05 Music, Designs, & LEDs (Loops)

This lesson introduces the concept of looping and iteration. In the lesson the concepts of using pins for buttons, connecting headphones or speakers to output music or sound, and the ability to connect external LEDs to the microbit through the external pins are introduced. By understanding how to connect external devices to the microbit it opens up a whole new world that can lead to all kinds of ***innovation.*** *Think of self driving cars that take inputs from sensors, process the data, and output the results by steering, driving, and braking the car. Smart homes with sensors, processors, and lights coming on or the heat being controlled are different kinds of innovation that a microbit can be a starting point for* ***innovation.***

Different kinds of loops will be introduced and implemented.

Lesson objectives

Students will…

- Understand the value of looping (iteration) in programming
- Understand looping as a form of iteration
- Learn how and when to use the Looping blocks ‘repeat’, ‘while’, and ‘for’
- Apply the above knowledge and skills to create a unique program that uses iteration and looping as an integral part of the program
- Think of how a microbit could lead to new types of innovation

Lesson structure

- Introduction: Lather. Rinse. Repeat.
- Unplugged Activity: Walk a Square pseudocode
- micro:bit Activities: Code a Heart Beat program, code a 2 tone siren, code the song “Frere Jacques”, code a number counter
- Project: Loopy Entertainment and Innovation!
- Project Mods: Use LED lights or servo motors to add a maker element to the project
- Assessment: Rubric
- Standards: Listed

Lesson plan

1. **Overview**: Looping
2. **Unplugged**: Walk a square

3. **Activity**: Loops demos
4. **Project**: Loopy Entertainment and Innovation!

Standards — CSTA K-12 Computer Science Standards

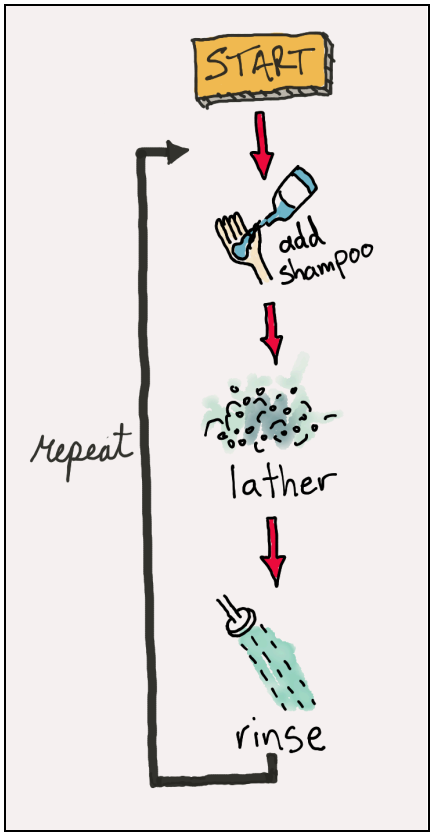
- CL.L2-05 Implement problem solutions using a programming language, including: looping behavior, conditional statements logic, expressions, variables, and functions.
- CL.L3A-03 Explain how sequence, selection, iteration, and recursion are building blocks of algorithms.

05.0 Topic Introduction

In computer programming, iteration is the repetition of a sequence of code. A loop is a form of iteration. A loop repeats code until a certain condition is met.

Questions for the students:

- Do you use shampoo to wash your hair? *Most will say 'Yes'*.
- Have you ever read the instructions on a bottle of shampoo? *Most will say 'No'*.



Most of us have never read the instructions on a bottle of shampoo, because we already know how to use shampoo.

What algorithm could you write for shampooing your hair? Example:

1. Wet hair.
2. Apply shampoo to wet hair
3. Scrub shampoo into hair
4. Rinse shampoo out of hair

If you did read the instructions on a bottle of shampoo, you may read similar instructions as the ones you just wrote with one added step at the end. That step is 'Repeat.' How does this one extra step affect the algorithm?

In computer programming, this is known as the 'shampoo algorithm' and is an example of a loop. It is also an example of an 'infinite' or 'endless' loop as the algorithm keeps repeating with no condition that ends the looping.

http://DBwebsolutions.com

'Rinse. Repeat.' has even become a meme and made its way into modern song lyrics. What other common activities involve repetitive actions? *Examples: Singing (choruses repeat), dancing, school cheers, walking and running, exercise routines…*

Optional - Lather, Rinse, Repeat

Share with your students the history of '**Lather, Rinse, Repeat**.'

