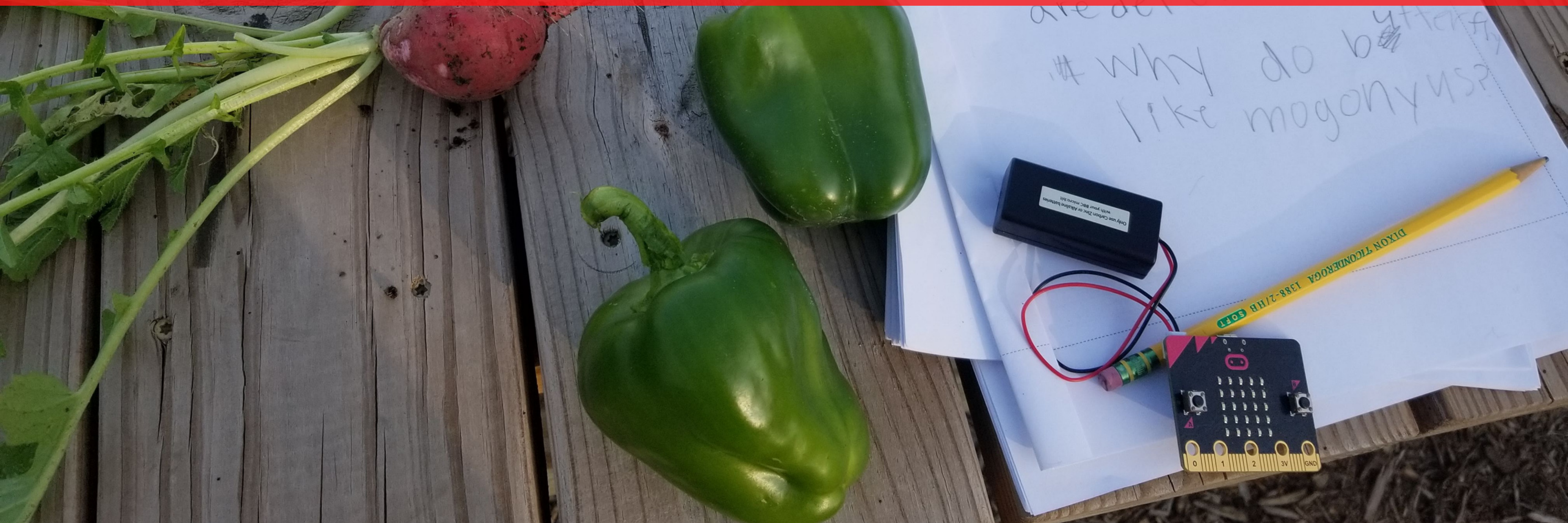
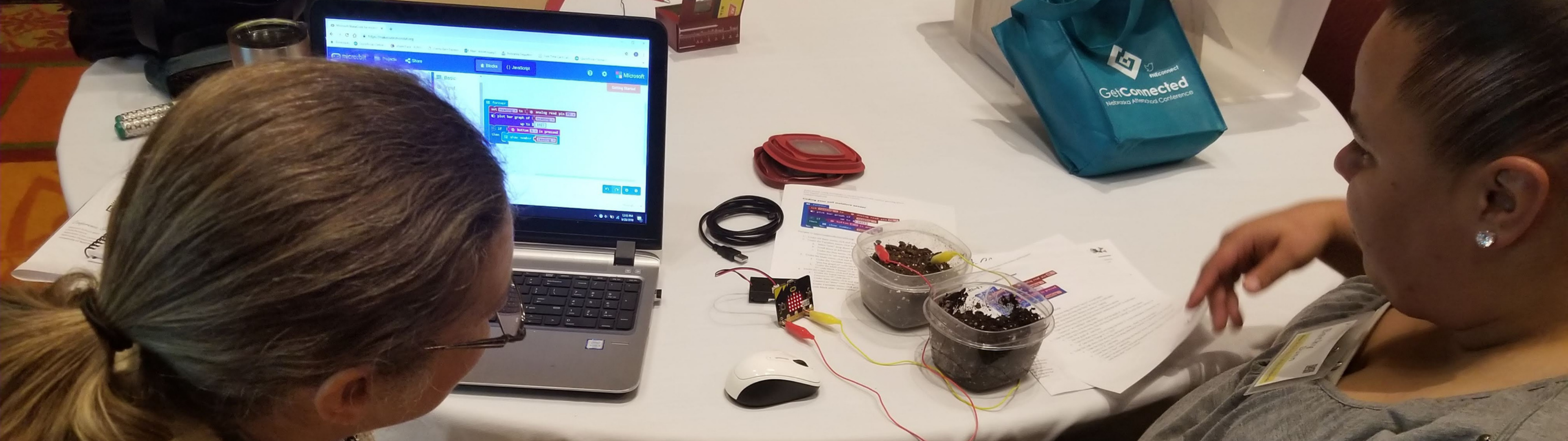


