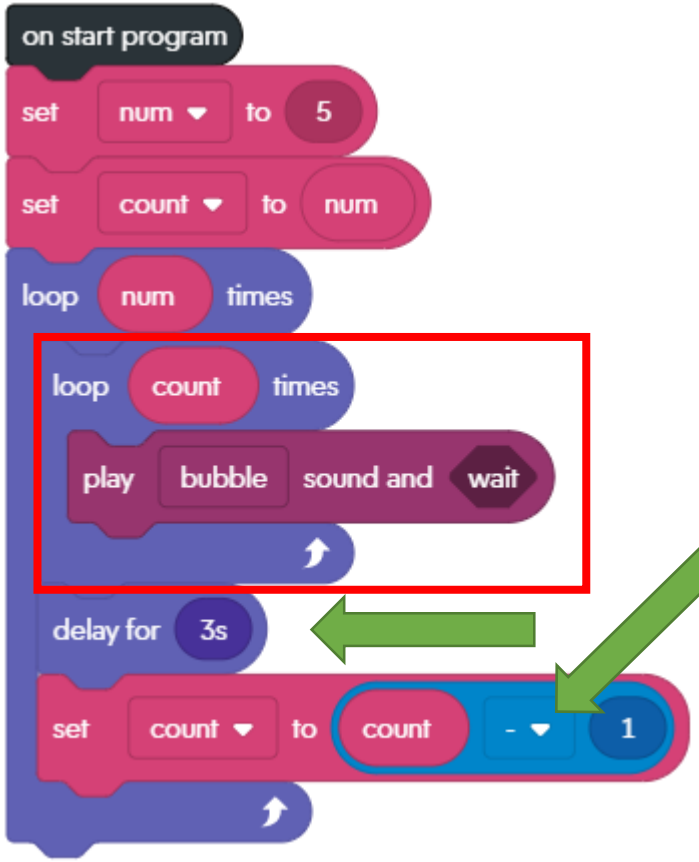
count 3

3s delay

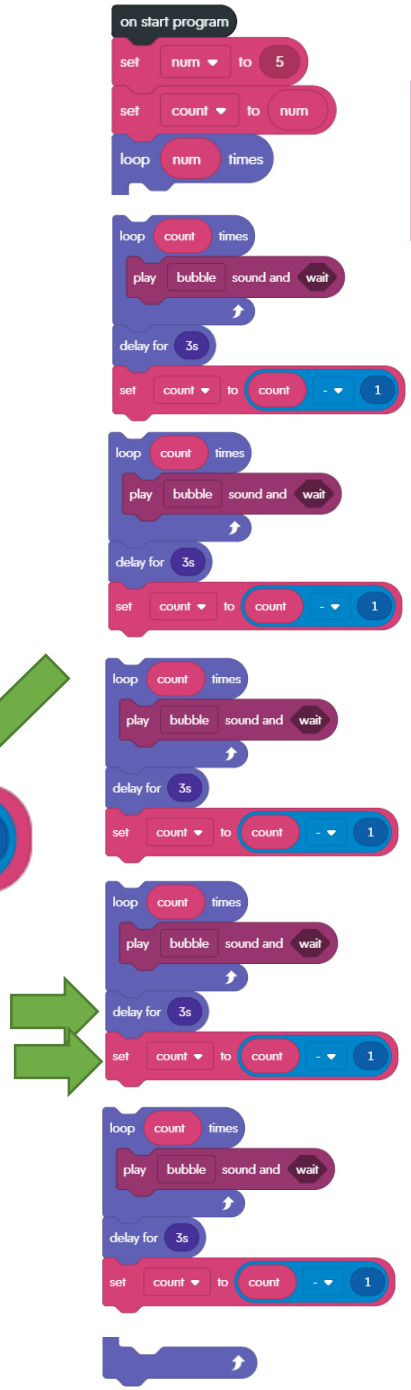
count 2



The Bubble Count Down



We delay and decrease count.