Lather, Rinse, Repeat: Hygiene Tip or Marketing Ploy By Lauren Goldstein October 11, 1999 http://archive.fortune.com/magazines/fortune/fortune_archive/1999/10/11/267035/index.htm (FORTUNE Magazine) – In Benjamin Cheever's novel The Plagiarist, a marketing executive becomes an industry legend by adding one word to shampoo bottles: REPEAT. He doubles shampoo sales overnight.

This bit of fiction reflects a small yet significant eddy of U.S. consumer angst: If we REPEAT, are we or are we not playing into the hands of some marketing scheme? It turns out that in real life there's a reason you should repeat, or at least there used to be. In the 1950s, when shampoos began to be mass-marketed, we didn't wash our hair all that often–once or twice a week, as opposed to five times a week as most of us do now. Also, we used a lot more goop in our hair. It was the age of Brylcreem and antimacassars, remember. Paul Wallace, the director of hair-care research and development for Clairol, says that when cleaning agents in shampoo came up against that amount of oil and goop, "it depressed the lather." A second application was needed to get the suds that consumers expected. Lots of suds mean that hair is already clean. Maybe too clean (there's no oil to break through), but consumers like it.

FORTUNE asked Frederic Fekkai, the noted and notably expensive New York City hairdresser, what he thought about the double lather. He says, "Yesterday I put oil on my hair for a different look and went to a restaurant where the smoke was horrible. This morning I realized I had to do two shampoos."

At any rate, Wallace says advances in shampoo technology mean that only one application of, for instance, Clairol's Herbal Essences is sufficient to break through the oiliest hair. The company has stricken the use of both REPEAT and REPEAT IF DESIRED from all Clairol products. Yet a lot of brands, like Suave by Unilever and L'Oreal, still say REPEAT. Others, like Unilever's Finesse and Revlon's Flex, opt for the less imperative REPEAT IF DESIRED. Procter & Gamble uses REPEAT IF NECESSARY on Pantene.

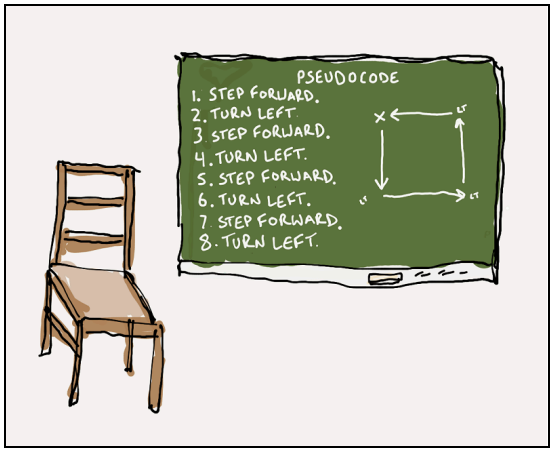
Getting consumers to wash twice can, of course, increase sales–in ways one might not imagine. Double sudsing leads to dry hair, Fekkai points out, and that means more beauty products! “When you do two shampoos, even if you don’t usually use a conditioner, you have to use a little,” he says. “The conditioner becomes very important.” REPEAT. FOLLOW WITH CONDITIONER. Words Cheever’s marketer could have retired on. –Lauren Goldstein

From Wikipedia (https://en.wikipedia.org/wiki/Lather,_rinse,_repeat): Lather, rinse, repeat (sometimes wash, rinse, repeat) is an idiom roughly quoting the instructions found on many brands of shampoo. It is also used as a humorous way of pointing out that such instructions if taken literally would result in an endless loop of repeating the same steps, at least until one runs out of shampoo. It is also a sarcastic metaphor for following instructions or procedures slavishly without critical thought.

05.1 Unplugged: Walk a square

Objective

To reinforce the concept of looping (iteration) by having students act out the repeated steps of an algorithm in real life.

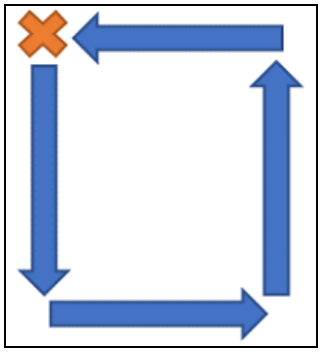


Overview

Students will give the teacher instructions to do a simple activity, then look for places where using iteration could shorten their code and make it more efficient.

