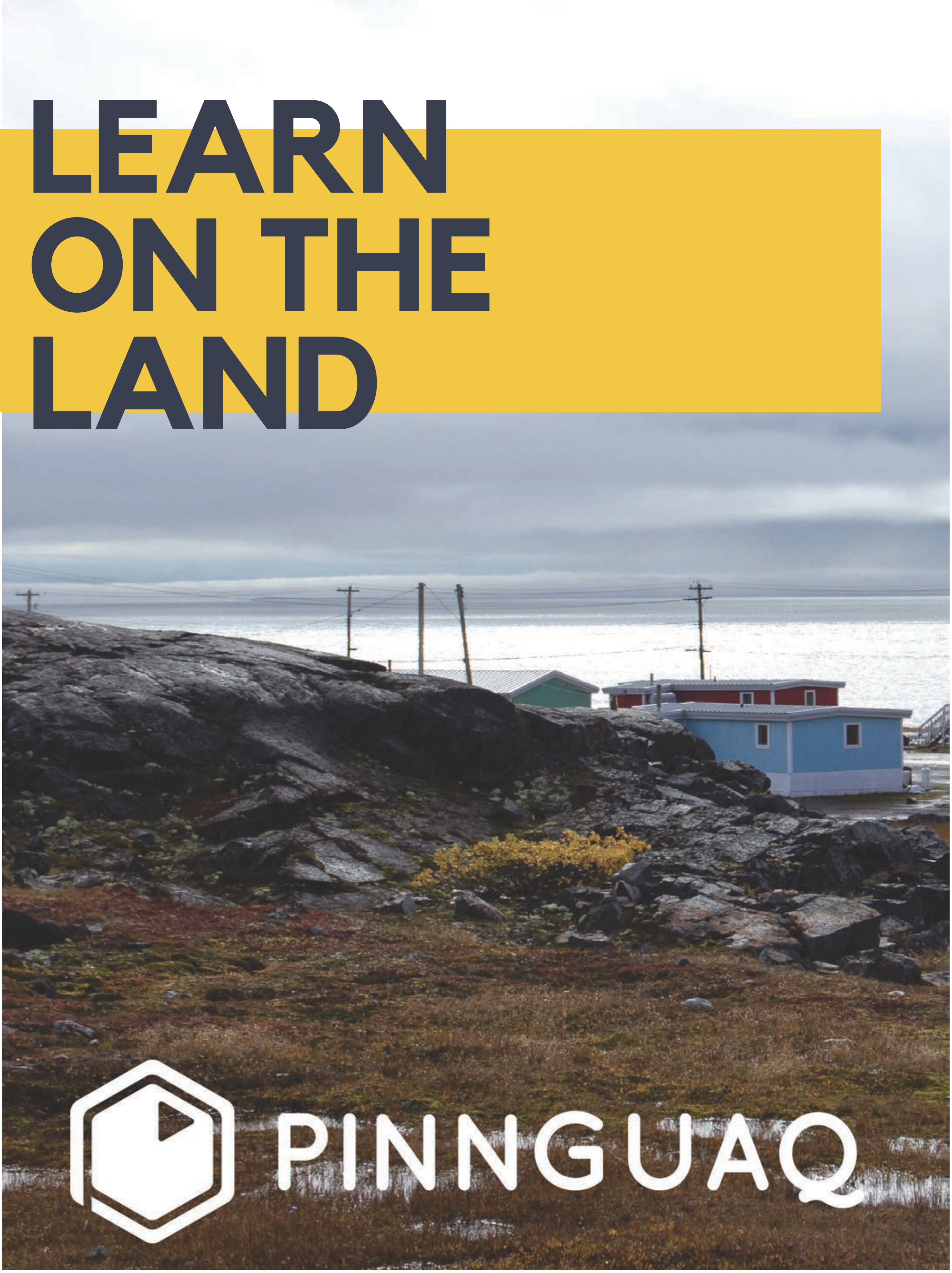
LEARN ON THE LAND



NOVEMBER 2021

PINNGUAQ

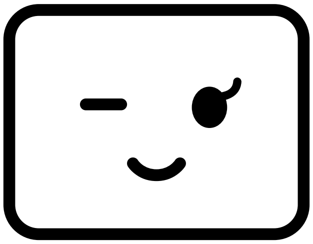
Welcome!

We are so glad to offer our Learning on the Land through STEAM kit to you! In this kit, we will explore our environment and our role as Environmental Stewards. We will read about the **Innunguiniq** principles (how to become a capable human being) to learn about our responsibilities towards our environment. In this kit, you can find instructions and information on how to complete each activity.

We hope you enjoy our STEAM activities, and that you have fun coding your own projects!

Sincerely,

Your friends at Pinnguaq



ACTIVITY OVERVIEW

01

INTRO TO THE BBC MICRO:BIT

Learn how to use the micro:bit and the MakeCode editor.

02

ENVIRONMENTAL STEWARD NAME BADGE

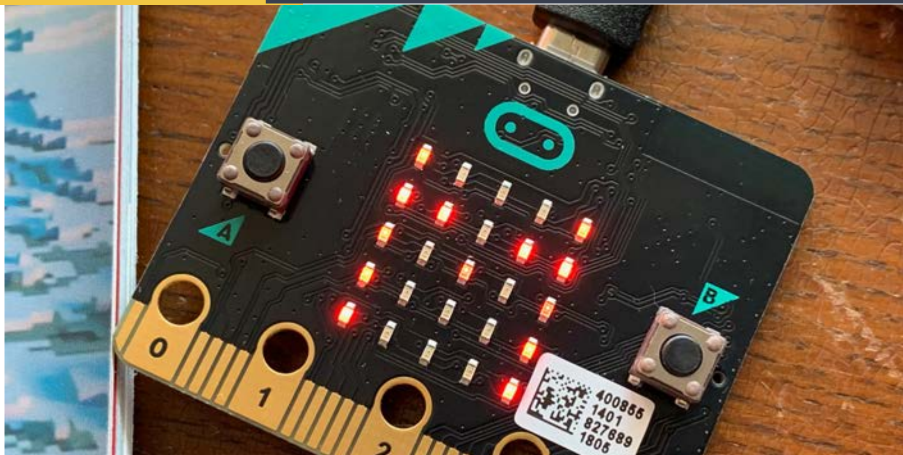
Code the micro:bit to display your name. Then make the microbit into a die to play the Race for Change game.

03

SPECIES COUNTER

Observe the species and make a counter!

PINNGUAQ



ACTIVITY OVERVIEW

04

COMPASS NORTH

Use the micro:bit as a compass!

05

HOW COLD IS IT?

Add the enviro:bit to take temperature and pressure readings.

06

HOW NOISY IS IT?

Use the enviro:bit to learn about noise pollution.

PINNGUAQ

ACTIVITY OVERVIEW

07

SIMPLE NIGHT LIGHT

Use the enviro:bit to make a night light.

08

WEATHER STATION

Combine all the weather sensors in the enviro:bit to create a weather station.

FINDING HOME

Use micro:bits radio signals to represent whale communication and show how noise pollution affects whales.

09

PINNGUAQ

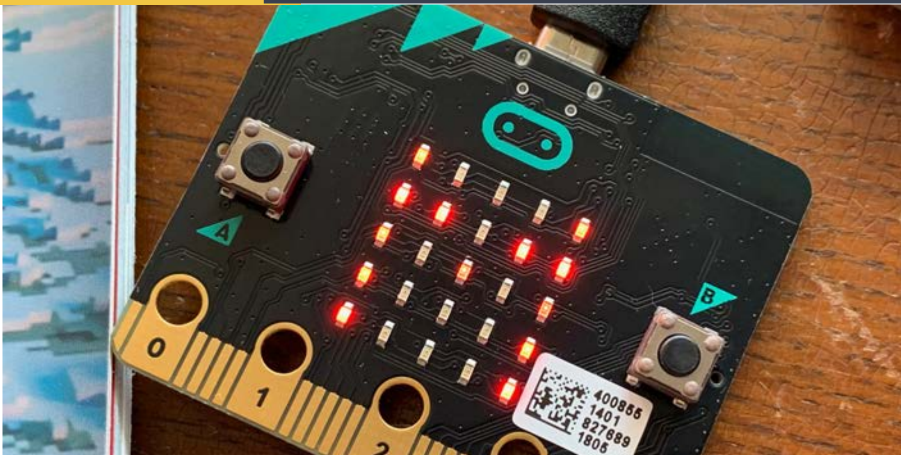


TABLE OF CONTENTS

TABLE OF CONTENTS

01 MICRO:BIT INTRODUCION ACTIVITY

OVERVIEW

Let's get your Micro:bit set up and get to know it. The micro:bit and enviro:bit technology will help you become an Environmental Steward and follow the Inuit principles of Qanuqtururangniq and Avattimik Kamatsiarniq. The micro:bit and the enviro:bit have incredible sensors that will give you information about your environment! It is important to observe the world around you as an Environmental Steward.

