# Loopy LEDs

Professor Bob Brown

College of Computing and Software Engineering
Kennesaw State University

Bob.Brown@Kennesaw.edu

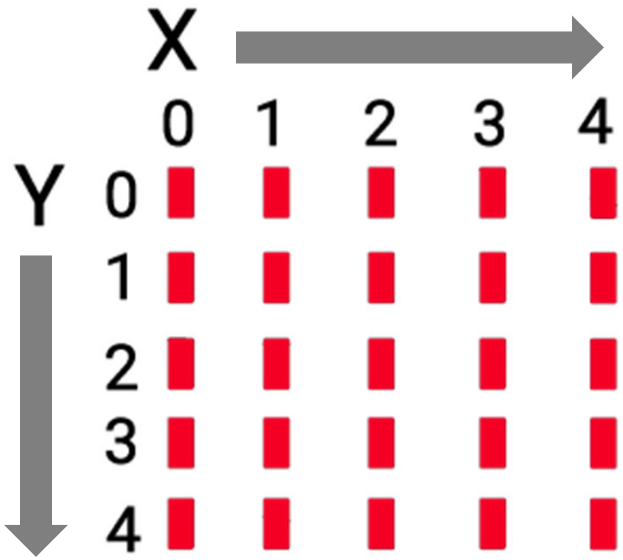
KENNESAW STATE UNIVERSITY

# Controlling the LEDs

- So far, we've let MakerCode manage the LEDs for us.
- The *leds* block lets us display any picture.
- There are special instructions to show icons, strings, and numbers, and to clear the screen.
- We can use the *plot* block to control individual LEDs.

# The Micro:Bit LEDs

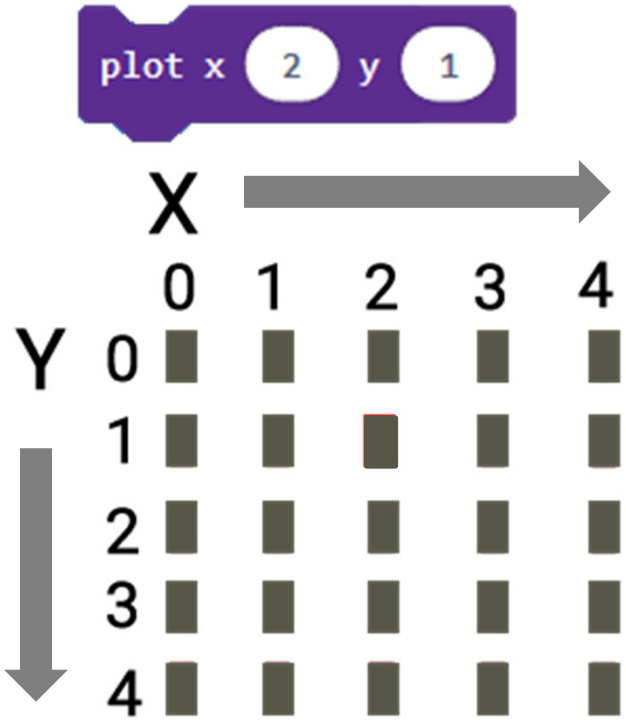
- The LEDs are organized as a grid.
- X goes left to right, Y goes top to bottom.
- Numbering starts with *zero.*




KENNESAW STATE UNIVERSITY

# Turn On A Specific LED

- The *plot* block turns on a specific LED.
- The *unplot* block turns it off.