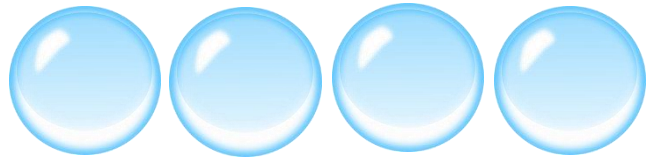
variables

num 5

count 5

3s delay

count 4

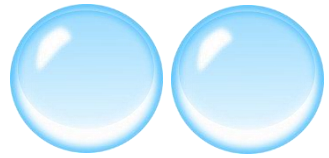


3s delay

count 3

3s delay

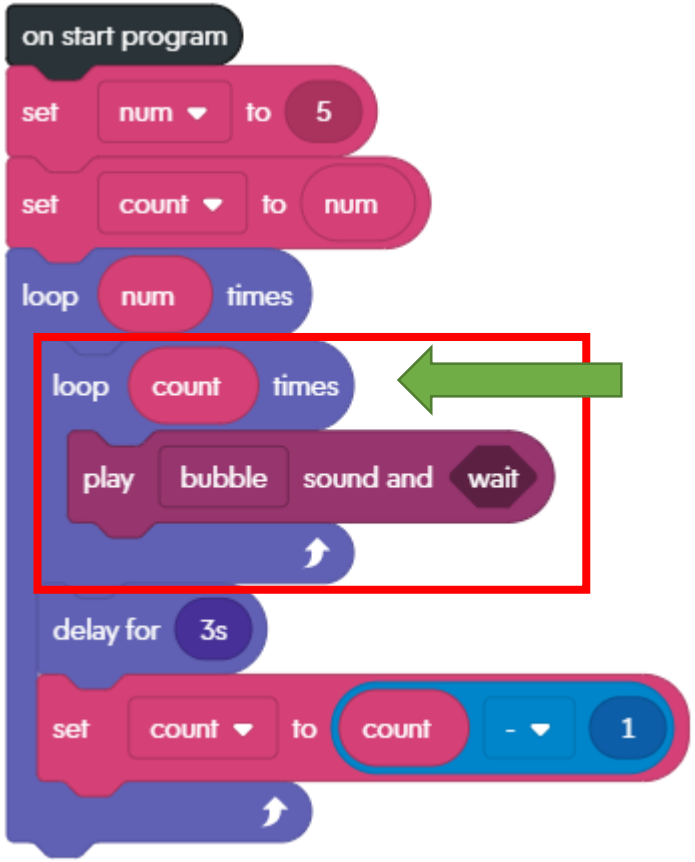
count 2



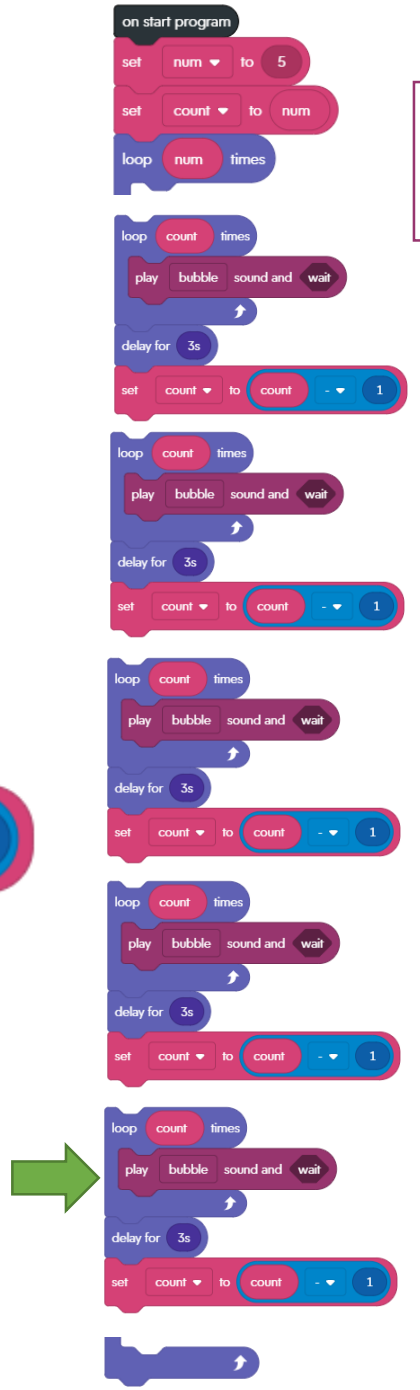
3s delay

count 1

The Bubble Count Down



The one bubble is displayed.