Definition:

Inuit Qaujimajatuqangit- "The extensive knowledge and experience passed from generation to generation" (Inuit Qaujimajatuqangit What Inuit Have Always Known to Be True) https://fernwoodpublishing.ca/book/inuit-qaujimajatuqangit

BACKGROUND INFO

The micro:bit is a pocket-sized computer.

You can program this mini computer with your computer and the MakeCode editor. You can use the online editor, or the offline editor that is installed onto your computer and loaded onto the USB. Or, you can install the offline editor that is on the USB stick that came in your kit onto any computer. Follow the steps below to get to know how to code and use your Micro:bit.

VOCABULARY

Micro:bit is a pocket sized computer with an LED display, and information inputs and outputs.

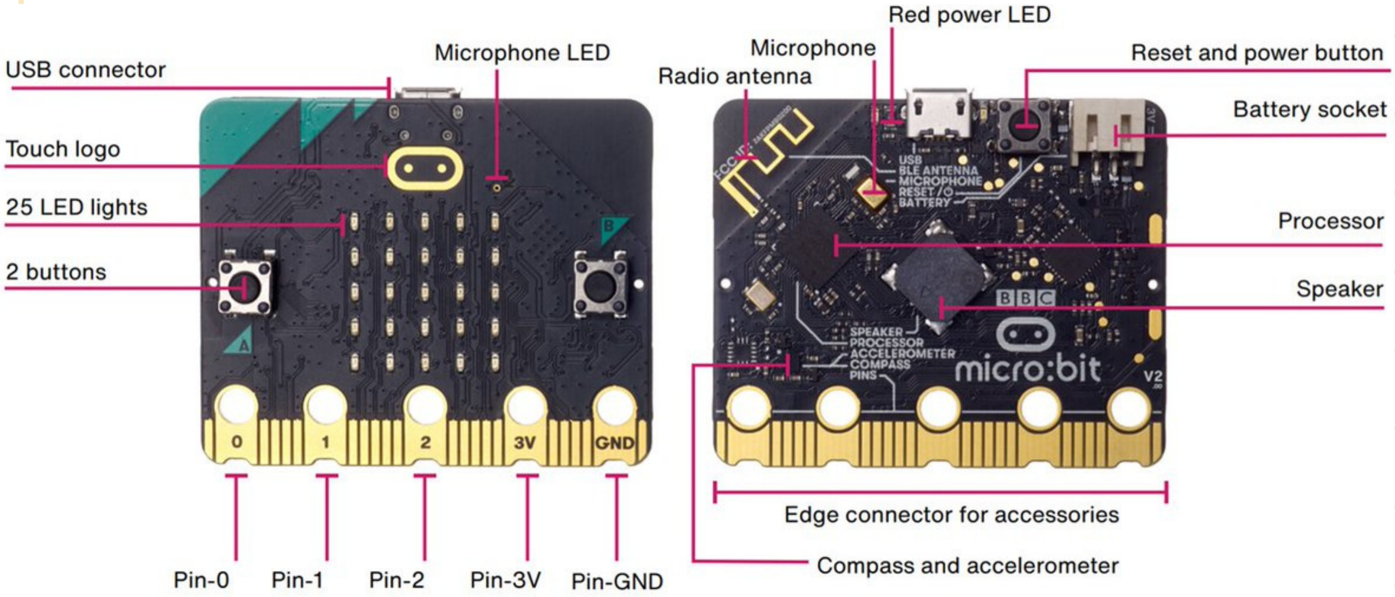
Code is the language we use to communicate with computers.

Materials you will need:

- a micro:bit
- two (3) AAA batteries
- A battery pack

01 GET TO KNOW YOUR MICRO:BIT! PLUG IT IN AND BE SURPRISED!

- Put the 2 AAA batteries into the battery pack.
- Match the + and - symbols to put each battery in correctly.
- Plug the battery pack into the micro:bit. Plug the battery pack into the white box at the top left corner of the micro:bit.
- Read the Hello message on the micro:bit display.
- Follow the arrow and press Button A.
- Follow the arrow and press Button B.
- Shake the micro:bit and watch the display.
- Play chase the dot by tilting your micro:bit.
- Clap 5 times. (New for V2)
- Now you are all set to learn how to code your micro:bit!
- In Activity Two, you will create a name badge with your micro:bit.

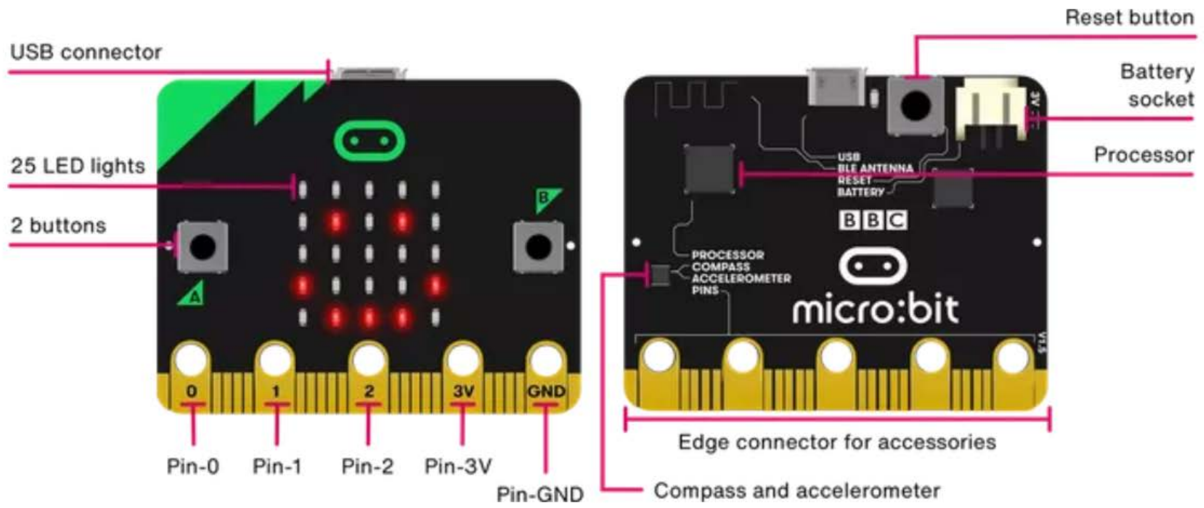


- .The Micro:bit has inputs (ways to take in information) and outputs (ways to send out information).
- Your Micro:bit is just like any other computer and will only do what you tell it to do.