Process

- Place a chair in the front of the room.
- Stand at the back right side of the chair facing the students.
- Ask the students what instructions they could give you that when followed would lead you to walk around the chair, ending up just as you started. You may want to demonstrate what this would look like by walking around the chair.
- Tell the students you can only process one instruction at a time, so their algorithm needs to be step-by-step.
- As students suggest instructions write them on the board or wherever everyone can see them. Their pseudocode will probably end up looking something like this:
    1. Step forward
    2. Turn left
    3. Step forward
    4. Turn left
    5. Step forward



6. Turn left
7. Step forward
8. Turn left

- Go ahead and follow their algorithm to prove that it works. But that's eight lines of code! Tell students that the same instructions can be written using just three lines of code. If they have not noticed already, have students look for places where the code repeats.
- Tell them that whenever you have code that repeats, you have an opportunity to use a loop to simplify your code.
- Prompts:
    - What lines are repeated? *(1) Step forward. (2) Turn left*.
    - How many times are they repeated? Four
    - So how could we rewrite this code? Students will suggest a version of the following: *Repeat 4 times: Step forward, Turn left*
- Go ahead and follow their revised algorithm to prove that it works.

There! They have just rewritten eight lines of code as three lines of code, by using a loop. The 'repeat' command creates a loop. The code within the loop gets repeated a certain number of times until a condition is met. The condition in this algorithm is that the code in the loop is repeated 4 times. Once this condition is met, the program exits the loop.

This is a great opportunity to have the students think of the benefits of having fewer lines of code. *Some possible reasons: Less typing, saves time, fewer chances of making a mistake, easier to read the code, fewer lines of code to debug…*

Notes

- Depending on the particular class, you can make this exercise more challenging, by requiring the students to be more specific in their instructions.

Example: Step forward 14 inches (you can have students actually measure the exact distance), turn left 90 degrees…

05.2 Activity: Loops demos

One of the goals of this module is to explore the connecting of external devices to the microbit and to use external sensors. This module will explore the use of speakers, external LEDs, the use of pins as buttons. As you try these activities think of other ways you might innovate by uses external connections.

Micro Python has 2 different kinds of loop:

- 'while' block
- 'for' block

While loop: In this course the 'while' loop has been used several times as a `while True:` statement to create a forever loop for events. It can also be used in the format of while condition

statement to loop only while a condition is true:

```
while temperature() < 18:
```

For loop: There are times when you want to do an action a specific number of times.The 'for' loop can be used as a counter to go through the loop a given number of times. The 'for' loop has a couple of different formats.

The first is:

```
for i in range (5):
    # action inside the loop
```

This loop will repeat upto 5 times before ending. The first time the value of 'i' is 0, then 1, 2, 3, 4 and it ends before 'i' is 5.

The second is:

```
for i in range (0, 8):
    # action inside the loop
```

This loop will repeat upto 8 times. The starting value is the first integer and it ends before the last integer.

Activity: 5.2a Heart Beat

The Heart Beat program uses a 'for' loop set to a 'heartbeat' up to 10. Instead of using a button to start the loop it use the pin0 pressed which is done by holding the ground with the right thumb and finger and then with the left hand quickly touching the 0 pin and letting it go. The touching with 2 hands makes a connection between the 0 pin and the ground. When the pin0 is connected the program will flash a big heart and a small heart as if the heart is beating. Sometimes it may take several tries to make the pin0 touch work.

Once it is downloaded to the microbit and working, have 2 students hold hands in the middle and one touch the ground and one touch the pin0. In other programs the other pins can be programmed to touch also. This gives the coder 3 additions "buttons" besides "A", "B", & "A+B".