# Garden TOOLS

## PROFESSIONAL DEVELOPMENT SERIES







# Coding environmental sensors

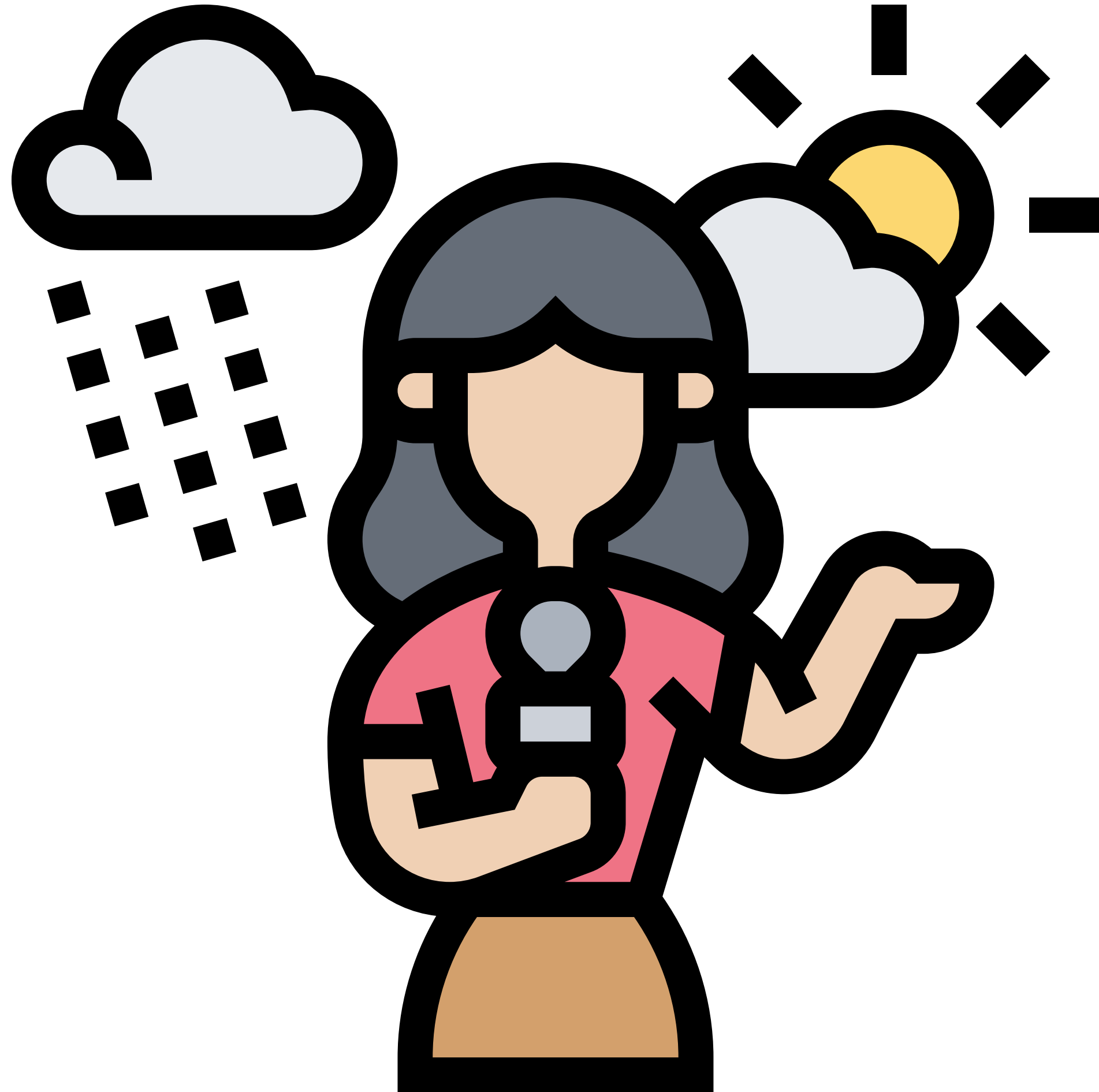


# Notice and Wonder

One of our goals is to support exploration and curiosity.

Coding and using sensors is one way to do this in a garden setting.





**I want you  
to think  
about the  
weather...**

Without using any tools or technologies, how would you describe the weather outside today?



**Imagine I give you a BBC micro:bit that measures one of the following:**



**LIGHT LEVEL**



**TEMPERATURE**



**SOIL MOISTURE**





**Go outside  
and explore**



# Take measurements for 5 minutes to answer the following questions:

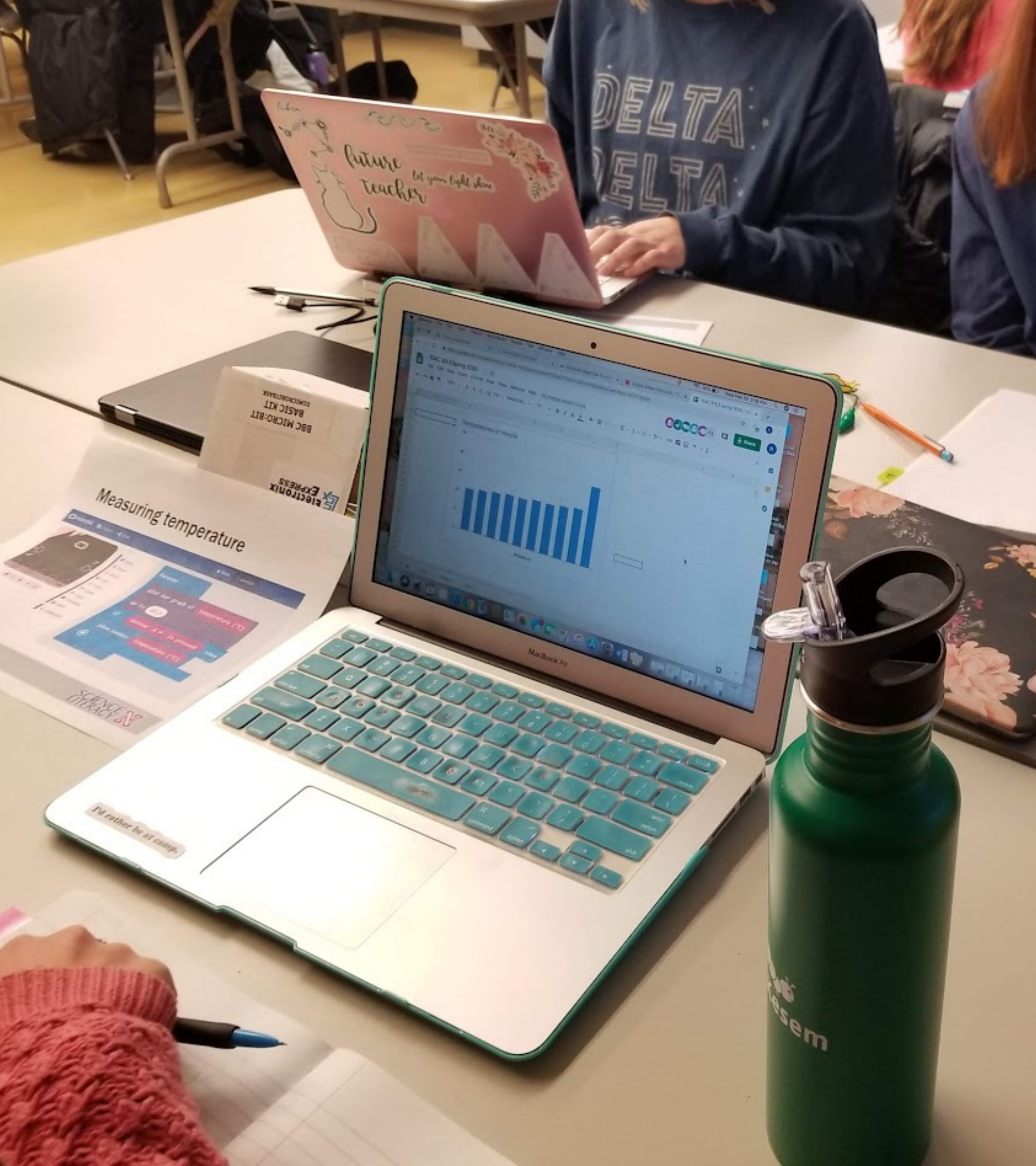
**1**

Are conditions the same everywhere?

**2**

What are the most extreme conditions you can find?





# Share and Compare

Imagine comparing your measurements with others.

How would YOU visualize the data?  
How would visualizing the data help you to look for patterns?



# Now that you know why, let's talk about how...

I am going to walk you through the basics of coding each of the three environmental sensors.

The code looks and functions very much the same for each.



TEMPERATURE



LIGHT LEVEL



SOIL MOISTURE





LIGHT LEVEL



TEMPERATURE



SOIL MOISTURE

```
forever
  plot bar graph of light level
  up to 255
  if button A is pressed then
    show number light level
```

```
forever
  plot bar graph of temperature (°C)
  up to 37.7
  if button A is pressed then
    show number temperature (°C)
```

```
forever
  set reading to analog read pin P0
  plot bar graph of reading
  up to 1023
  if button A is pressed then
    show number reading
```