TAKE A CLOSER LOOK AT YOUR MICRO:BIT –– V1 AND V2

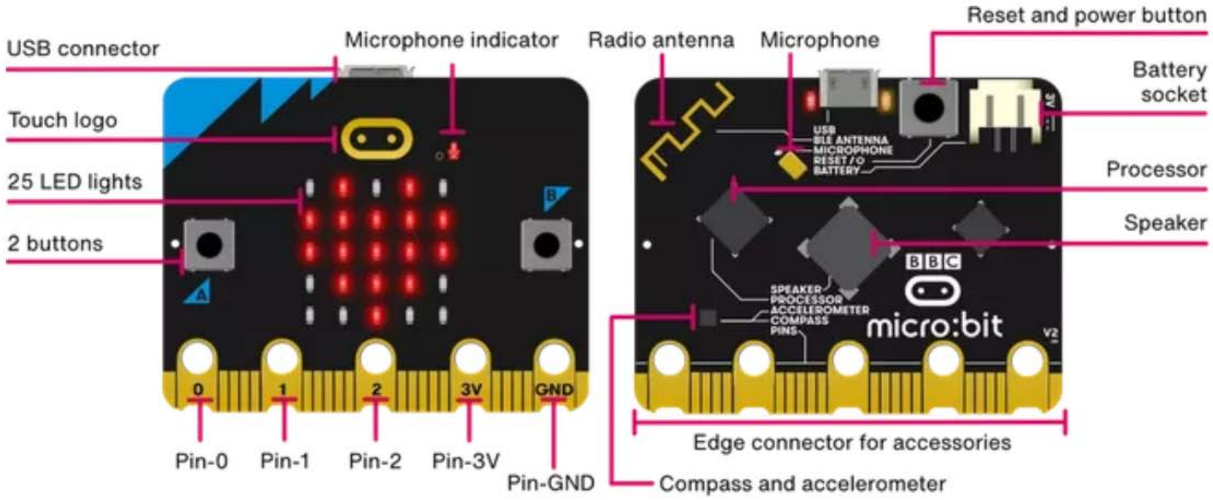
V1

Original micro:bit



V2

New micro:bit with sound

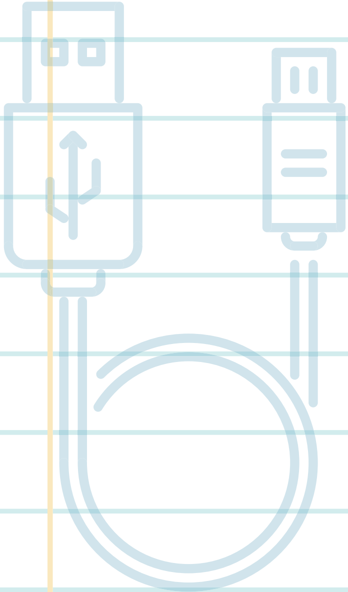


YOU CAN DO ALL OF THE ACTIVITIES IN THIS KIT WITH EITHER MICRO:BIT V1 OR V2

MATERIALS NEEDED FOR ACTVITIES

- Micro:bit
- Makecode editor
- Computer/Laptop
- USB Cord
- Enviro:bit

* Refer to page 25 when adding the extension.



02 ENVIRONMENTAL STEWARD NAME BADGE

OVERVIEW

Let's Get set up with the Microbit and make a fun Environmental Steward Name badge to get to know how to code the micro:bit.

BACKGROUND INFO

The Micro:bit is a mini computer. You can program the micro:bit computer using code. Code is the language we use to tell computers what to do. We are going to use the MakeCode language. You can use the online code editor https://makecode.microbit.org/ or use the offline code editor installed on your computer. The offline editor is also on the USB stick so you can install the offline MakeCode editor on any computer.

WHAT'S AN ENVIRONMENTAL STEWARD?

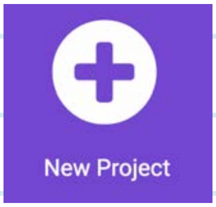
Environmental Stewards are people who watch and take care of our land and water. The Inuit values and beliefs show the importance of living a good life and taking care of our environment.

> "Avattimik kamatsiarniq requires we pay careful attention to the environment as an important part of survival and sustainability. The care and importance we place on this also impacts on our personal health and well-being. For all of these reasons, it is important to think about the environment we grow up in and how we become good caretakers and good contributors to a healthy and positive environment."
>
> -Shirley Tagalik, on Inuit Principles of Conservation

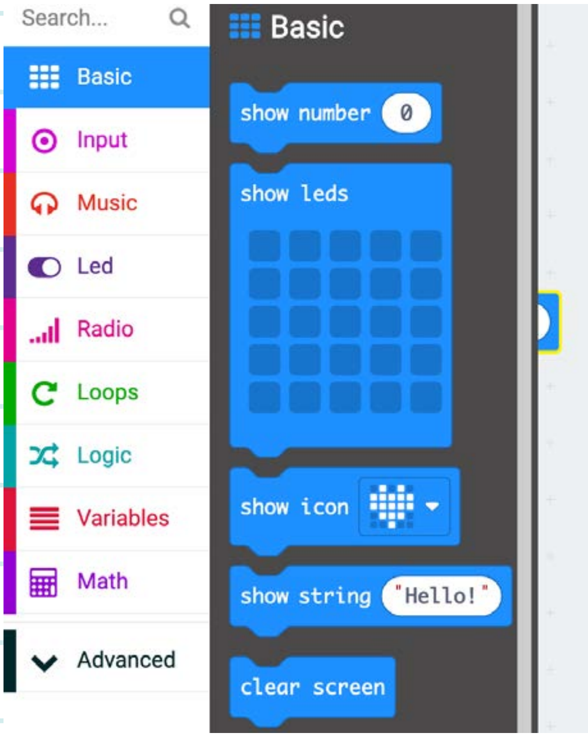
STEP BY STEP

1 Go to the microbit MakeCode editor online https://makecode.microbit.org/ or
the offline editor on your computer.

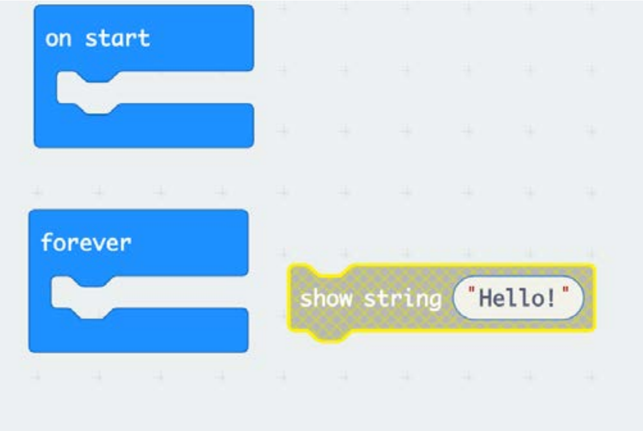
2 Click to create a New Project.
Give your project a name.
For example: "Name Badge".



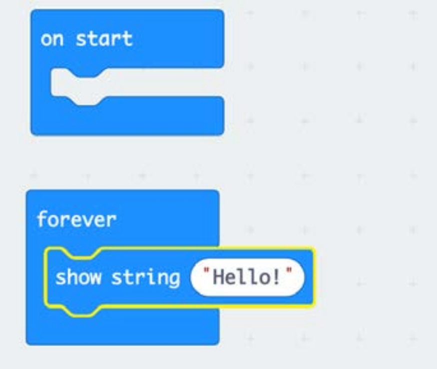
3 Click on the blue Basic block tab and find: "Show String".