Algorithm and Pseudocode

- *Add comments to the beginning of the program*
- *On pin0 pressed*
    - *Repeat 10 times*
    - *Show icon "Big Heart"*
    - *Show icon "Small Heart"*
- *Clear screen*

Coding Heart Beat

The coding of this activity basically follows the algorithm and pseudocode in the plan.

```
# 5.2a Heart Beat - Repeat Loop
# by C Lyman
# July 2019
# Activity from Module 5 - Music, Designs, LEDs, etc. (Loops)
# of Coding & Innovation using Microbits - Python
# This project uses a loop and pin0 to display a heart beat.

from microbit import *

display.scroll("HEART")
# set up variables and assign values of 0

# forever loop for Events
while True:
      # Event - pin0 touched?
      if  pin0.is_touched():
      # action when pin0 & ground are touched
      for i in range (10):
            display.show(Image.HEART)
            sleep(1000)
            display.show(Image.HEART_SMALL)
            sleep(1000)
      display.clear()
```

Link to Python code:

https://github.com/CarlLyman/Python-Microbits/blob/master/docs/05loops/5.2a_Repeat-Heart-Beat.py

Ideas for modifications

- Try 2 other icons
- Change the number of times the icon repeats in the ‘for’ loop

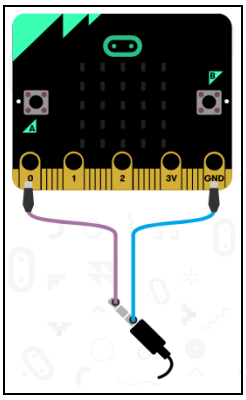
Activity: 5.2b Music Tunes

MicroPython has several tunes already pre-programmed to play music. In this activity an external speaker or headphones need to be connected. Here are instructions on more about using music in Python on a microbit. https://microbit-micropython.readthedocs.io/en/v1.0.1/tutorials/music.html

Built-in Tunes

music.DADADADUM	music.BLUES	music.CHASE
music.ENTERTAINER	music.BIRTHDAY	music.BA_DING
music.PRELUDE	music.WEDDING	music.WAWAWAWAA
music.ODE	music.FUNERAL	music.JUMP_UP
music.NYAN	music.PUNCHLINE	music.JUMP_DOWN
music.RINGTONE	music.PYTHON	music.POWER_UP
music.FUNK	music.BADDY	music.POWER_DOWN

Connecting headphones to a microbit



Here is a diagram on how to connect headphones or an external speaker to the microbit. Alligator clips can be used to make the connections. To use sound with your micro:bit, you will need to connect it to some speakers, a buzzer, or headphones. There are several connectors that can be 3D printed from http://thingiverse.com. A small “breadboard” can also be used to make the connections with connecting wires.

Coding a Pre-Built Music Tune

```
# 5.2b Music Tunes- Repeat Loop
# by C Lyman
# July 2019
# Activity from Module 5 - Music, Designs, LEDs, etc. (Loops)
# of Coding & Innovation using Microbits - Python
# This project plays 2 predefined tunes when A button is pressed.

from microbit import *
import music

display.scroll("TUNES")
while True:
      # Event - button A pressed?
      if button_a.is_pressed():
      # repeat for 2 times
      for i in range (2):
            music.play(music.PRELUDE)
            music.play(music.ODE)
```

Link to Python code:

https://github.com/CarlLyman/Python-Microbits/blob/master/docs/05loops/5.2b_Music-Tunes-Repeat.py

Ideas for modifications

- Try different tunes
- Change the number of times the tune repeats in the 'for' loop
- Add an icon to display while the tune plays

Activity: 5.2c Frere Jacques Song

Write music using loops with the song, "Are You Sleeping Brother John?" (Frere Jacques)

https://www.letsplaykidsmusic.com/wp-content/uploads/2015/02/frere-jacques-music.pdf

In the song each line repeats twice so a 'for' loop will allow the program to coded more easily.

Algorithm & Pseudocode

- *Add comments to the beginning of the program*
- *Adjust tempo as needed*
- *OnButton A pressed*
    - *Repeat 2 times*
        - *Play ¼ notes C D E C*
    - *Repeat 2 times*
        - *Play ¼ notes E F*
        - *Play ½ note G*
    - *Repeat 2 times*
        - *Play ⅛ notes G A G F*
        - *Play ¼ notes E C*
    - *Repeat 2 times*
        - *Play ¼ notes C low G*
        - *Play ½ note C*

Python code

```
# 5.2c Frere Jacques Song - Repeat Loop
# by C Lyman
# July 2019
# Activity from Module 5 - Music, Designs, LEDs, etc. (Loops)
# of Coding & Innovation using Microbits - Python
# This project uses an A button pressed to play music.
# Each line of the song repeats 2 times so each line
# can be put in a loop and repeated.

from microbit import *
```