variables

num 5

count 5

3s delay

count 4

3s delay

count 3

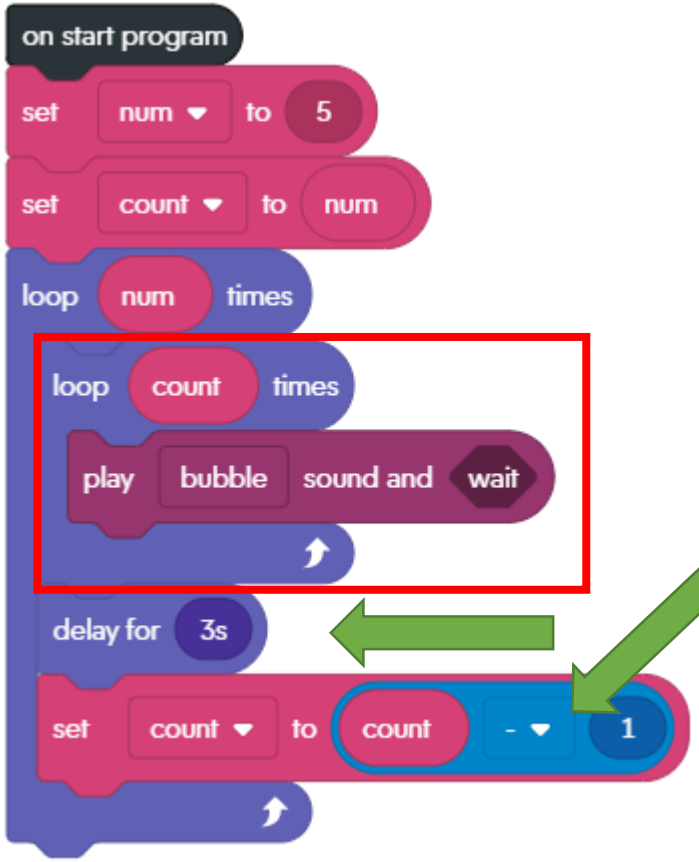
3s delay

count 2

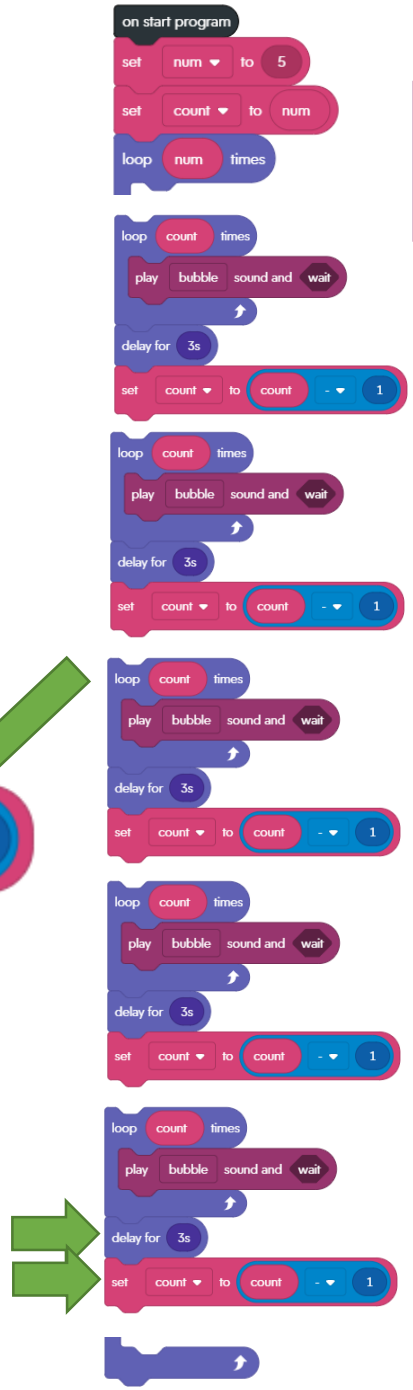
3s delay

count 1

The Bubble Count Down



We delay 3s and decrease count.