4 Drag the "Show string" block onto the coding area.

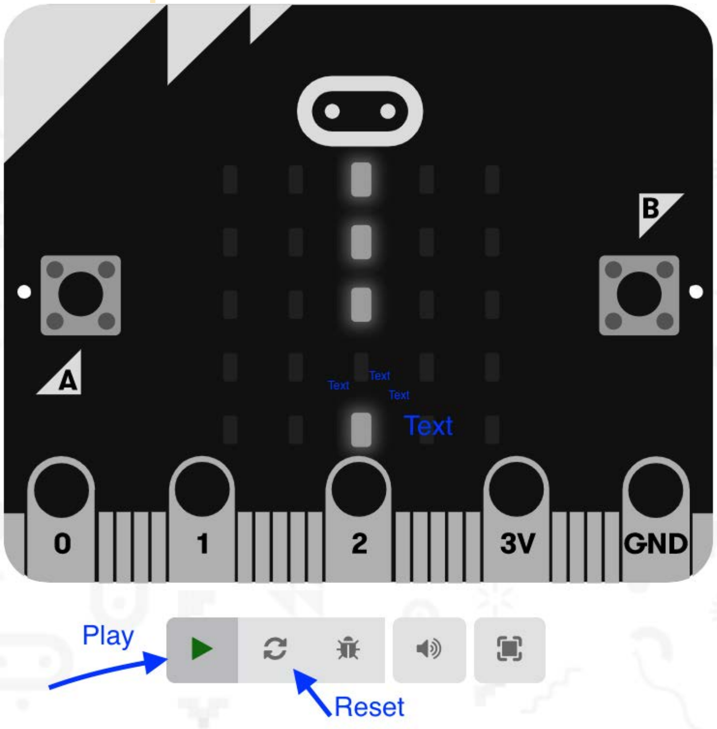


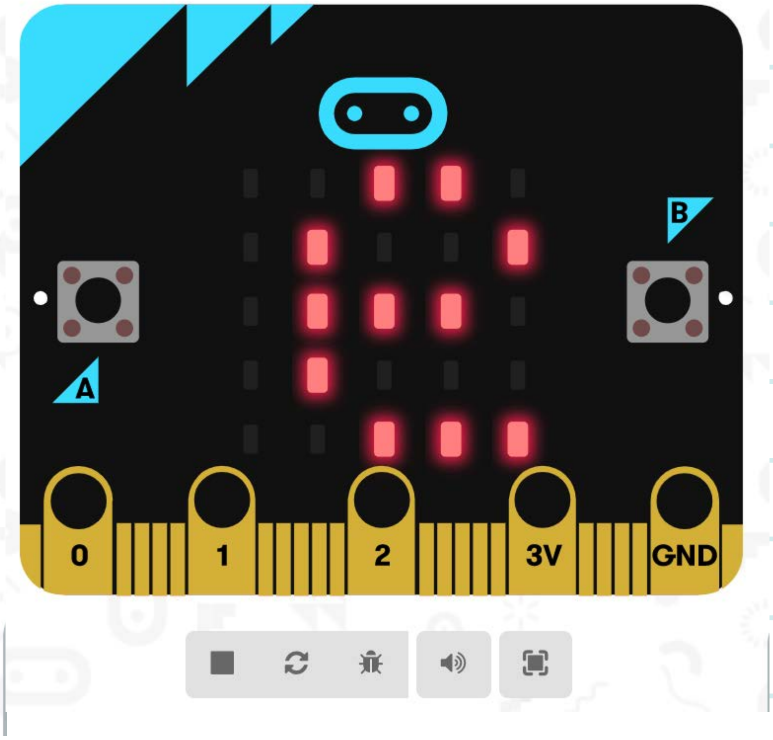
5 Drag the "show string" block into the space inside the Forever block. Computers do exactly as we tell them to do. The Forever block tells the computer to keep running the code we put inside the block (forever). The Forever block is a loop block.



STEP BY STEP

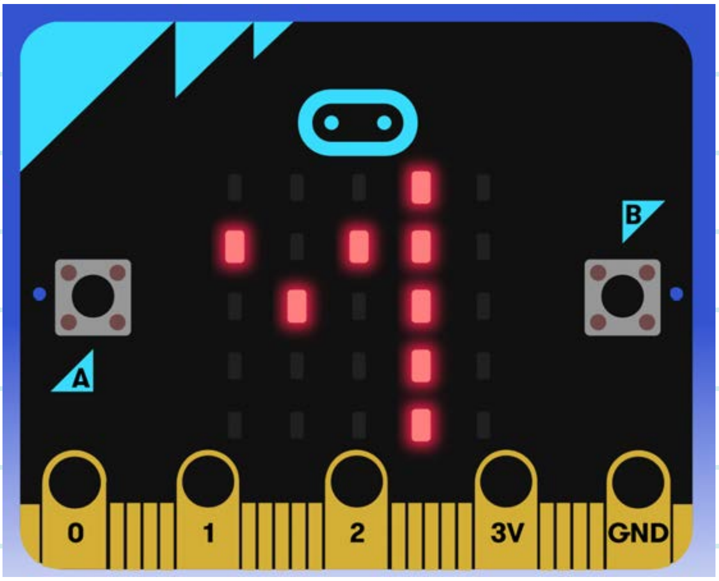
6 Click on the Play arrow to see the word Hello scroll across the display of the simulator. You may need to click on the reset symbol to start your code.





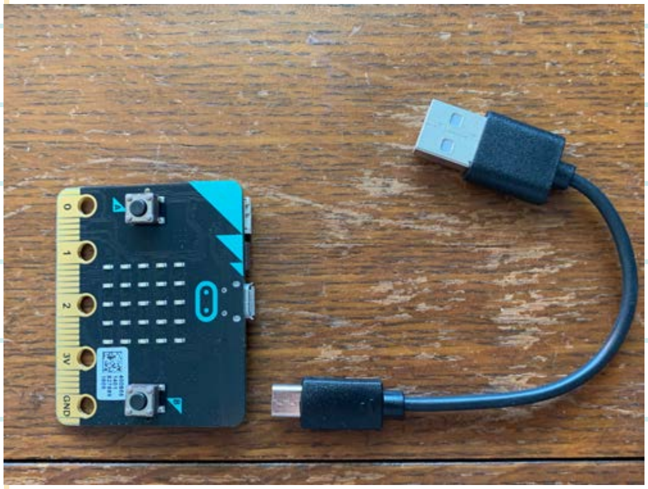
7 Double click on the word Hello inside the text bubble and type your name. Press enter. Make sure you see double quotation marks around your name like this. “Mary”

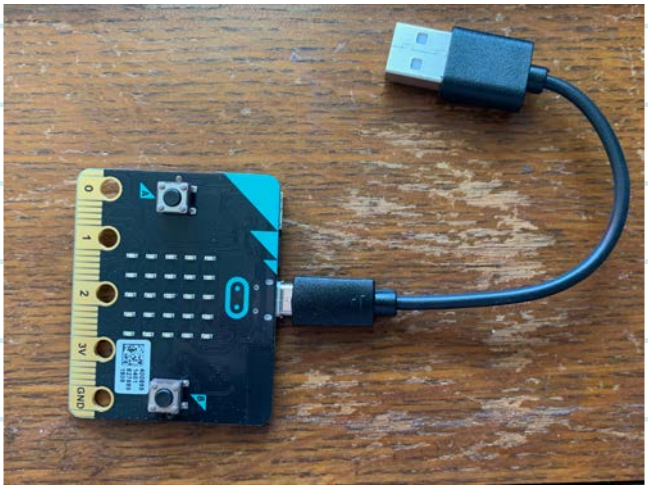
8 Now look at the simulator to see if you can see your name scrolling across the display of LED lights! (In the picture below you can see part of the M for my name, Mary.)



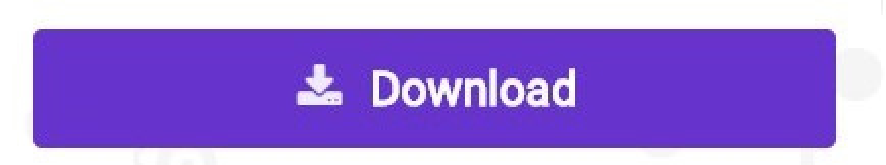
STEP BY STEP

9 Connect the mini USB cord from your kit to your micro:bit, then connect your micro:bit to the computer with the micro USB cord.