**What similarities do you notice?**



**In all cases, the code functions to do two things:**

**1**

**SHOW A GRAPH**

**2**

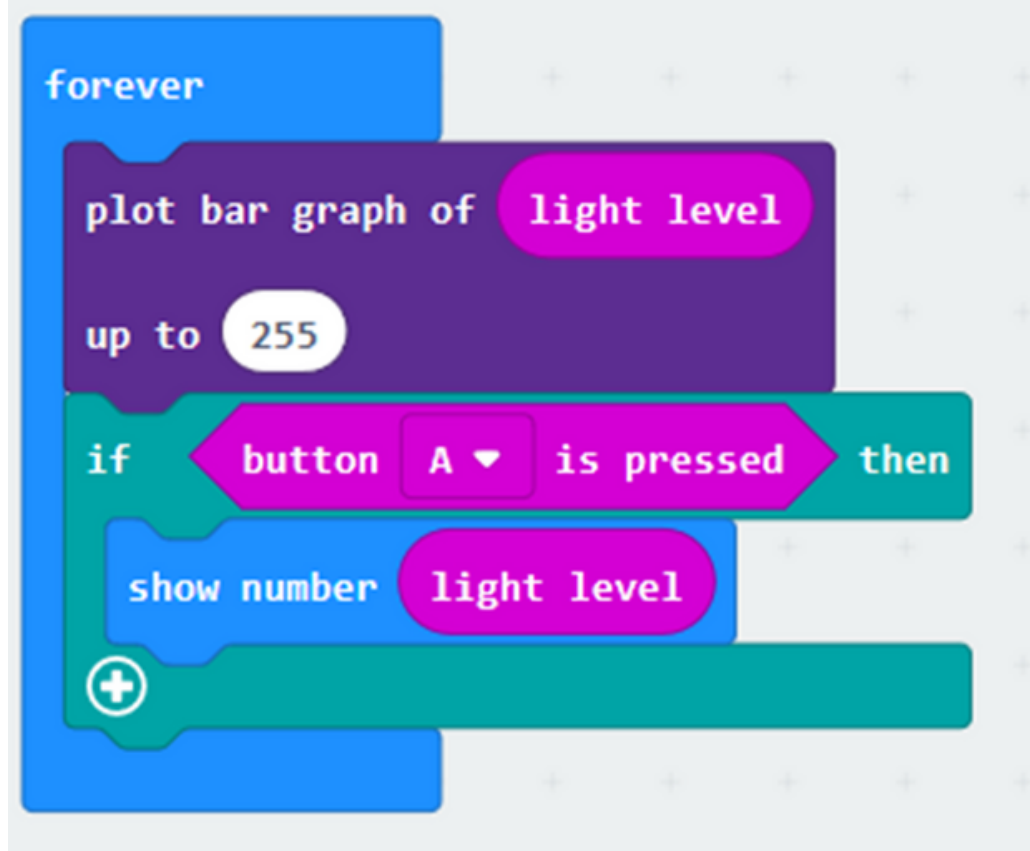
**SHOW A NUMBER  
WHEN THE "A"  
BUTTON IS PRESSED**



# In all cases, the code functions to do two things:

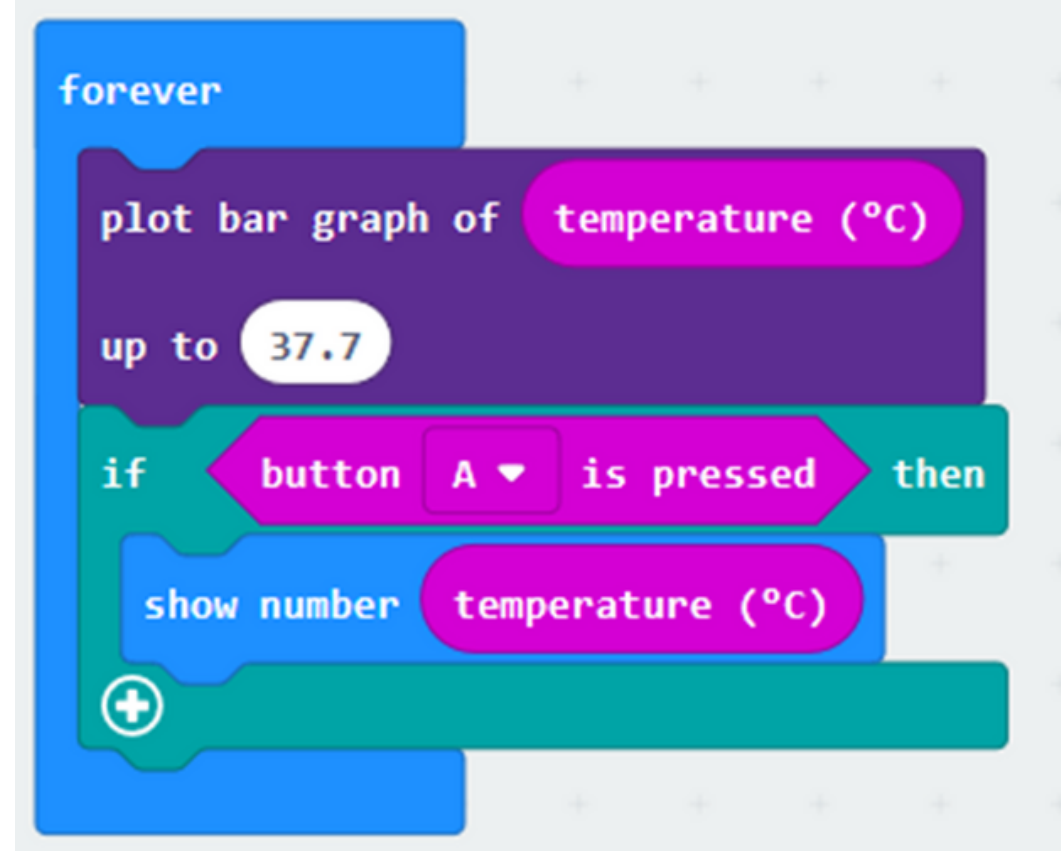
1

SHOW A GRAPH



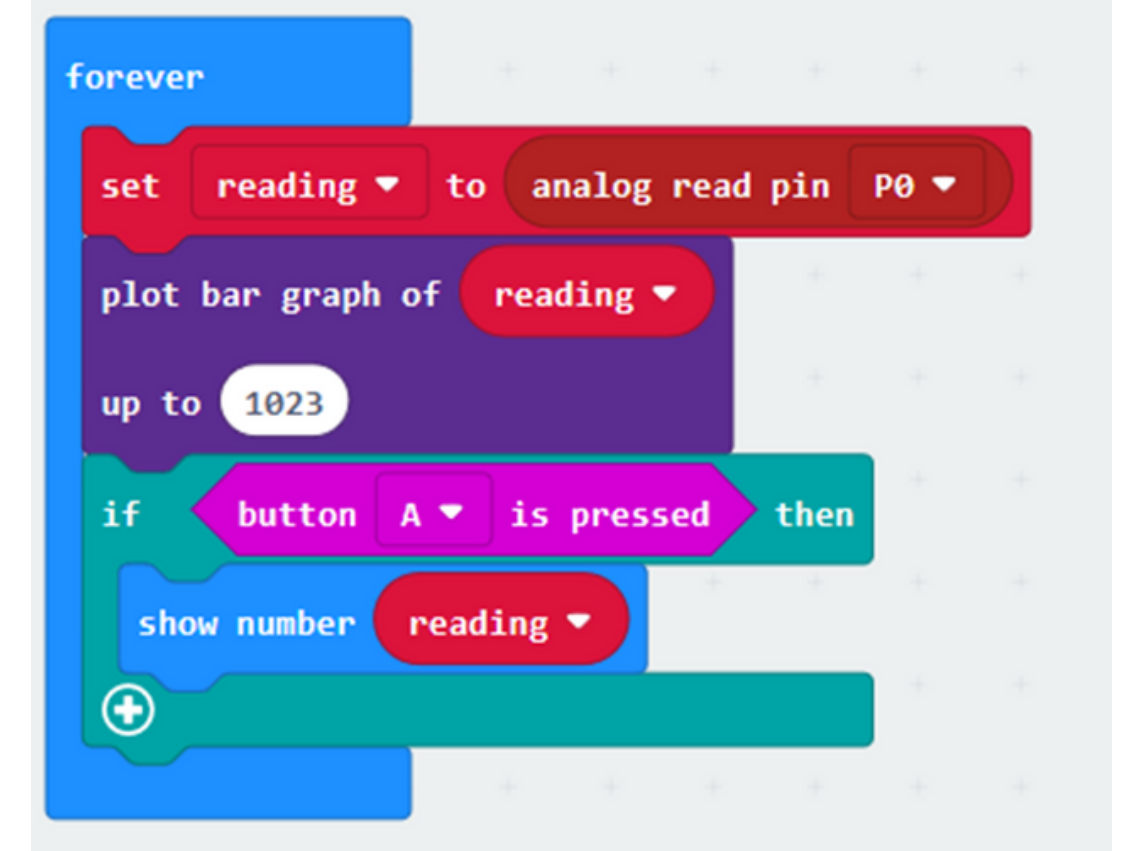
```
forever
  plot bar graph of light level
  up to 255
  if button A is pressed then
    show number light level
```