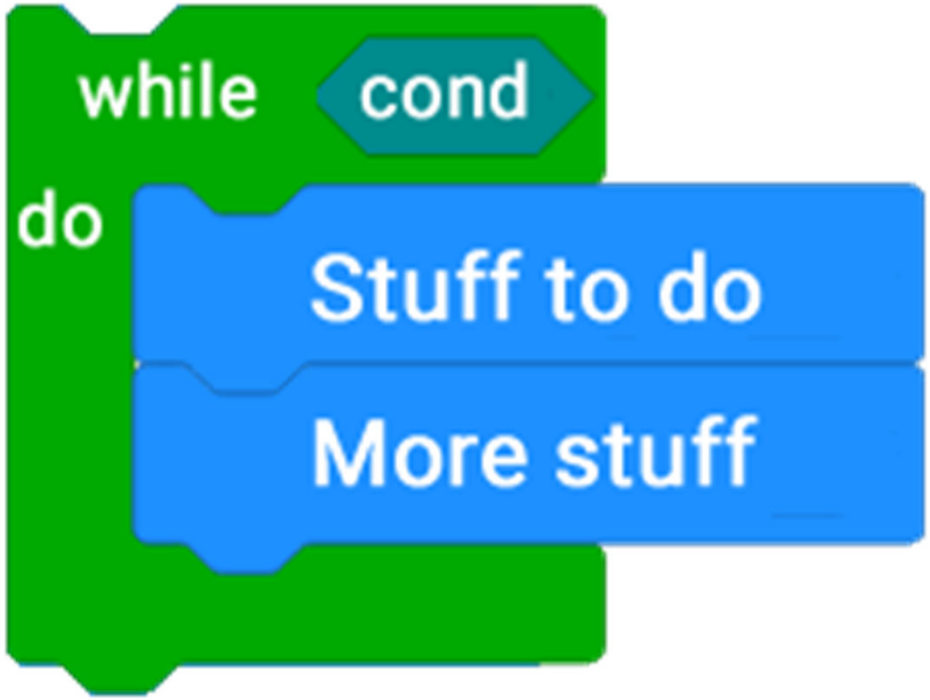



KENNESAW STATE UNIVERSITY

# Introducing the Loop

- In computer programming, a *loop* repeats the same action, possibly with different data.
- Looping is also called *iteration*.

# Looping with Micro:Bit Blocks




**Loop condition (when to stop)**

**Body of loop**

**Some loops have other features, like automatic counters.**

KENNESAW STATE UNIVERSITY

# Light those LEDs

- If we wanted to make each one of the 25 LEDs light up one after another,
  - we'd need 25 *plot* and 25 *unplot* blocks
  - and it would only happen once.
- Let's do it with loops instead.

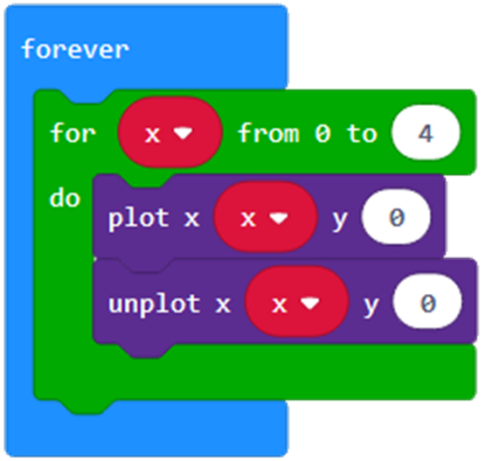
# Algorithm

- Set x to 0
- Turn on the LED at x,0
- Turn it off
- Add 1 to x
- If x > 4, stop.

We'll make this run forever by putting it inside a *forever* block.

# At First, Top Row Only

- Make a variable, x.
- Use a *for loop* to go from zero to 4.
- Turn on with *plot*, turn off with *unplot*.

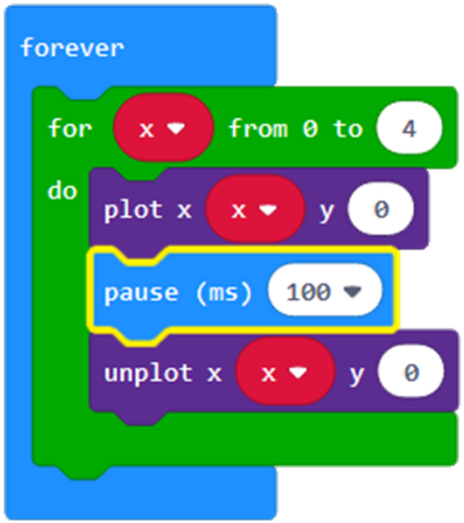


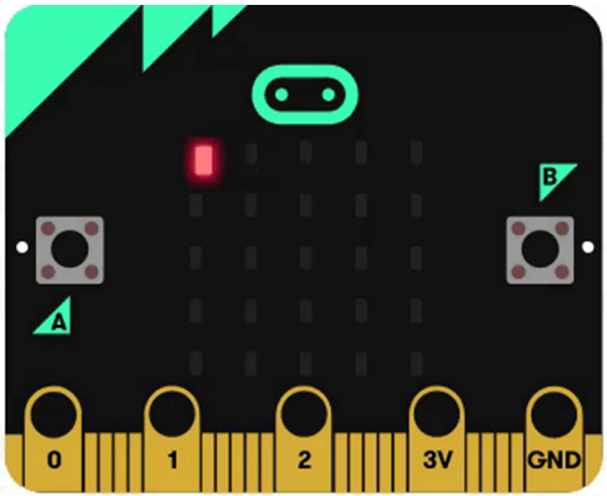

- Nothing seems to happen.  Why not?