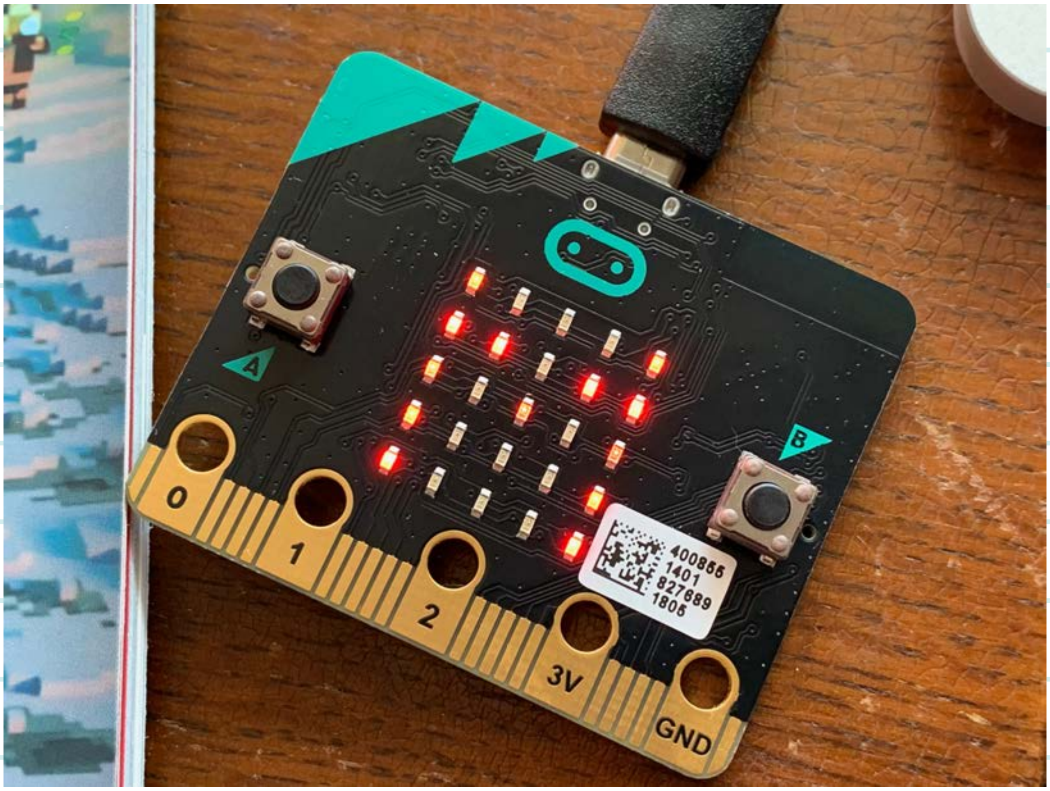


11 Click on the Download Button. This will transfer (flash) your code file to your micro:bit. The single LED light on the back of your micro:bit will flash when the code is downloading. When it is finished flashing (downloading) the file, your code will automatically start running on your micro:bit.

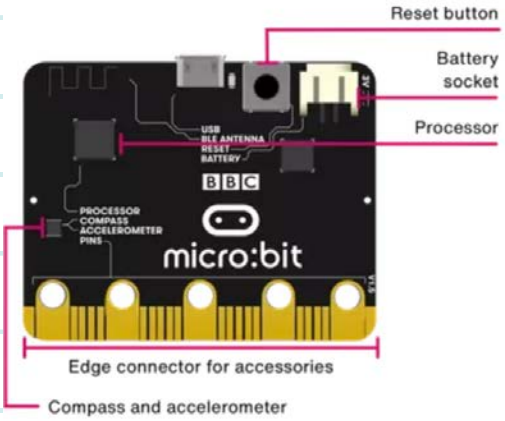


STEP BY STEP

12 When the code has finished downloading, you will see your name scrolling across the LED display!



13 Troubleshooting: If your name isn't showing, check your code. Next, try pressing the reset button on the back of your micro:bit.



Congratulations! You are an Environmental Steward!

02 MICRO:BIT DICE AND RACE FOR CHANGE GAME

OVERVIEW

Turn your micro:bit into a die (one) or dice (two) to use to play the Race for Change Board game in the publication.

BACKGROUND INFO

Open the publication to page 36 and 37 and play this board game to explore issues surrounding climate change. Learn about what is causing the climate crisis and how we can work to prevent it.

The game was created by Alana McCarthy. Alana is an illustrator, letterer and designer from Toronto.

VOCABULARY

Micro:bit -- a pocket sized computer with LED display, and information inputs and outputs.

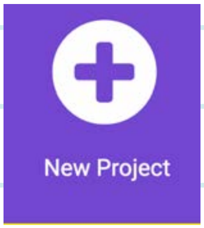
Enviro:bit -- an accessory for the micro:bit with powerful sensors that let you do more with your micro:bit.

Accelerometer -- a motion sensor and is one of the sensors on the micro:bit. It measures the acceleration, or movement of objects.

Random -- an action that happens without order or reason.

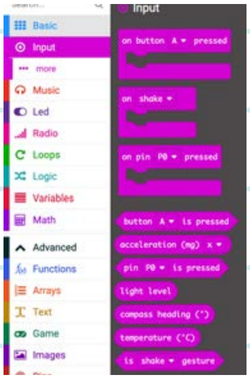
STEP BY STEP

Using the accelerometer of the micro:bit we can create either one die or a pair of dice. We can code the micro:bit to show random numbers when we shake it.

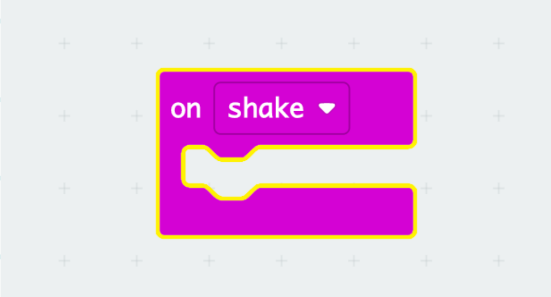


1 Open the make code editor and click the plus icon to create a new project. we will give our project a name, for example: "Dice".

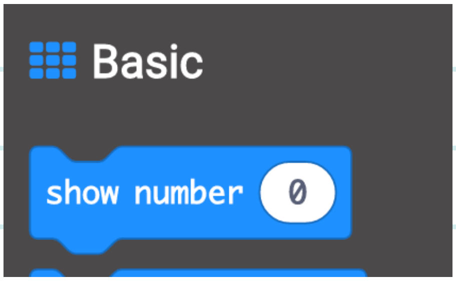
2 Click on the Input tab.

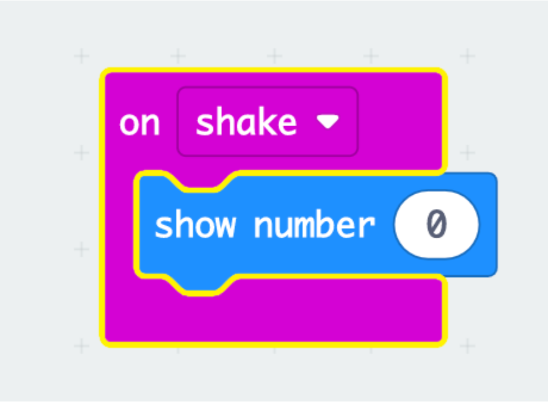


3 Drag the purple on shake block into the coding area. This means that shaking your micro:bit is the input and will trigger the micro:bit to run the code.



4 From the blue Basic tab, Drag the show number block. Put this block into the middle of the on shake block.

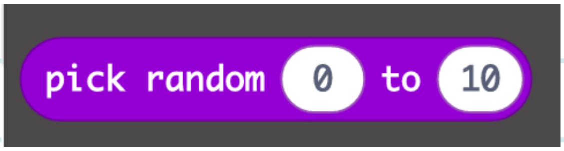




STEP BY STEP

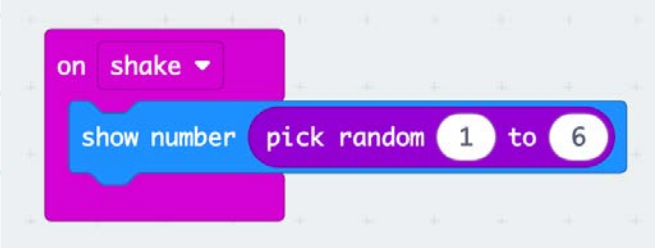
5 Each die (one of the dice) shows numbers from 1 to 6. When you roll the dice, you get a random number. You can code your micro:bit to show numbers from 1 to 6 to be one die, which is just one of two dice. Or, you can code it to show numbers from 1 to 12 to be two dice.