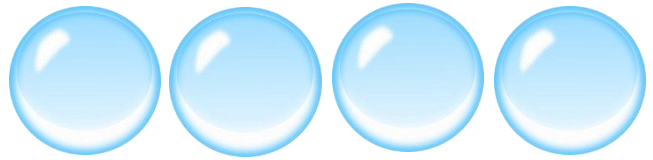
variables

num 5

count 5

3s delay

count 4

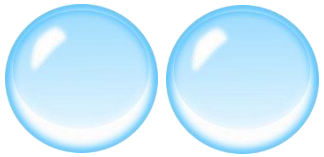


3s delay

count 3

3s delay

count 2



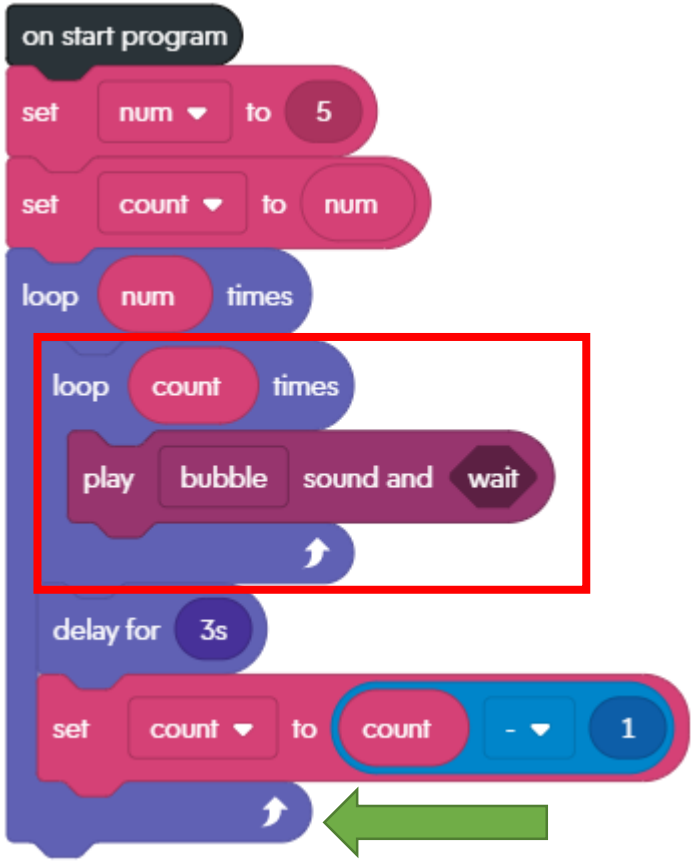
3s delay

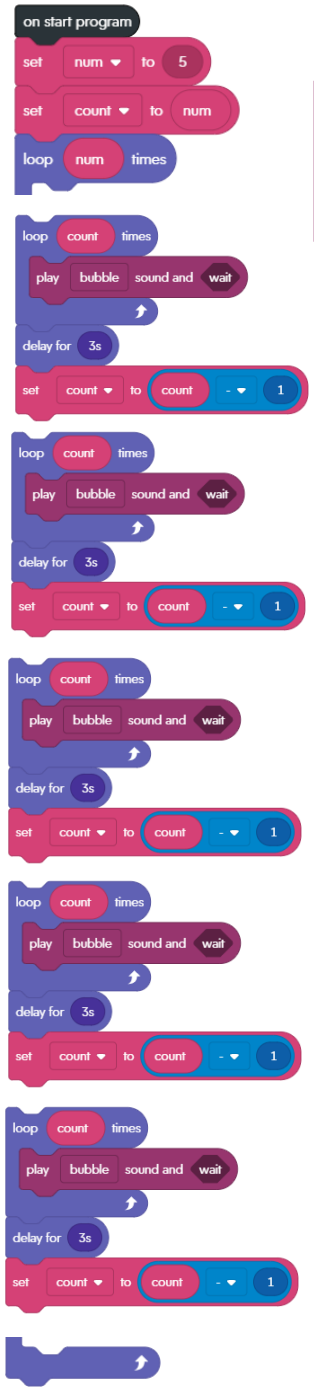
count 1

3s delay

count 0

The Bubble Count Down





Both loops are now finished

variables

num 5

count 5

3s delay

count 4

3s delay

count 3

3s delay

count 2

3s delay

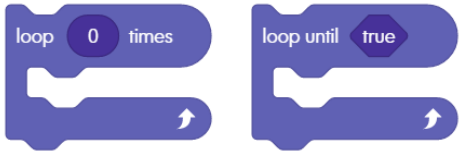
count 1

3s delay

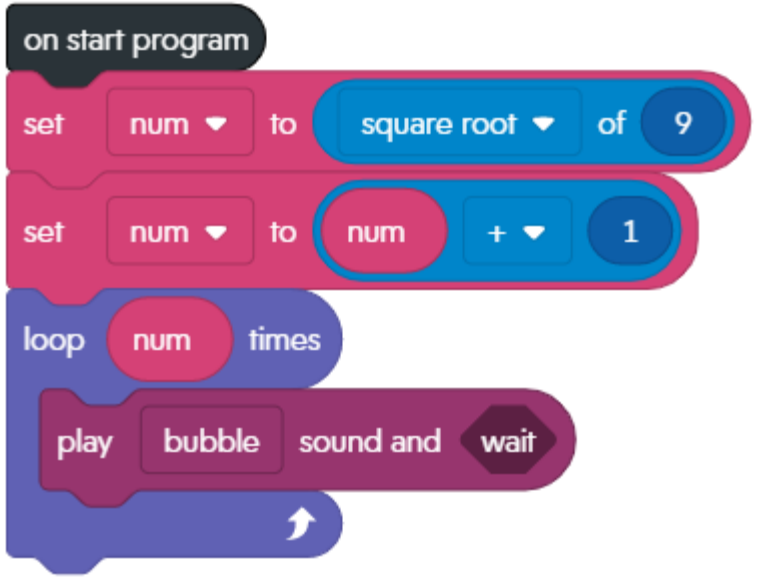
count 0

All done

Basic Coding Components

Output	*Changes the robot:* Moves. Spins. Re-directs to a new angle. Makes sound. Speeds up. Lights up.	
Math	*Calculation that results in a number:* +, - , *, /, square root	
Boolean Expressions	*Calculation that results in **true or false**:* >, <, =, >=, <=, and, or, not	
Control	Ifs — ***Decides** which piece of code to run:* Uses a Boolean expression and output or math.	
	Loops — ***Repeats** code:* Uses a Boolean expression and output or math.	
Variables	Named pieces of memory where you can store things to use later OR to store calculation results.	
Functions	Named pieces of code where you can group code together to use it later.	

In this code, how many bubble sounds are played? Make 2 new math equations using other functions to result in 8 bubble sounds being played.