The code block is a blue 'forever' loop containing a purple 'plot bar graph of light level' block with 'up to 255' in a white circle. Below it is a green 'if button A is pressed then' block with a blue 'show number light level' block inside.



```
forever
  plot bar graph of temperature (°C)
  up to 37.7
  if button A is pressed then
    show number temperature (°C)
```

The code block is a blue 'forever' loop containing a purple 'plot bar graph of temperature (°C)' block with 'up to 37.7' in a white circle. Below it is a green 'if button A is pressed then' block with a blue 'show number temperature (°C)' block inside.



```
forever
  set reading to analog read pin P0
  plot bar graph of reading
  up to 1023
  if button A is pressed then
    show number reading
```

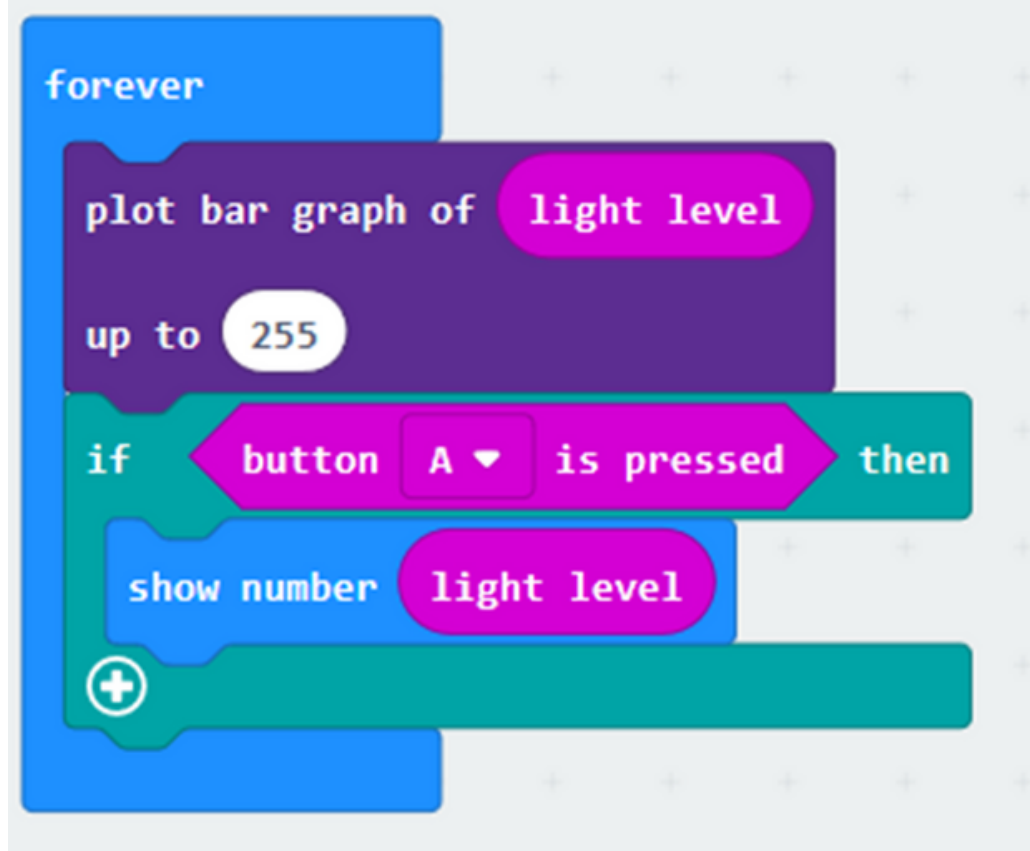
The code block is a blue 'forever' loop containing a red 'set reading to analog read pin P0' block, a purple 'plot bar graph of reading' block with 'up to 1023' in a white circle, and a green 'if button A is pressed then' block with a blue 'show number reading' block inside.