6 From the purple Math tab, choose the pick random block.

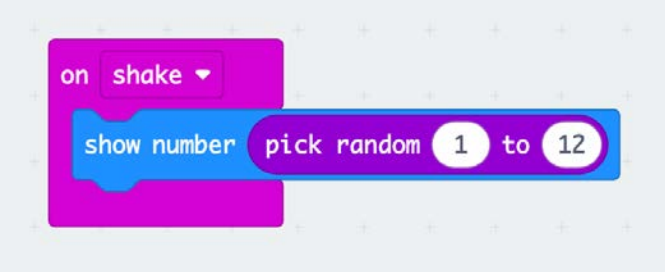


```
pick random 0 to 10
```

7 Drag this block into the oval opening in the show number block. Type in the numbers 1 to 6 to make your micro:bit into one die, or 1 to 12 to be two dice. I've made mine into one die.



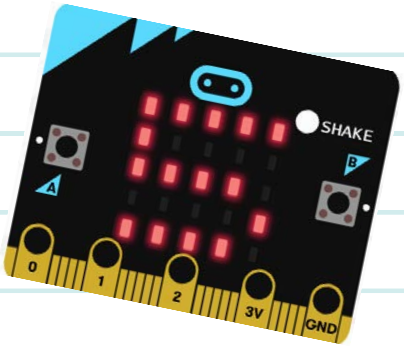
```
on shake
    show number pick random 1 to 6
```



```
on shake
    show number pick random 1 to 12
```

8 Connect your micro:bit to your computer with the USB cord that came in your kit, and select Download. The single LED light on the back of your micro:bit will flash while the code is downloading.

9 Now shake your micro:bit and test your dice! Now you are ready to play the game!



03 SPOT THE SPECIES COUNTER

OVERVIEW

In this activity, we will think about the wildlife in our local habitat. We will observe what species live in our local environment and create a counter to track them.

BACKGROUND INFO

WHY IS OUR LOCAL WILDLIFE IMPORTANT?

Think about the local ecosystems. What habitats exist in our environment. What changes are happening? Global warming and climate change is happening quickly, and it's affecting the environment. This affects how long we can access the sea ice for travel, hunting, and fishing. Humans are not the only beings affected by climate change. Plants and animals are also affected, and it's crucial to observe their patterns so we can predict how they will react to these changes. We can observe the species in our environment by studying:

- Their migration: where they go and for how long
- The population: how many of them live in a certain place.

Our job as environmental stewards is to note the changes that are happening to the plants and animals in our environment and take action to protect them. As Inuit say: "our actions come back to us."

VOCABULARY

Ecosystem -- an ecosystem is a community of different living organisms that live and interact with each other in an environment

Habitat -- a place where a living organism makes its home

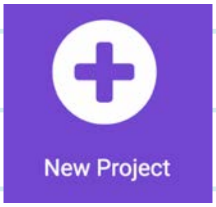
Climate change -- Climate change is the process of the planet getting warmer. This means that the weather will be more extreme and unpredictable.

STEP BY STEP

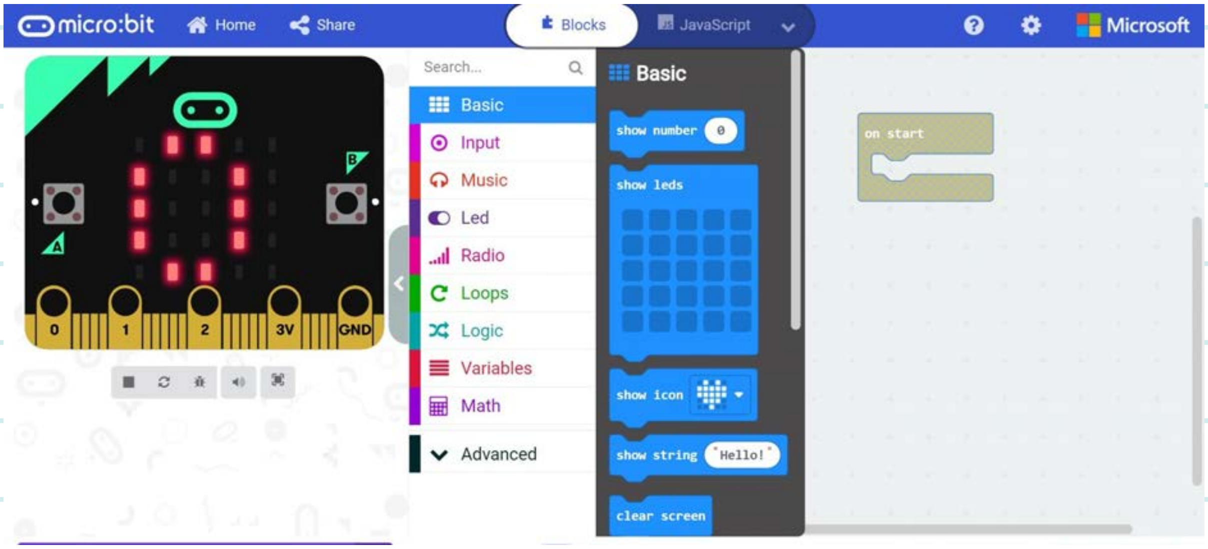
! We want our micro:bit to track the wildlife we see. We will add code to our project so that when we press a button, the program will add one to the counter and keep track of how many times we've added to it.

1 Choose an animal from the local environment you want to study.

2 Open the make code editor and click the plus icon to create a new project. we will give our project a name, for example: "Arctic Fox Counter."

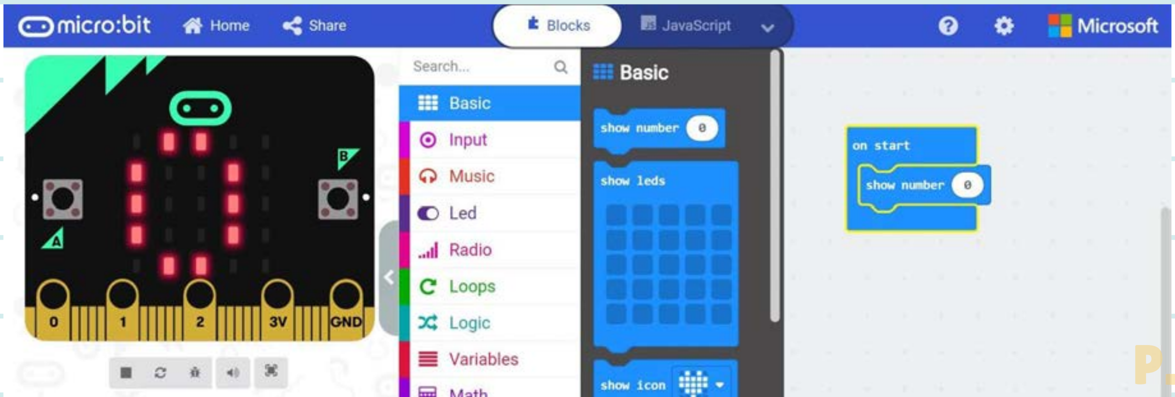


3 We want the counter to start at 0. In the code library on the left, click the first tab to find the blue "on start" block and drag it to the code area.



```
on start
```

4 From the same library, click and drag the "show number 0" block to the workspace. Place it under the first block. This will show us how many species we've tracked so far.