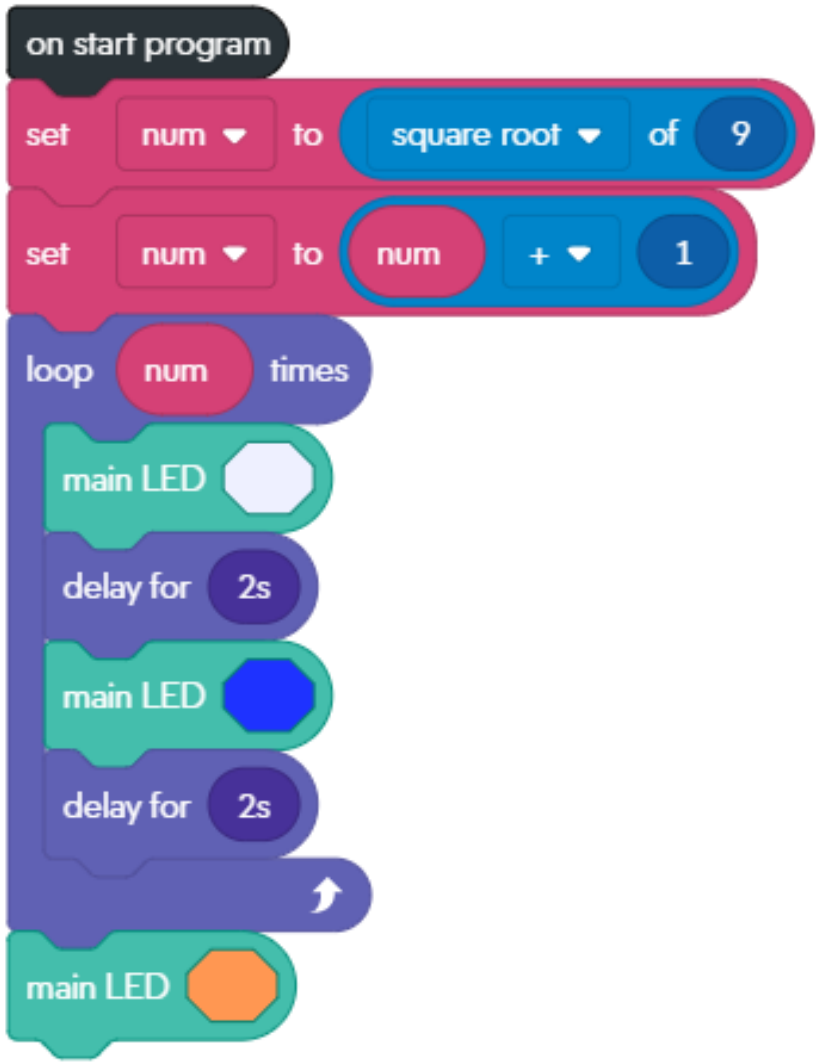


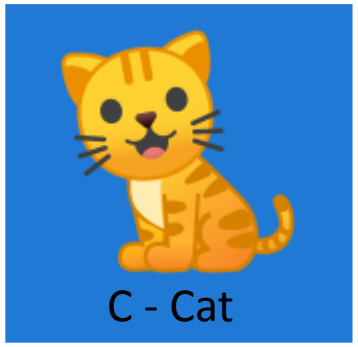
B - Bat

Run this code and figure out what it does.

Make two new math equations that make it run 7 times. Make the colour sequence inside the loop red, green, blue.

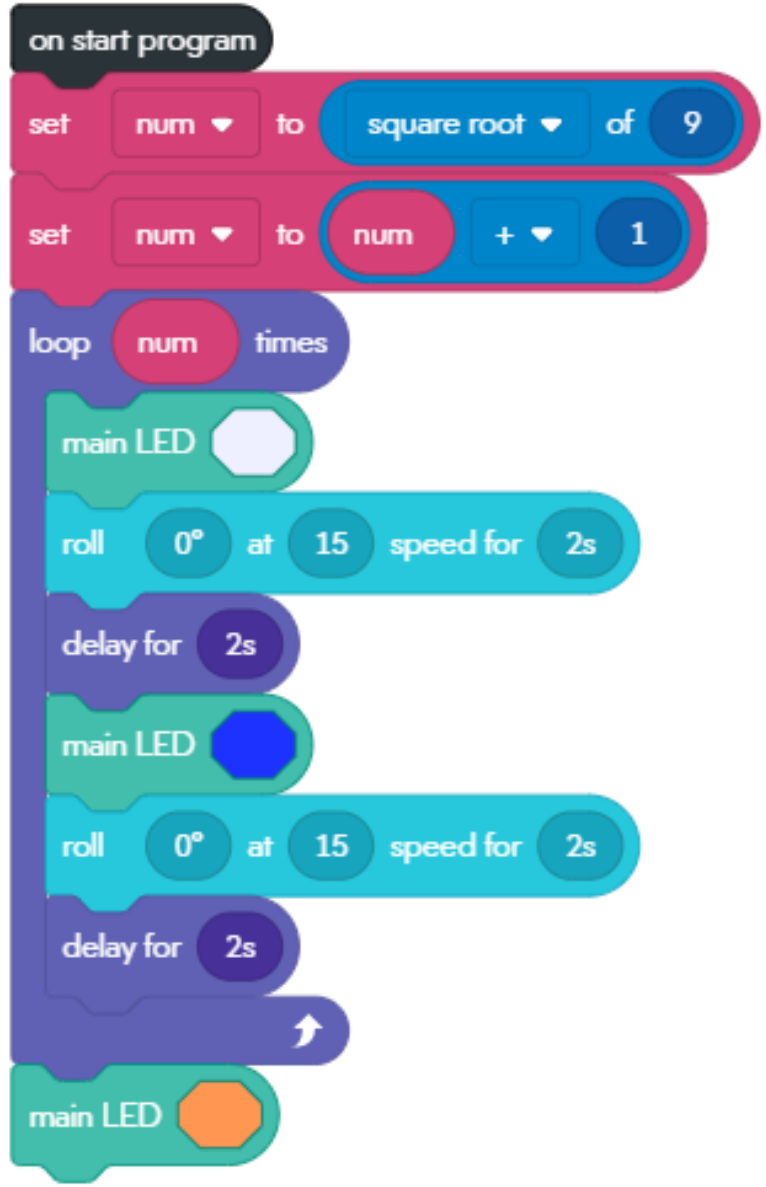
Test it on your Sphero.