KENNESAW STATE UNIVERSITY

# Faster than the Eye Can See

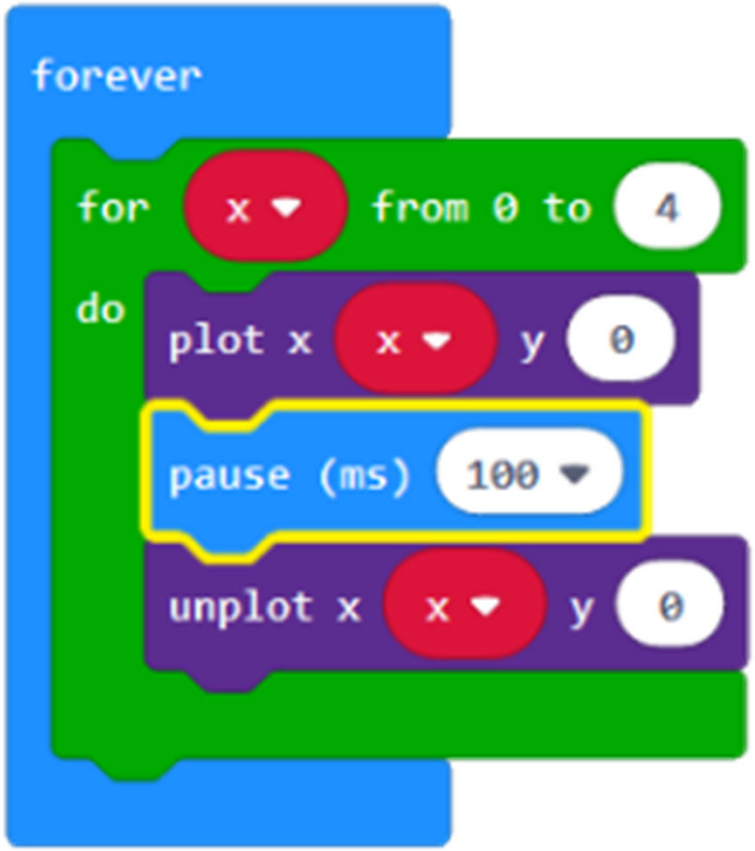
- My program was turning the LED on…
- And *immediately* turning it off.
- That happened so fast we couldn't see it. Insert a brief pause.




KENNESAW STATE UNIVERSITY

# Try It and See




- Using a *for loop* automagically creates an *index* variable.
- It is meaningful to call our variable *x.*
- You can delete *index*, or just ignore it.

KENNESAW STATE UNIVERSITY

# Thought Challenge

- The LED moves left-to-right; can you think of a way to make it move right-to-left?
- *Hint:*
  - Do not change *x* within the loop!
  - You will need a new, temporary variable.
  - You can do arithmetic on it, such as setting it to *x* – something.
  - There's an "absolute" operator in the Math bin.