# In all cases, the code functions to do two things:

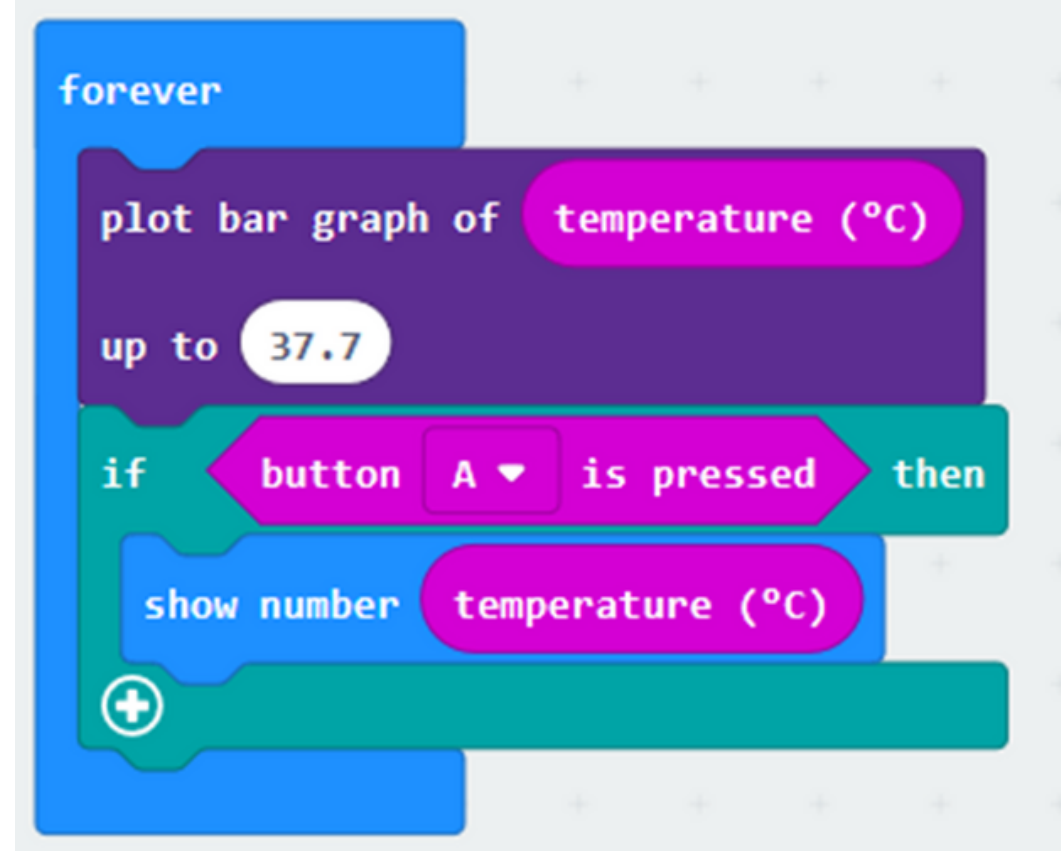
## 2

SHOW A NUMBER  
WHEN THE "A"  
BUTTON IS PRESSED



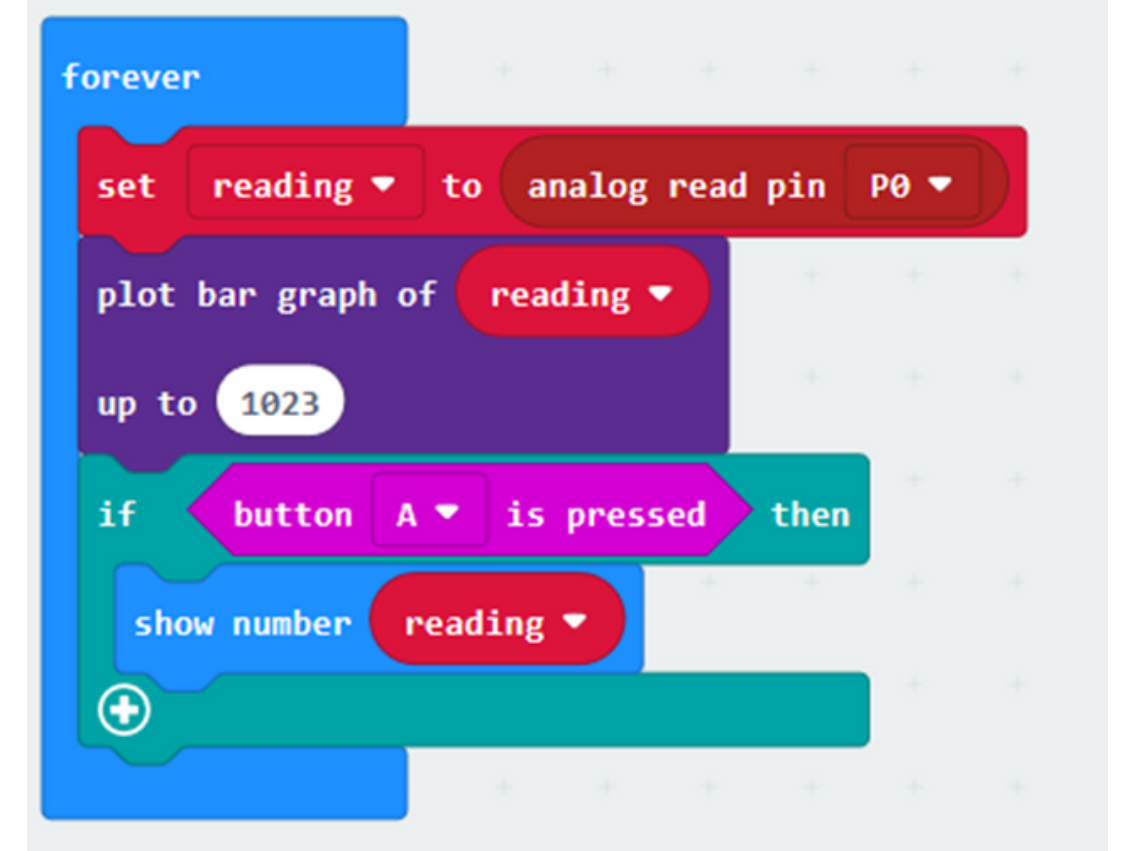
```
forever
  plot bar graph of light level
  up to 255
  if button A is pressed then
    show number light level
```

The code block is a blue 'forever' loop containing a purple 'plot bar graph of light level' block with 'up to 255' and a green 'if button A is pressed then' block containing a blue 'show number light level' block.



```
forever
  plot bar graph of temperature (°C)
  up to 37.7
  if button A is pressed then
    show number temperature (°C)
```

The code block is a blue 'forever' loop containing a purple 'plot bar graph of temperature (°C)' block with 'up to 37.7' and a green 'if button A is pressed then' block containing a blue 'show number temperature (°C)' block.



```
forever
  set reading to analog read pin P0
  plot bar graph of reading
  up to 1023
  if button A is pressed then
    show number reading
```

The code block is a blue 'forever' loop containing a red 'set reading to analog read pin P0' block, a purple 'plot bar graph of reading' block with 'up to 1023', and a green 'if button A is pressed then' block containing a blue 'show number reading' block.



# In all cases, the code functions to do two things:

1

SHOW A GRAPH

2

SHOW A NUMBER  
WHEN THE "A"  
BUTTON IS PRESSED

```
forever
  plot bar graph of light level
  up to 255
  if button A is pressed then
    show number light level
```

```
forever
  plot bar graph of temperature (°C)
  up to 37.7
  if button A is pressed then
    show number temperature (°C)
```

```
forever
  set reading to analog read pin P0
  plot bar graph of reading
  up to 1023
  if button A is pressed then
    show number reading
```



```
forever
  plot bar graph of light level
  up to 255
  if button A is pressed then
    show number light level
```

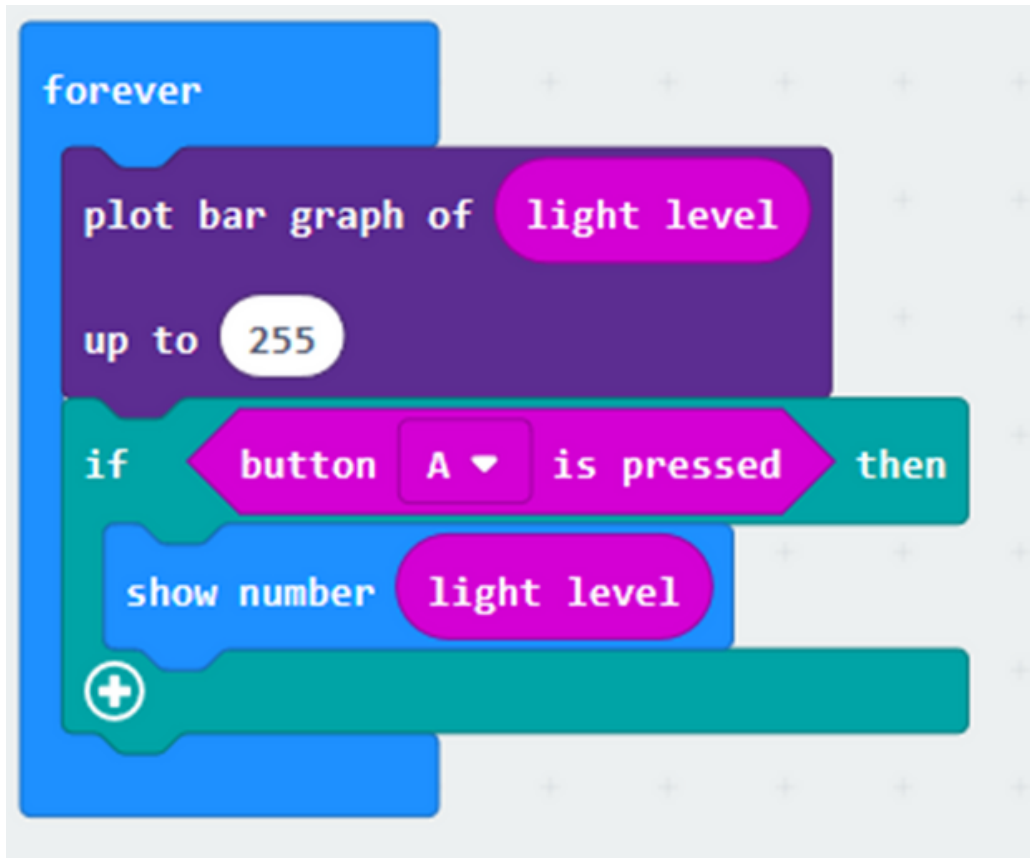
```
forever
  plot bar graph of temperature (°C)
  up to 37.7
  if button A is pressed then
    show number temperature (°C)
```

```
forever
  set reading to analog read pin P0
  plot bar graph of reading
  up to 1023
  if button A is pressed then
    show number reading
```