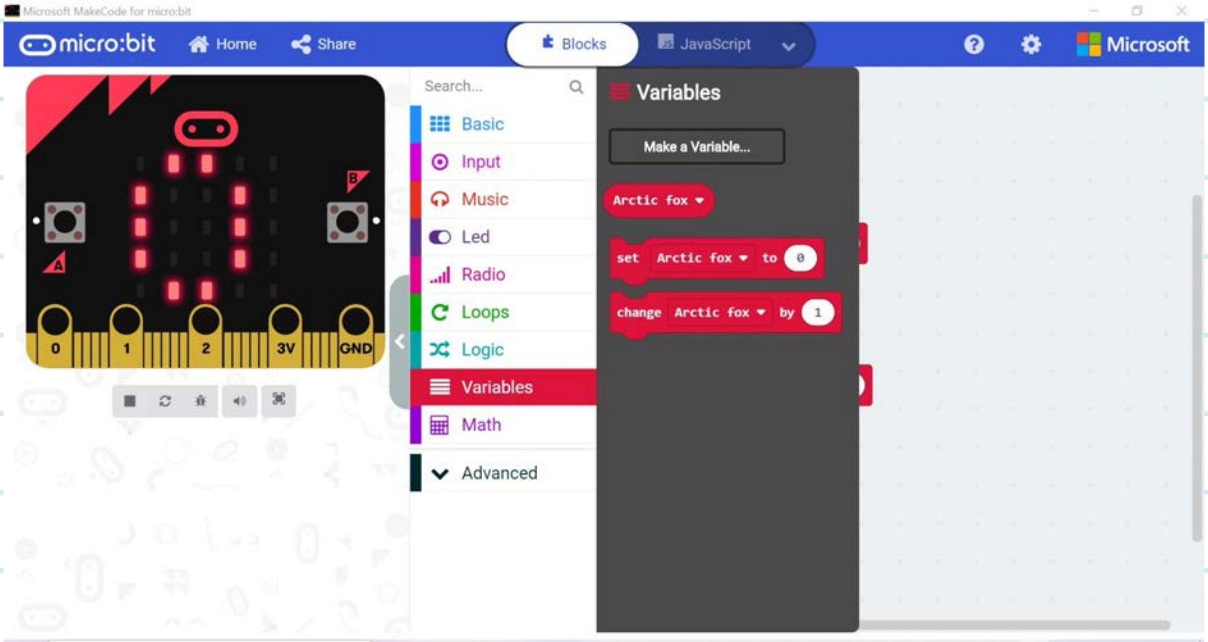


```
on start
    show number 0
```

STEP BY STEP

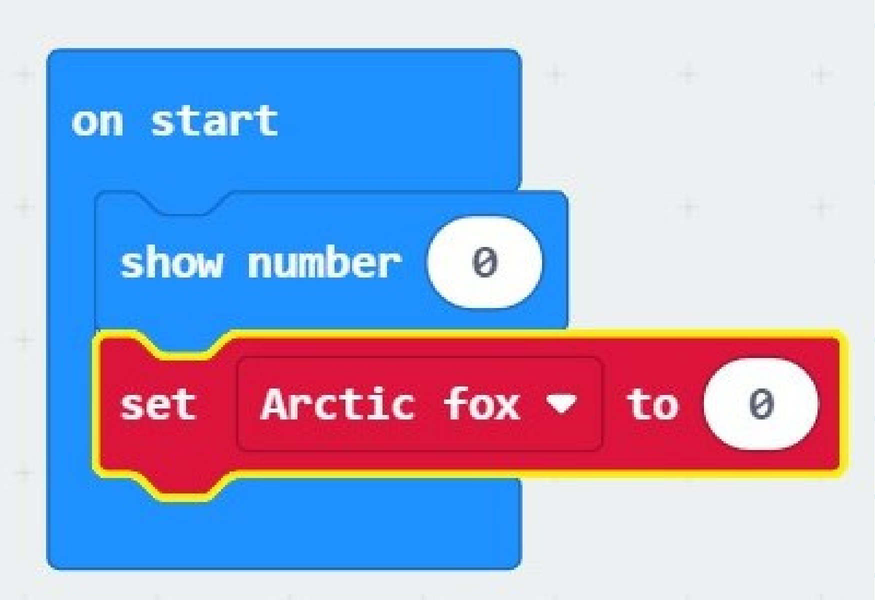
! The “Variables” blocks are handy for keeping track of things that change. We will use these blocks to keep count of the arctic fox or other species we are tracking.

5 In the code library on the left, click the red “Variables” tab. Click “make a new variable” and give it the name of the species you chose in Step 1.



```
Arctic fox
set Arctic fox to 0
change Arctic fox by 1
```

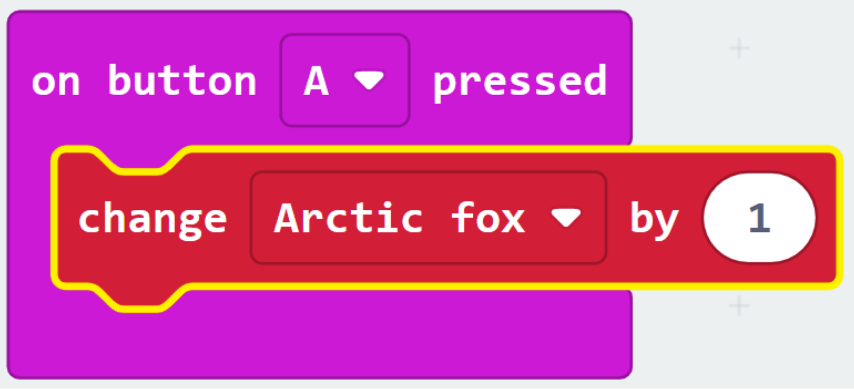
6 Drag the red “Set Arctic fox to 0” block to the code area and place it under the blue code blocks.



```
on start
  show number 0
  set Arctic fox to 0
```

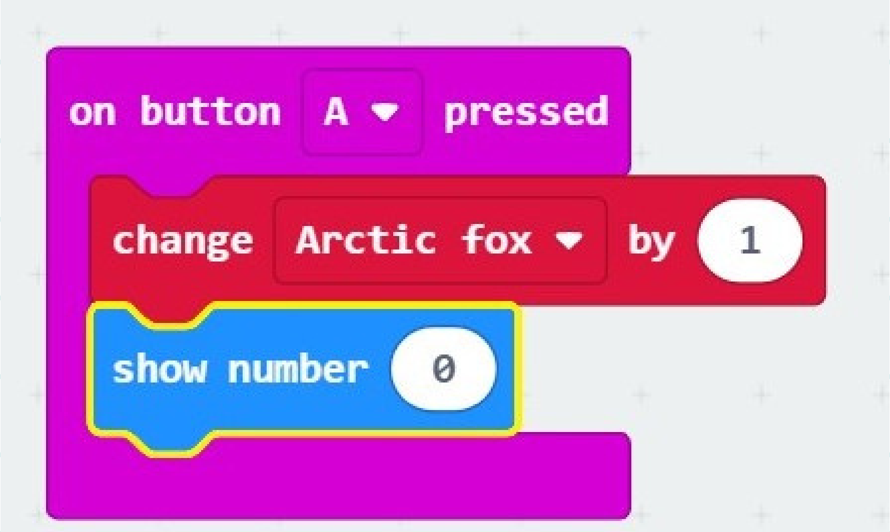
STEP BY STEP

7 We need the counter to change the number and show us how many species we have tracked when we press the A-button. In the red “Variables” tab, drag the “Change Arctic fox by 1” block and put it under the purple block.



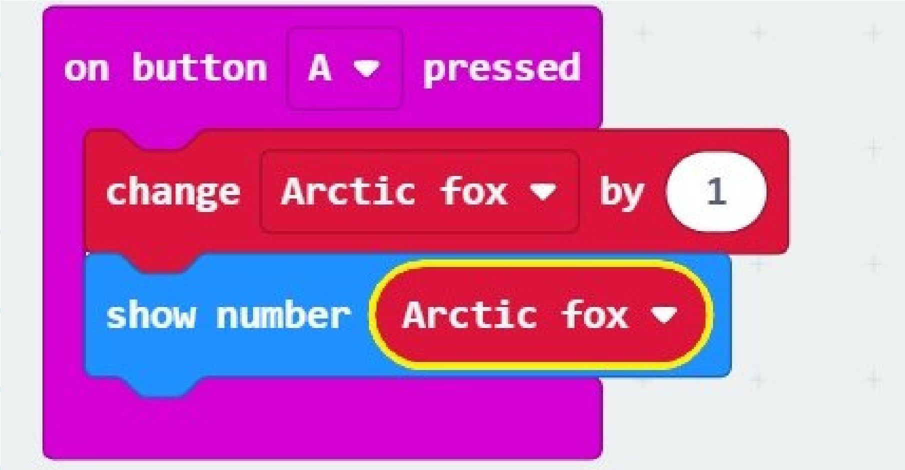
```
on button A ▾ pressed
    change Arctic fox ▾ by 1
```