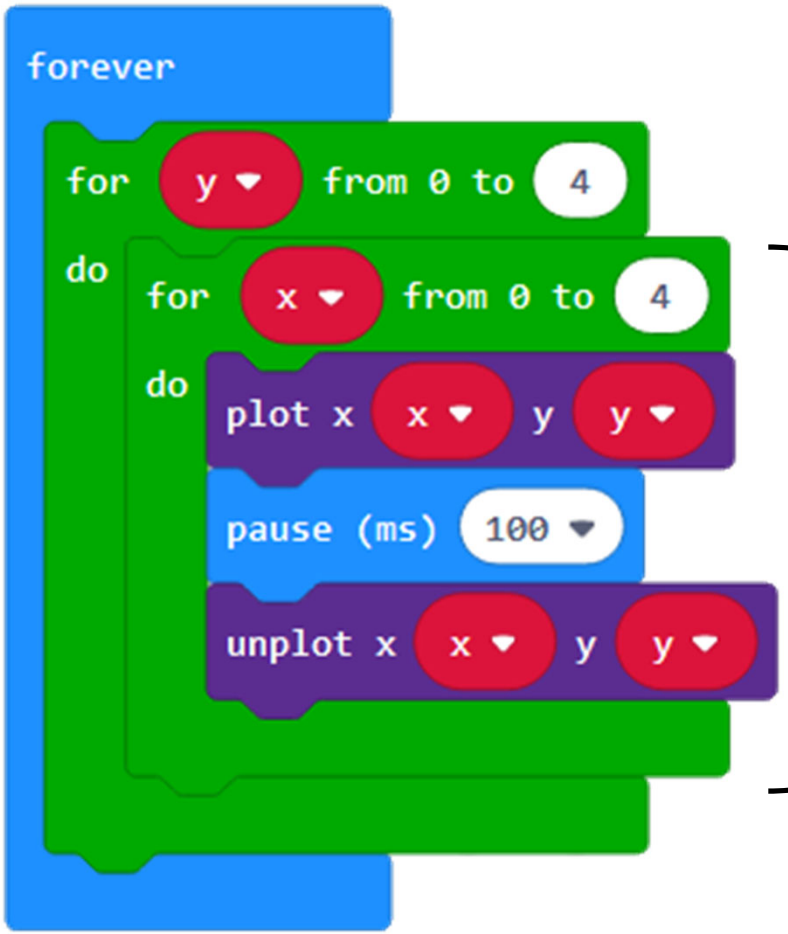
# Go Through *All* the LEDs

Algorithm

- Set *y* to 0
  - Go through the *x* loop, 0 to 4
- Add 1 to *y*
- If *y* > 4, stop.

- Changing *y* will be the *outer loop*
- Changing *x* will be the *inner loop*
- The complete inner loop runs once for each step of the outer loop.

# Go Through *All* the LEDs




The *inner loop* runs all four times for *each* step of the outer loop.

KENNESAW STATE UNIVERSITY

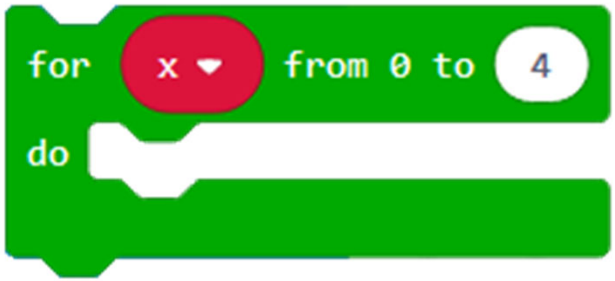
# Thought Challenge

- The program on the previous slide goes across, row-wise.
- Could you make it go **down** the first column, then the next, and so on? How?

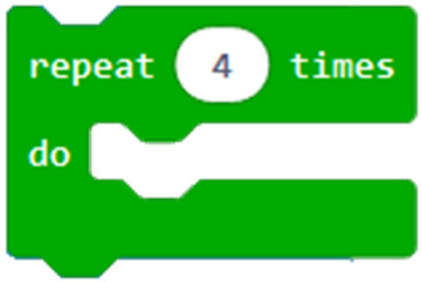
# Stop that Loop!

- The loops in our program count 0, 1, 2, 3, 4 and stop.
- They repeat only because they're inside a *forever* block.
- We used a *for loop*.
- There are other kinds of loops for different purposes.

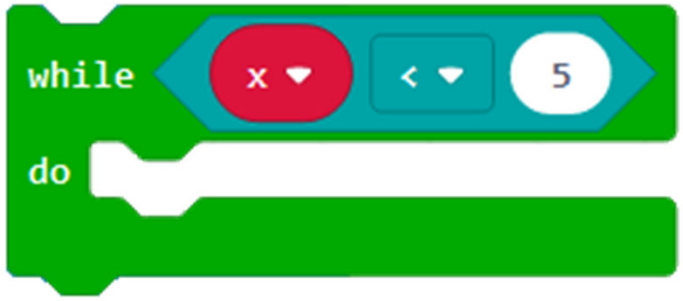
# Kinds of Loops




We used a **for loop** in our LED program. Variable *x* changes each time through the loop, and can be used inside the loop.




The **repeat loop** just repeats a predetermined number of times. Nothing within the loop is changed automagically.




The **while loop** runs as long as the conditional expression is true. To stop the loop, something within the loop must make the expression false.

KENNESAW STATE UNIVERSITY

# What Have We Learned Today?

- Looping, also called *iteration*, repeats the same actions, possibly with different values.
- There is more than one kind of loop.
- Looping is one of the important methods of program control.
- The other two are *sequence*, one thing at a time, and *selection*, choosing a path, possibly with an *if block*.

# What Else?

- The LEDs are arranged in a grid with X and Y axes.
- We can use *nested loops* to cover the whole grid.
- The grid can be covered rows first or columns first.

# Loopy LEDs

Professor Bob Brown

College of Computing and Software Engineering
Kennesaw State University

Bob.Brown@Kennesaw.edu

KENNESAW STATE UNIVERSITY