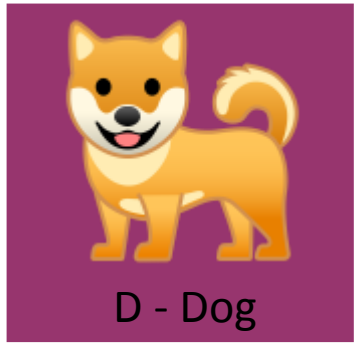




Run this code and figure out what it does. (Put the Sphero on the ground first)

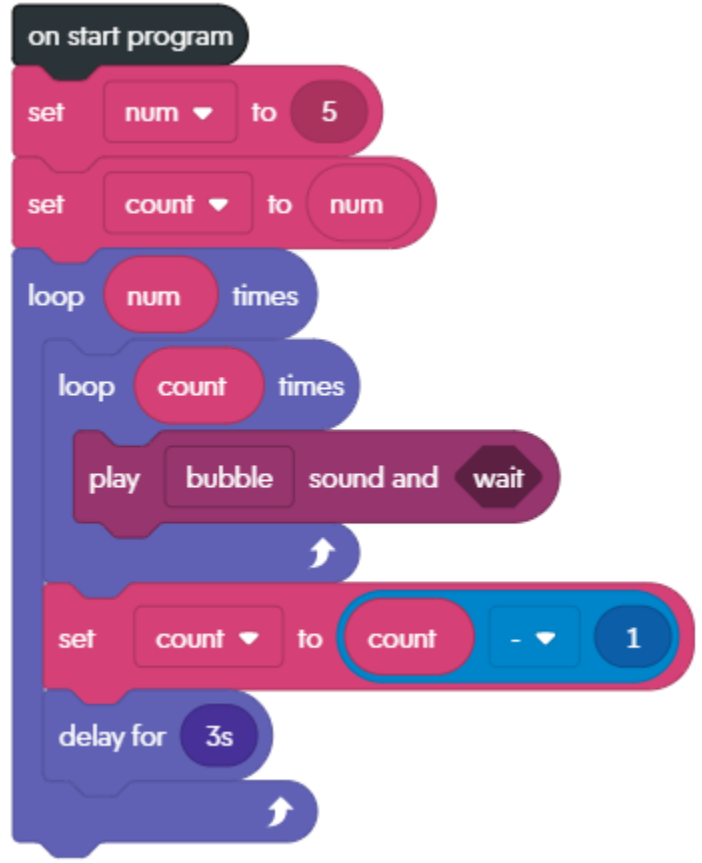
Make two new math equations that make it run 5 times. Make the Sphero roll in a zig-zag pattern.

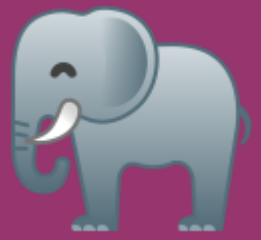




Make the Bubble sounds (or whatever you choose) count up instead of counting down.

Test it on your Sphero.