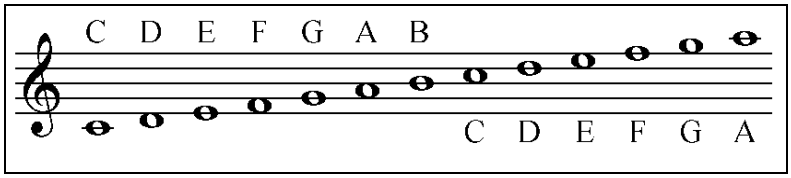
```
import music

display.scroll("SONG")
# Adjust tempo
music.set_tempo(bpm=180)

# forever loop for Events
while True:
      # Event - button A pressed?
      if  button_a.is_pressed():
      # action when button A is pressed
      #repeat for 2 times
      for i in range (2):
            music.play("C4:4")
            music.play("D4:4")
            music.play("E4:4")
            music.play("C4:4")
      #repeat for 2 times
      for i in range (2):
            music.play("E4:4")
            music.play("F4:4")
            music.play("G4:8")
      #repeat for 2 times
      for i in range (2):
            music.play("G4:2")
            music.play("A4:2")
            music.play("G4:2")
            music.play("F4:2")
            music.play("E4:4")
            music.play("C4:4")
      #repeat for 2 times
      for i in range (2):
            music.play("C4:4")
            music.play("G3:4")
            music.play("C4:8")
```

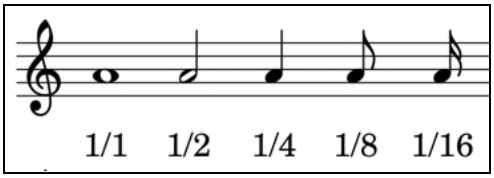
Link to Python code:

https://github.com/CarlLyman/Python-Microbits/blob/master/docs/05loops/5.2c_Frere_Jacques%20Song%20Repeat.py

Notes & Note Values

Writing music for a microbit can be written using the **"play tone 'note' for '....' beat"** block of code. When the 'note' is click on a piano key shows up and the name of the note is shown when the mouse is over the note on the piano keyboard.





These activities become good "cross-curricular" activities to help student learn to basically read music. The project at the end of this unit might be one that is coordinated with a music teacher.

"How to Read Music" https://www.musicnotes.com/blog/2014/04/11/how-to-read-sheet-music/

Modifications

- Try using the repeat loop with a "**start melody…**" block. The block has its own repeat at the end of the block.
- Find a song from sheet music and code it
- Write a song of your own

Activity: 5.2d European Siren!

A "forever" loop will run as long as the microbit is on. The "while" loop block is useful when you want your program to loop until a certain event happens or a different condition is met. When one loop is placed inside another loop it is called a "nested" loop. This program will have a nested loop. In this program a "while" loop with a certain condition is set to turn on or turn off the siren loop. A European siren is created by playing 2 notes like middle C and middle F. In this program it will be programmed to turn on when the light is sensed above a certain level. (Light levels varies between 0 'dark' and 255 'bright light'. The microbit uses the LED display as the light sensor.)

The condition in the while loop has to be a Boolean condition which can be evaluated as either "true" or "false". This is the same as the condition in an "if (condition) then …." statement.

Algorithm & Pseudocode

- *Add comments to the beginning of the program*
- *Adjust tempo as needed*
- *Check light level and store it in a variable called 'light'*
- *In a "forever" loop*
    - *Place a "if" statement with the condition (lightLevel >75)*

        - *Play whole note C*
        - *Show bright LED block*
        - *Play whole note F*
        - *Show bright LED block*
    - *Check light level again and store it in the variable called 'light'*

Coding European Siren

Use the algorithm to program the microbit. Download it and test it to see if it works.

Python Code - European Siren

```
# 5.2d European Siren While Loop
# by C Lyman
# January 2020
# Activity from Module 5 - Music, Designs, LEDs, etc. (Loops)
# of Coding & Innovation using Microbits - Python
# This project plays 2 notes based on light level

from microbit import *
import music

display.scroll("SIREN")

light =  display.read_light_level ()
while True:
      if light > 75:
      music.play("C4:4")
      display.show(Image("00000:09990:09990:09990:00000"))
      music.play("F4:4")
      display.show(Image("00000:03330:03330:03330:00000"))
      light =  display.read_light_level ()
```

Link to Python code:

https://github.com/CarlLyman/Python-Microbits/blob/master/docs/05loops/5.2d_European%20Siren.py

Note Modifications

- Try 2 other notes.
- Try a different spread between the notes.
- Change the light level needed to make the while loop work.
- Try another sensor and value to make the siren work.