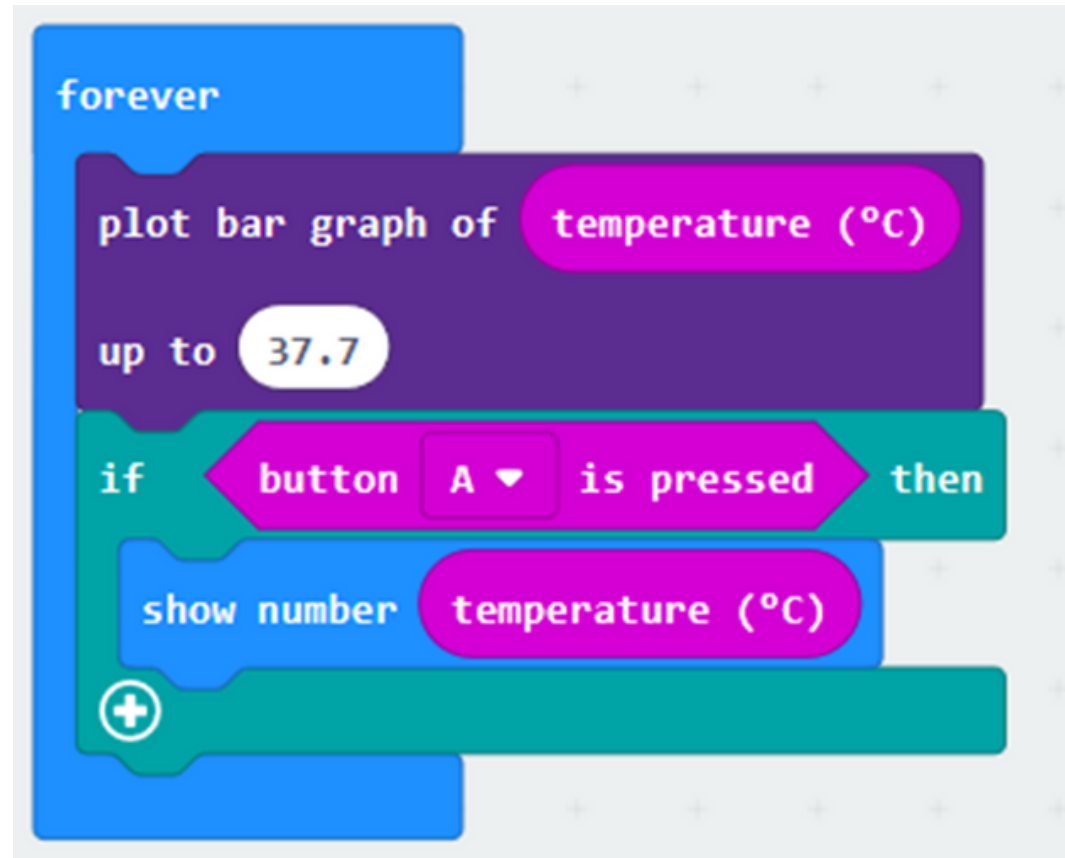
What is measured and how it is communicated depends on the details you include in the program.



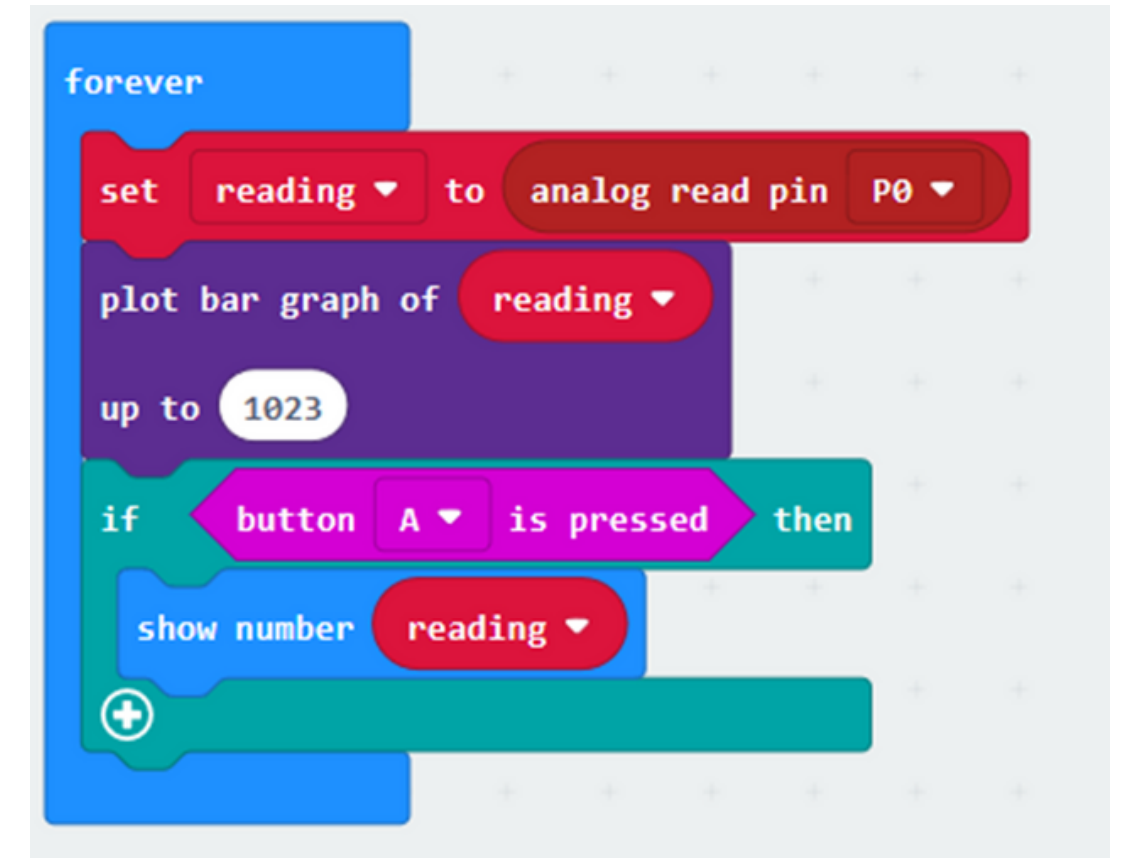
# A great example of abstraction!



```
forever loop:  
  plot bar graph of light level  
  up to 255  
  if button A is pressed then  
    show number light level
```



```
forever loop:  
  plot bar graph of temperature (°C)  
  up to 37.7  
  if button A is pressed then  
    show number temperature (°C)
```