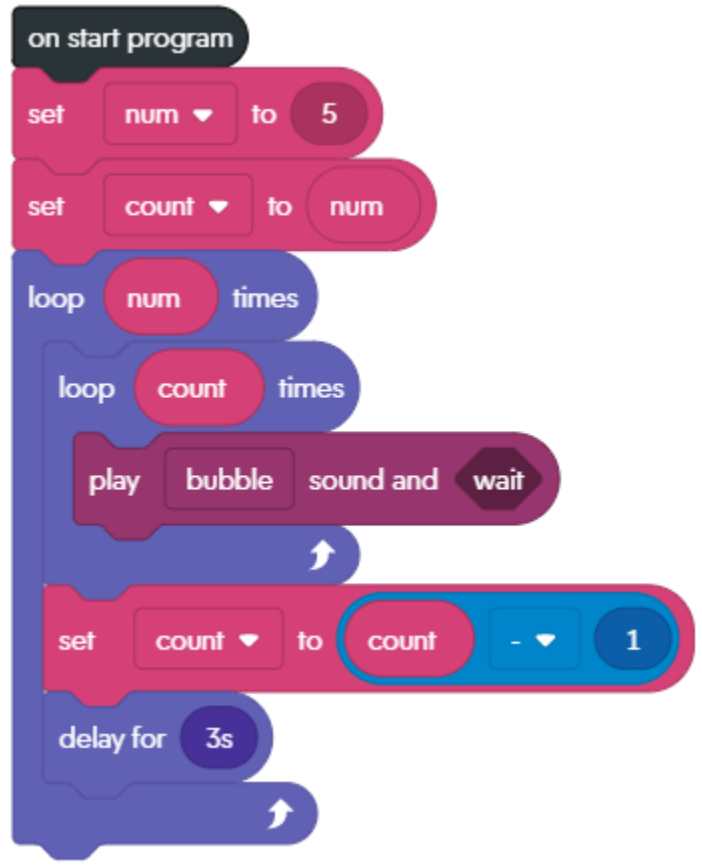




E - Elephant

Make the Bubble sounds (or whatever you choose) start at 12 and count down by 2s.

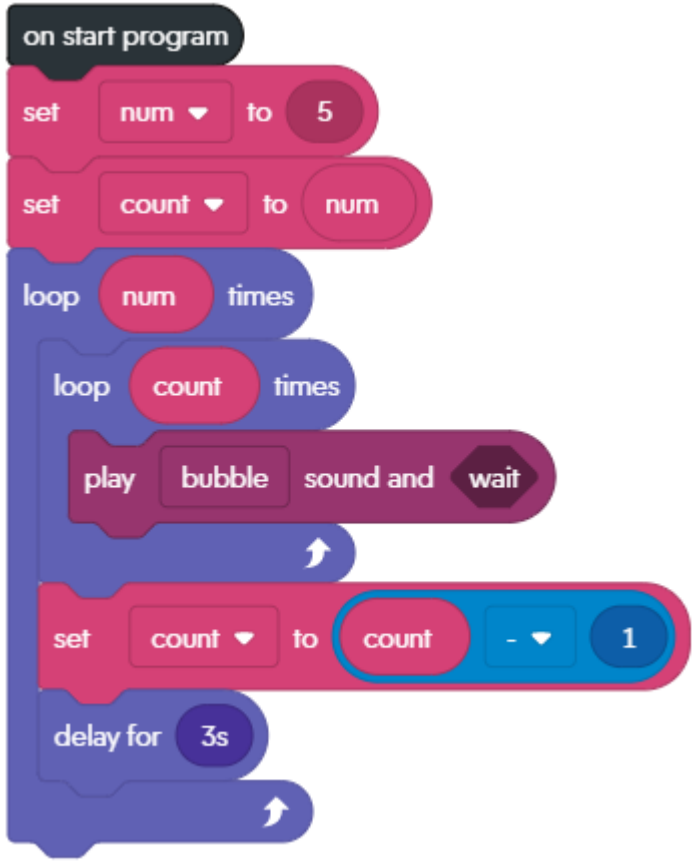
Test it on your Sphero.



F - Fish

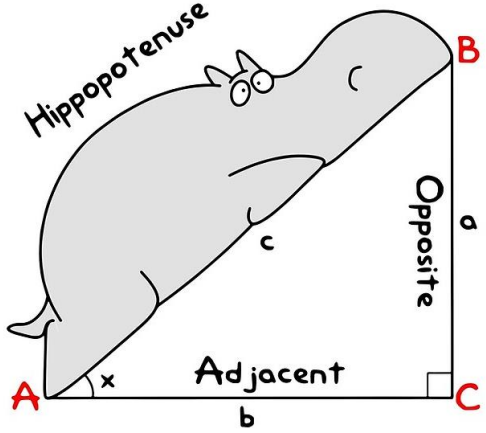
Make the Bubble sounds (or whatever you choose) start at 3 and count up to 15 by 3s.

Test it on your Sphero.



G - Giraffe

Write a program that calculates the size of the hypotenuse of a triangle.



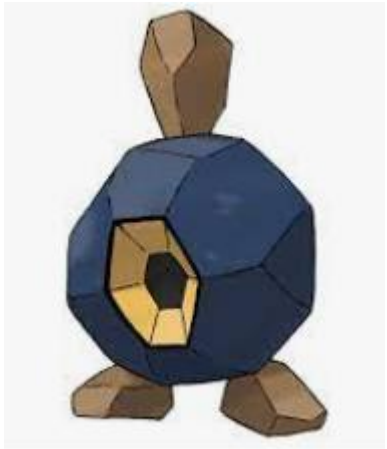
$$c = \sqrt{a^2 + b^2}$$

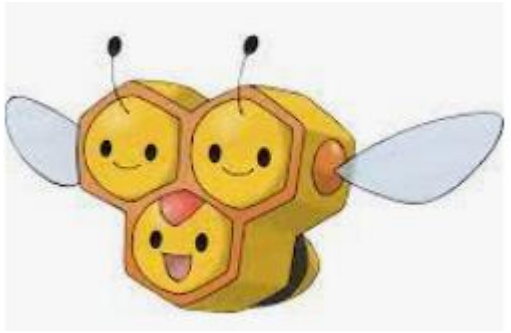
H - Horse

Write a program that counts down from ten and then “blasts off”. You can do this with a loop. You can also start the Sphero moving after the “launch sequence”.