- Try using a different position of the microbit to turn the siren on and off

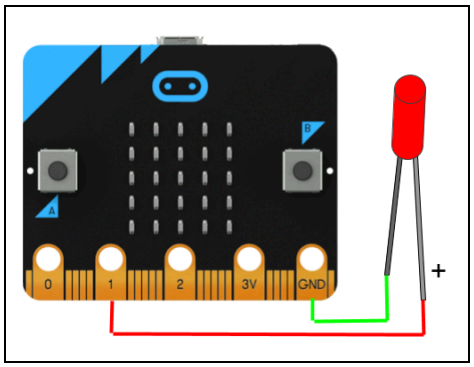
Activity: 5.2e European Siren - External LED

This modification of adding an external LED will take the European Siren and add a couple of lines of code to turn an LED on and off.

Connecting an external LED

Once the "Siren" is working add an externally connected LED light to pin1 and turn it on and off after each note is played. LEDs have negative '**-**' and a positive '**+**' sides when it is connected it needs to be connected with the '+" side to the power (pin0, pin1, or pin2) and the '-' to the ground (GND).

The LED can be powered with '`pin1.write_digital(1)`' command and turned off with the '`pin1.write_digital(0)`' command. The external device, in this case the LED, can be powered with a '**pinN.write_digital**' with a '1' turning on the power or a '0' turning off the power. A external device can also be powered with '**pinN.write_analog(*number*)'** with a value between 0 and 1023. The higher the number the greater the amount of power applied to the external device. This allows the lighting of the LED from a very dim to a bright lighted one.



Just as an LED can be lighted, other external devices can be powered with the microbit with up to 3 volts from the power supply to the microbit.

Connecting an external sensor

Just as external LED can be powered so can values from external sensors be read using the following commands:

```
variable = pinN.read_digital()
```

This will return a value of either a 0 or a 1.

```
variable = pinN.read_analog()
```

This will return a value between 0-1023.

External LED Python Code

Connect an external LED to the microbit at pin1 and the ground with the long wire on the LED to the pin1 and the short wire to the ground. The music is playing to the speaker using pin0 so for this

project the external LED is connected to pin1. The value of 1 will light the LED and the value of 0 will turn off the LED. (The LED and speaker can be connected to different pins.)

Algorithm & Pseudocode

- *Add comments to the beginning of the program*
- *Adjust tempo as needed*
- *Check light level and store it in a variable called 'light'*
- *In a "forever" loop*
    - *Place a "if" statement with the condition (lightLevel >75)*
        - *Play whole note C*
        - *Show bright LED block*
        - *Turn on external LED on pin1*
        - *Play whole note F*
        - *Show bright LED block*
        - *Turn off exteranal LED on pin1*
    - *Check light level again and store it in the variable called 'light'*

Use the algorithm to program the microbit. Connect the speakers and external LED then download it and test it to see if it works.

Python Code - European Siren - x LED

```
# 5.2e European Siren While Loop
# by C Lyman
# January 2020
# Activity from Module 5 - Music, Designs, LEDs, etc. (Loops)
# of Coding & Innovation using Microbits - Python
# This project plays 2 notes based on light level

from microbit import *
import music

display.scroll("SIREN")

light =  display.read_light_level ()
while True:
     if light > 75:
          music.play("C4:4")
          display.show(Image("00000:09990:09990:09990:00000"))
          pin1.write_digital(1)
          music.play("F4:4")
          display.show(Image("00000:03330:03330:03330:00000"))
```

```
        pin1.write_digital(0)
light =  display.read_light_level ()
```

Link to Python code:

https://github.com/CarlLyman/Python-Microbits/blob/master/docs/05loops/5.2e_European%20Siren%20xLED.py

Modifications

- Try an “analog write pin (P1) to (value) to light the LED from a dim value at 0 or a bright value at 1023
- Try a different colored LED
- Add a second LED to pin2 that is different color and turn one on then off when the second LED is lit
- Add an external sensor and read and display its value