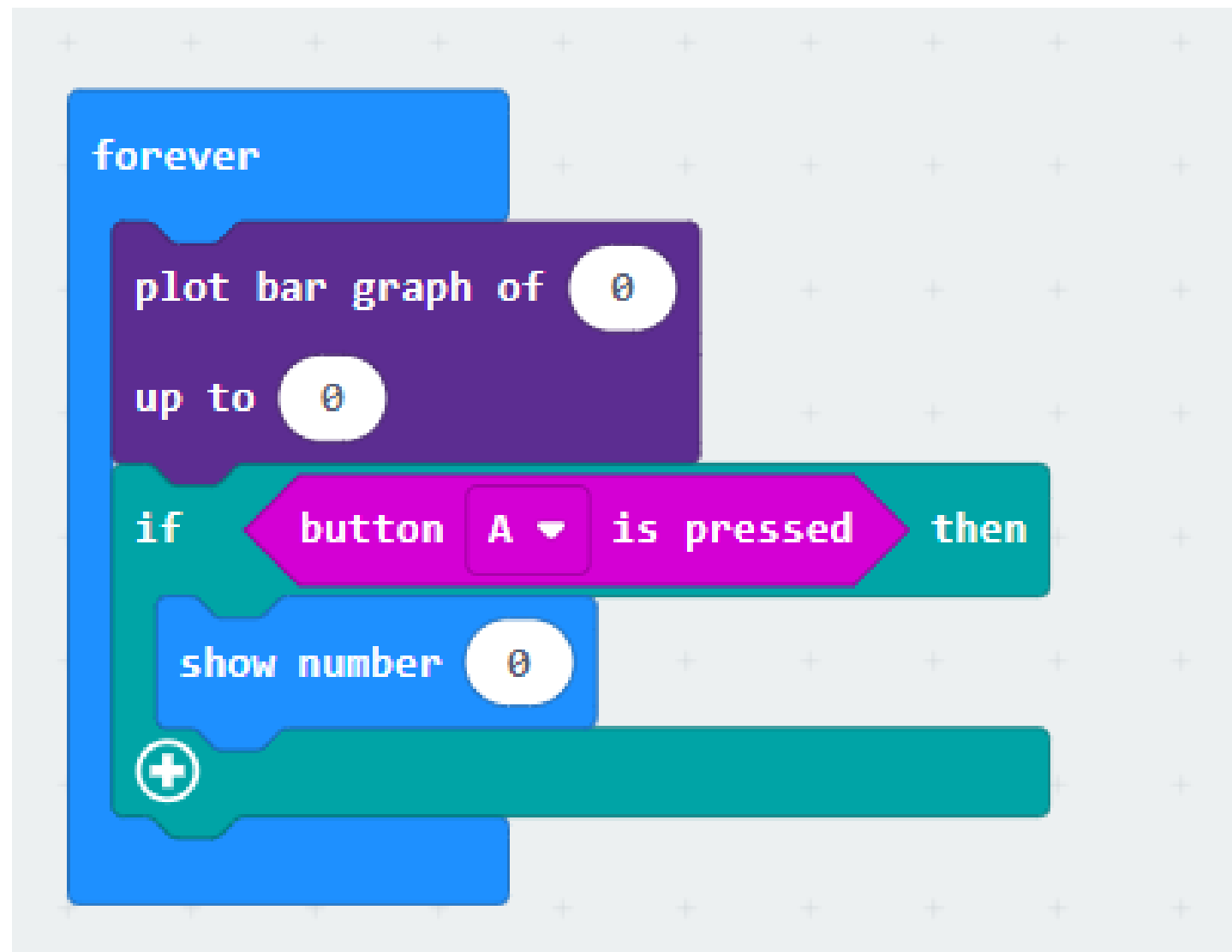
```
forever loop:  
  set reading to analog read pin P0  
  plot bar graph of reading  
  up to 1023  
  if button A is pressed then  
    show number reading
```

Take away all the details that are specific to a single program.

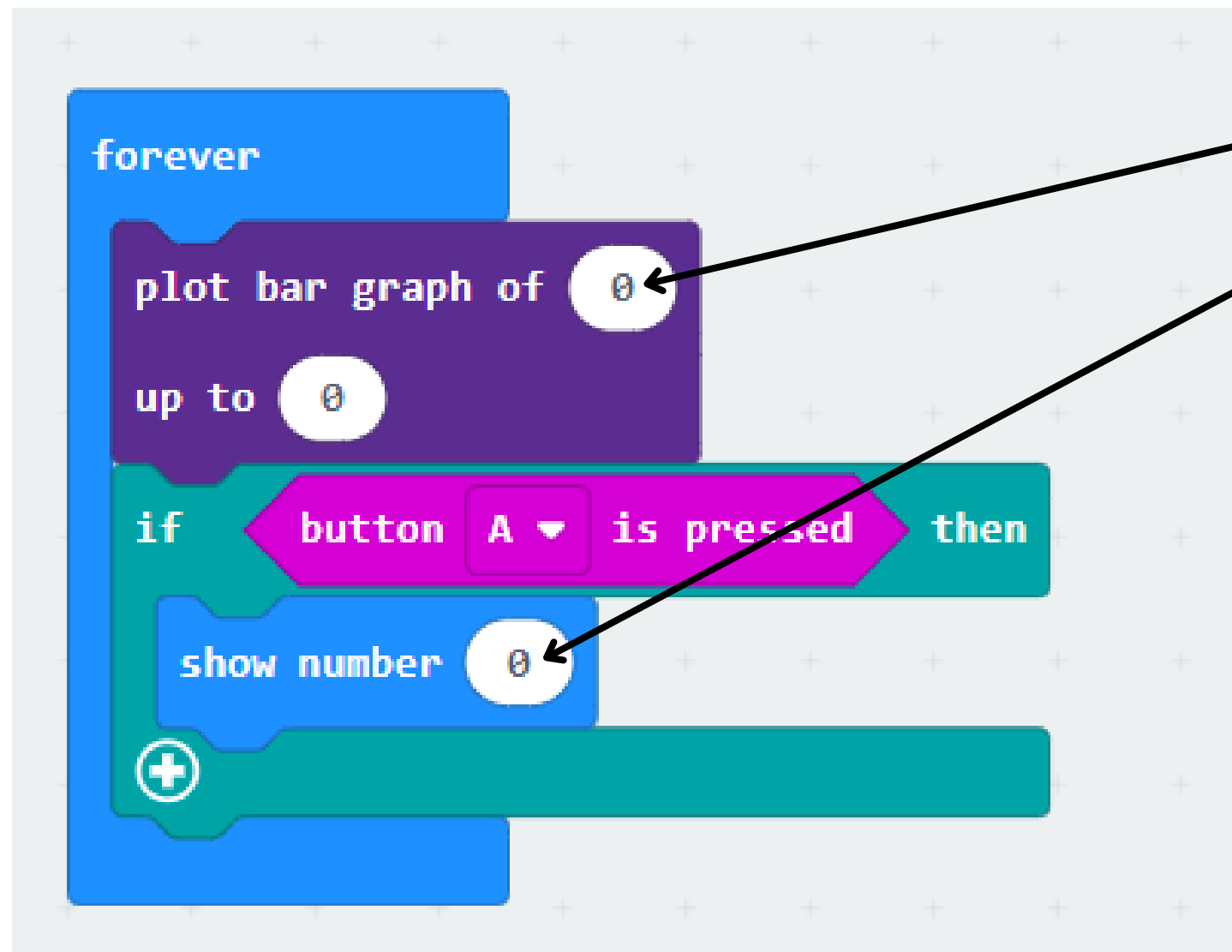
Leave everything else.