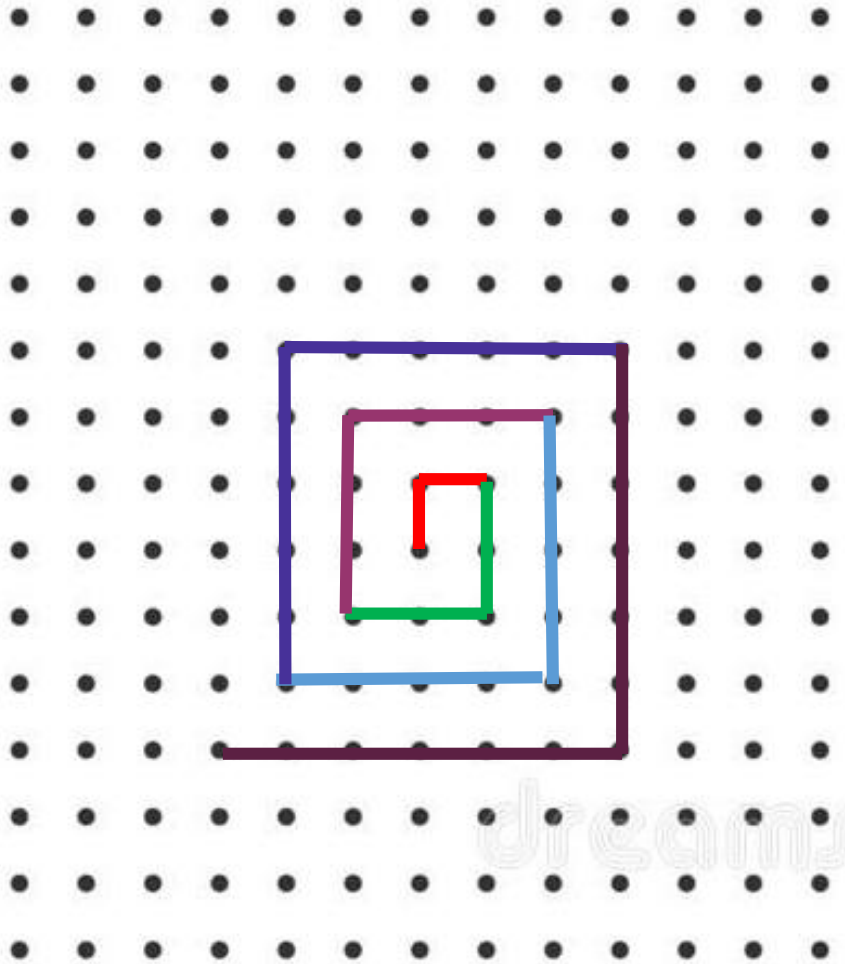
Test it on your Sphero.

Write a Sphero program that creates an hexagon. This can be done using a variable for the angle.

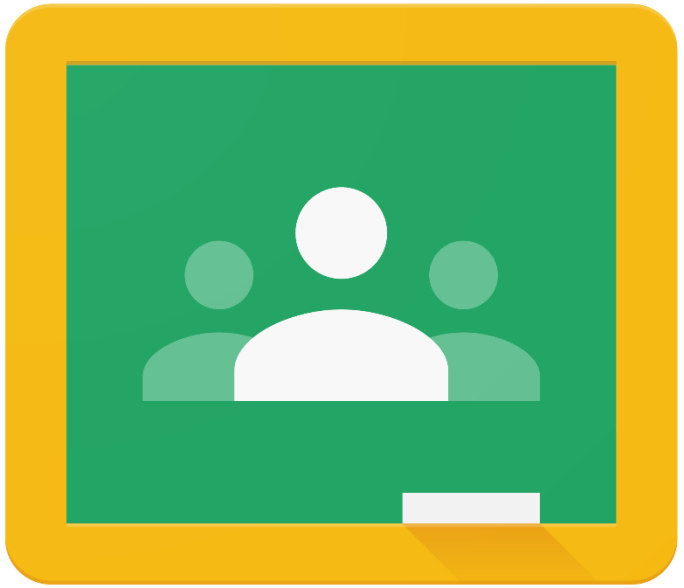




Write a program that allows the Sphero to travel in a spiral pattern like this Anishinaabe basket.



When you are done, there are check-your-understanding questions on Google Classroom.



Google Classroom