Activity: 5.2f Counting Numbers - for Loop

The “**for**” loop lets a programming count from a beginning number until an end number reached. The “**for**” loop can also tell what to count by. With the normal count being by 1s. The loop keeps track of what number it is currently on with the variable “index”. The value of “index” increases by 1 each time it loops in a normal loop. Example: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, … last number. It starts with the key word “**for”** the variable **“index”** from the starting number **“0**” to the end number “**4”**. It counts by 1s as a default..

```
for i in range (0, 4):
      # action inside the loop
```

Historical use of ‘I’ and index

In the programming language, FORTRAN, from the 1950’s “for” loops could only have a single letter variable “I”, “J”, or “K”. Even today many programmers will use a single letter “i” as the variable name for the index of the loop. Programmers also start counting with “0” instead of “1” in most cases. (https://skillcrush.com/2013/01/17/why-programmers-start-counting-at-zero/) It is all about efficiency. If you look at a child’s age, the first year is the zero year. Only after 1 year is a child’s age counted as 1.

Structure of a for ... in range loop

In JavaScript the starting, ending number, and count by number can be changed. The ‘i’ variable is the “index” can also be changed if it needs to be.

```
for index in range (start number, end number, countby number):
      # action inside the loop
```

In the example above the first “index” = 0 is the starting number. The “;” separates the starting number and the phrase the marks the ending number. The ‘ending’ phrase says, “keep doing the loop while index is less than or equal to 20 or whatever the end number is.” The last part of the statement says, “index++”, this means to count by 1 each time it loops. Other ways to write this part of the statement could include:

Counting forward examples:

```
for i in range (5):
     # i will go from 0 to 4

for i in range (4, 10):
     # i will go from 4 to 9

for i in range (2, 20, 2):
     # i will go from 2 to 18 by 2s
```

Counting backwards

```
for i in range (10, 0, -1):
     # i will go from 10 to 1

for i in range (100, 0, -7):
     # i will go from 100 to 1 by 7s
```

Counting Numbers with ‘for’ loop

In this project the microbit will set up to count numbers using a ‘for’ loop when the onButton A is pressed. It will start by counting from 0 to 10 then the program can be modified to count to another ending number.

Algorithm & Pseudocode

- *Add comments to the beginning of the program*
- *When the an onButton “A” is pressed start counting*
- *In a ‘for’ loop set the end number to 10*
    - *Display current number “i” inside the loop*
- *In a ‘for’ loop set range 0 to 10 and count by 2s*
    - *Display current number “i” inside the loop*
- *In a ‘for’ loop set the start number to 10 the end number to 0 and count backwards by 1s*
    - *Display current number “i” inside the loop*
- *Display word “FIRE”*

Coding Counting Loops

Use the algorithm to program the microbit. Download and test it on the microbit..

Python Code - Counters

```
# 5.2f Counter
# by C Lyman
# January 2020
# Activity from Module 5 - Music, Designs, LEDs, etc. (Loops)
# of Coding & Innovation using Microbits - Python
# This project displays the current value of 'i' counted on the screen

from microbit import *

#Count by 1s
display.scroll("COUNT1s")
for i in range (10):
      display.show(i)
      sleep (500)

# Count by 2s
display.scroll("COUNT2s")
for i in range (0, 10, 2):
      display.show(i)
      sleep (500)

# Count Down by 1s
display.scroll("CNTDWN")
for i in range (10, 0, -1):
      display.show(i)
      sleep (500)
display.scroll("FIRE")
```

Link to Python code:

https://github.com/CarlLyman/Python-Microbits/blob/master/docs/05loops/5.2f_Counter.py

Modifications

- Make a second loop that counts backwards
- Create a Music Note player loop. Each note has a value on the microbit. Here is a list of the notes by value. Set up a loop that starts with the index at 262 goes to 554. http://bit.ly/microbitnotevalues

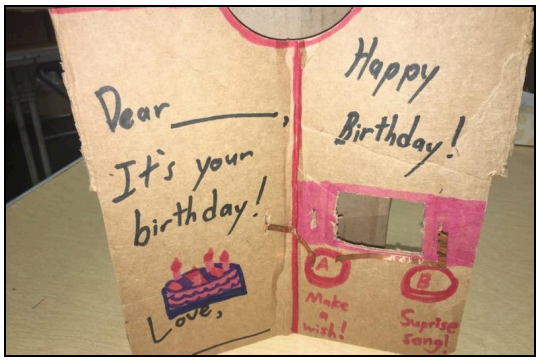
- Create an **American Siren** by playing notes going up and then down and repeating it in a for loop. Also adjust the loop so it counts by 10s up and then down by 10s.
- Connect and external LED and use the “analog write pin (P1) to index” inside a ‘for’ loop from 0 to 1023 so that the LED starts dim and grows brighter. In JavaScript have it count backwards from 1023 to 0 so the LED goes from bright to dim.