# It looks a little something like this...



# Now fill in the blanks with details...



Choose your input

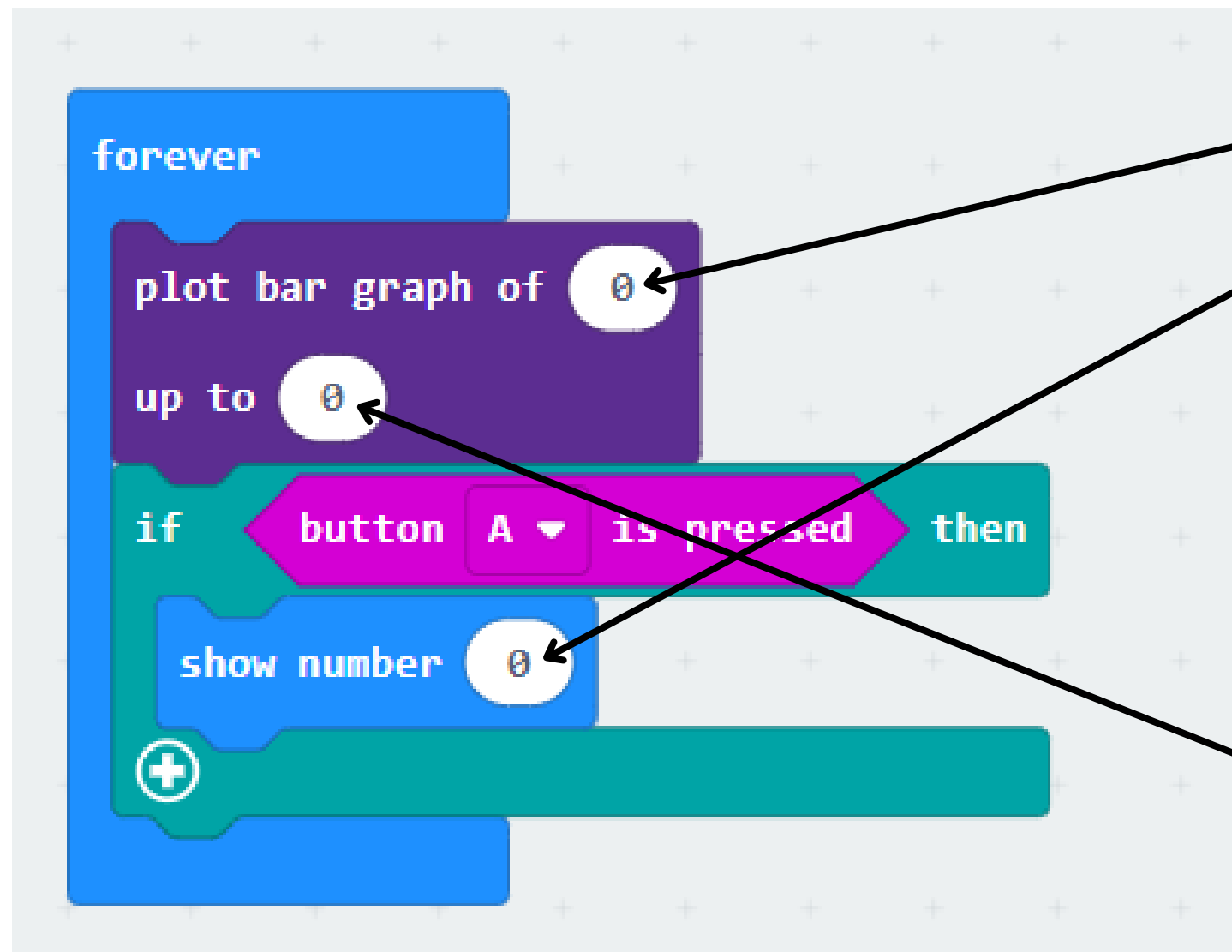
Light level

Temperature

Create a variable  
(Soil moisture)



# Now fill in the blanks with details...



Choose your input

Light level

Temperature

Create a variable  
(Soil moisture)

Type in the maximum number to graph

255



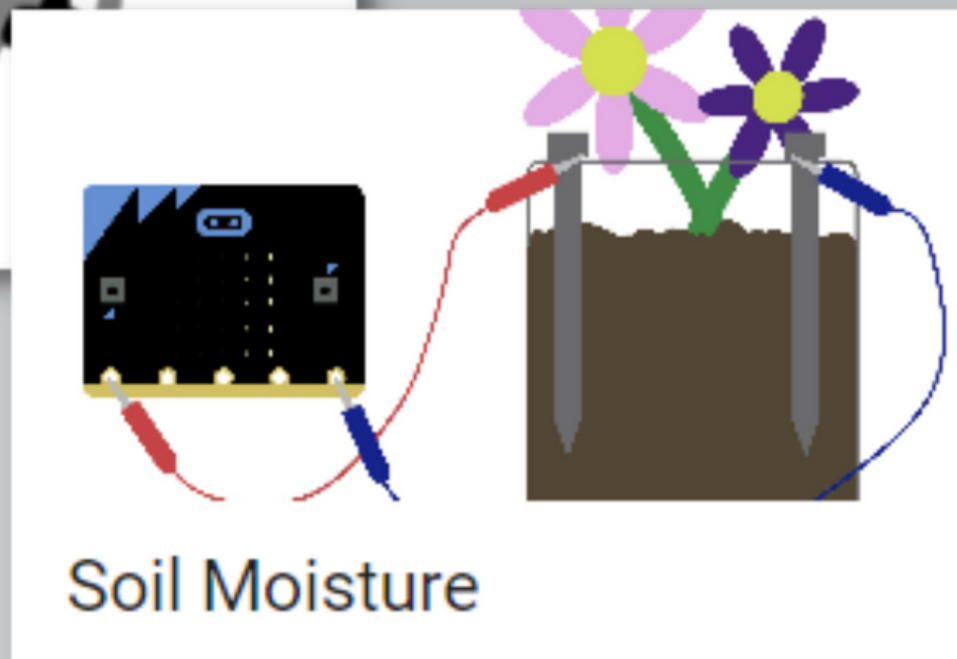
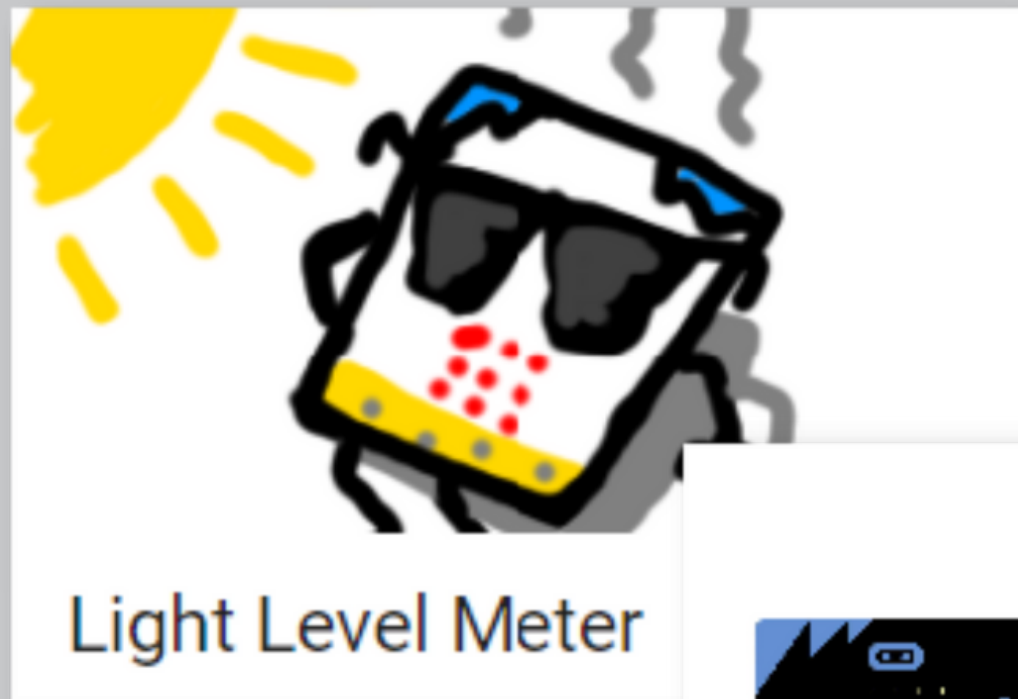
37.7



1023



# Need help? You've got options!



- Use **tutorials on MakeCode** for light level and soil moisture sensing



## Measure soil moisture

How wet or dry is your garden's soil? Code, build, and use this soil moisture sensor to find out. Look for dry and wet areas outside.

Materials needed:

- Microbit



## Measure temperature

Are there areas in your garden that are warmer or cooler? Code this sensor to help measure temperature in the sun, shade, or anywhere in between.



## Measure light intensity

How sunny or shady is it in your garden? How does this compare to inside your house or school? Code this sensor to measure brightness.

Materials needed:

- Microbit
- Computer with internet access
- USB cord
- Battery pack (if you want to take your microbit outside)

Short on time? [Download the finished code!](#)

Need a video tutorial? [Watch one here.](#)

- Check out our **how-to coding guides** on our website
  - Read the **written instructions with diagrams**
  - Watch the **video tutorials**
  - **Download the finished code** directly