05.3 Innovation Project: Loopy Entertainment and Innovation!

There are many different ways to use the three types of loop blocks. By also using the external pins as input sensors and outputs to LEDs or speaker a student can start thinking creatively like and innovator! Enhanced modifications for some of the sample programs could be used

Recall the different common repetitive actions you thought of back at the beginning of this lesson.

- How will you use loops to create something useful, entertaining, or interesting?
- What might you make?



Project Ideas, Design, & Plan

Here are some suggestions:

- Create an animated gif (looping image that changes) and add music that matches.
- Create animation that repeats for one of the melodies included in Make Code (like Happy Birthday).
- Create an electronic greeting, birthday, get well, or other kind of card with music and lights.
- Create different animations that run when different buttons are pressed.
- Make a jukebox player that can play different tunes.
- Create an alarm that includes sound and images. What will set the alarm off? What will make the alarm stop sounding?
- Create a holiday display with music, LED animation and external LEDS.
- Create a piece of “wearable technology” with lights and music.
- Use servo motors to create a creature that dances and changes its expression while a song plays.
- Create a water alarm when 2 wires come in contact with water and then the microbit will play a warning noise and flashing alarm.
- Create a burglar alarm that will play a warning noise and flashing alarm when the microbit is moved or 2 wires come in contact with each other.

Example:

Hat Man Project

Hat Man Videos
micro:bit Hat Man
https://youtu.be/Xvybu_T5IL8

micro:bit Hat Man - inside view
https://youtu.be/ZfKgFQjygQQ

This project uses the micro:bit light sensor to display a happy face when it is sunny, and a frowning face when it is dark. The micro:bit is connected to a servo mounted on the inside of the container, and the smile and frown are attached to plastic coffee stirrers with tape and hot glue.

Juke Box

Create a music juke box that will play different tunes using different events for each song.

Loop Modifications

Any of the loop activities could be enhanced using some of the ideas in the modifications.

Reflection

Have students write a reflection of about 150–300 words, addressing the following points:

- Explain how you decided on your particular “loopy” idea. What brainstorming ideas did you come up with?
- What type of loop did you use? For, While, or Repeat
- What was something that was surprising to you about the process of creating this program?
- Describe a difficult point in the process of designing this program, and explain how you resolved it.
- What feedback did your beta testers give you? How did that help you improve your loop demo?

Handout to help writing your reflection. http://bit.ly/2tVsCmG or “Coding Microbits using Python” booklet http://bit.ly/MicrobitPythonReflections

Assessment - Competency scores

Competency	4	3	2	1
Loops	At least 3 different loops are implemented in a meaningful way.	At least 2 loops are implemented in a meaningful way.	At least 1 loop is implemented in a meaningful way.	No variables are implemented.
Variables (parameters)	All variable names are unique and clearly describe what information values the variables hold.	The majority of variable names are unique and clearly describe what information values the variables hold.	Few variable names are unique or clearly describe what information values the variables hold.	None of the variable names clearly describe what information values the variables hold.
Sound, Display, & Motion	Uses sound, display, and motion in a way that is integral to the program.	Uses only two of the required elements in a way that is integral to the program.	Uses only one of the required elements in a way that is integral to the program.	None of the required elements are used.
Micro:bit Program	micro:bit program: 1) Uses loops in a way that is integral to the program 2) Compiles and runs as intended 3) Meaningful comments in code	micro:bit program lacks 1 of the required elements.	micro:bit program lacks 2 of the required elements.	micro:bit program lacks 3 or more of the required elements.
Collaboration Reflection	Reflection piece includes: 1) Brainstorming ideas 2) Construction 3) Programming 4) Beta testing	Reflection piece lacks 1 of the required elements.	Reflection piece lacks 2 of the required elements.	Reflection piece lacks 3 of the required elements.

Credits:

This project heavily utilizes the open source work done by Douglas & Mary Kiang who wrote the course “Intro to CS using Microbits”. It is published by Microsoft MakeCode team and Microbit.org. https://makecode.microbit.org/courses/csintro