8 We also want the micro:bit to show the number of times we pressed the A-button, so we will go back to the blue “Basics” tab and grab the “show number 0” block. Place it under the red “change Arctic fox by 1” block.



```
on button A ▾ pressed
    change Arctic fox ▾ by 1
    show number 0
```

9 Go back to the “Variables” tab and drag the red “Arctic fox” bubble to the “show number 0” block.



```
on button A ▾ pressed
    change Arctic fox ▾ by 1
    show number Arctic fox ▾
```

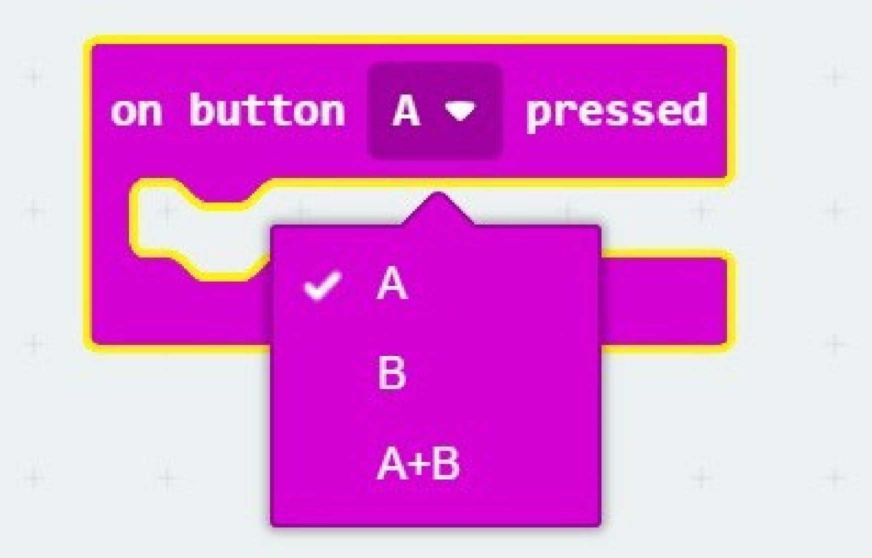
STEP BY STEP

10 Now try it out on the simulator on the left side of the screen! Click the A-button to add to the counter. Each time you press the a-button, the simulator will add one and show you on the screen.

SHOW THE COUNTER

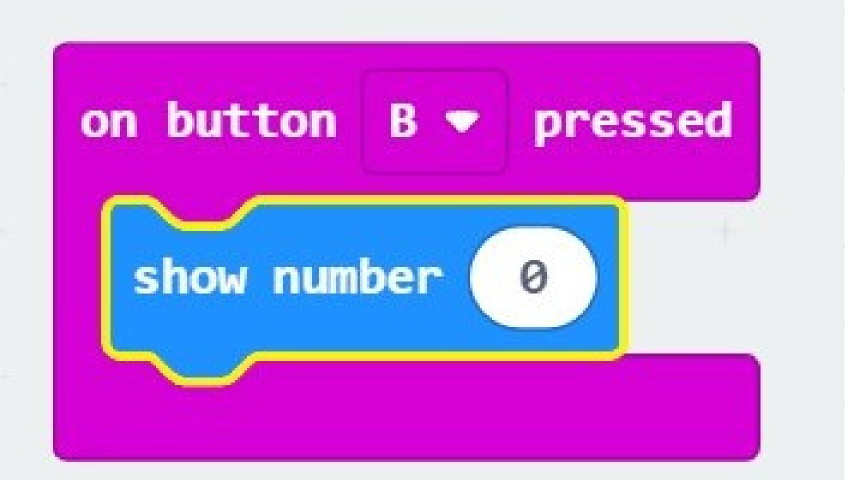
The micro:bit can only show one-digit numbers on the screen, so when we reach the number 10, it will scroll and then disappear. To solve this problem, we will add code to show us what number we have reached on the counter. We will add code to the B-button.

1 Click the second "Input" tab and select the "when A-button pressed" block and bring it to the code area. Change it from A to B by clicking the A and selecting B instead.



```
on button A ▾ pressed
```

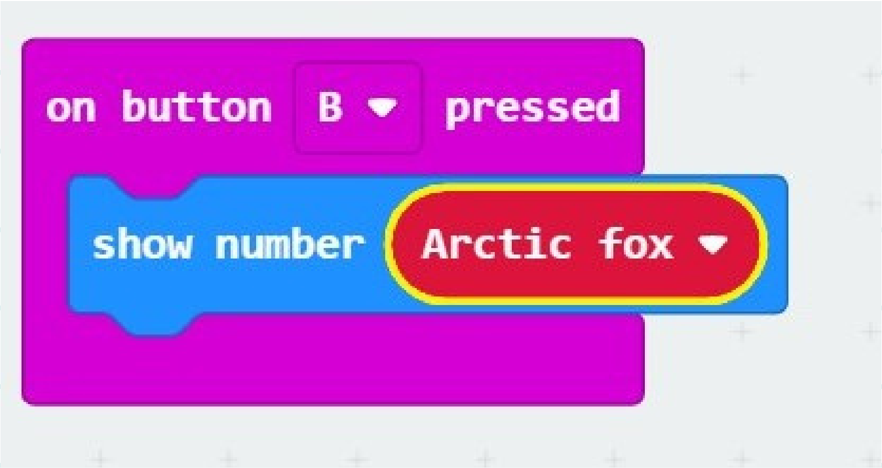
2 Add the code block "show number 0" from the "Basics" tab.



```
on button B ▾ pressed
  show number 0
```

STEP BY STEP

3 Add the red “arctic fox” bubble from the variables tab.

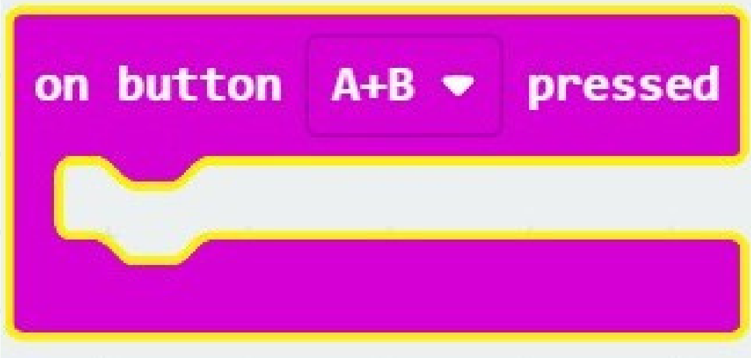


```
on button B ▾ pressed
    show number Arctic fox ▾
```

RESETTING THE COUNTER

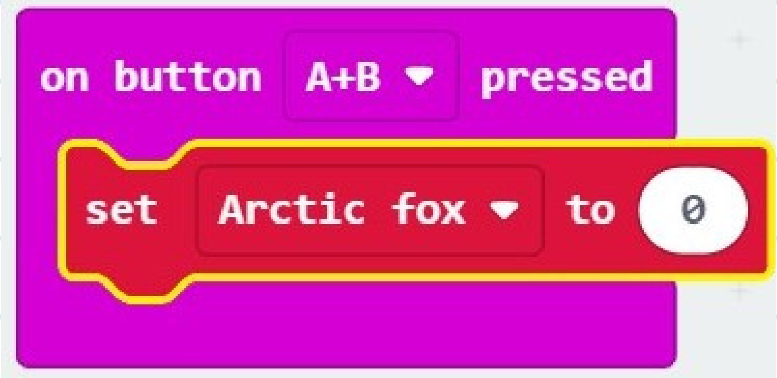
! We want to program the micro:bit to reset the counter when the A and B buttons are pressed at the same time.

1 In the library, drag the purple “when A button pressed” block to a space in the code area. Click the A button to open the menu and select A+B.



```
on button A+B ▾ pressed
```

2 From the “Variables” tab, add a red “set arctic fox to 0” block to this code.



```
on button A+B ▾ pressed
    set Arctic fox ▾ to 0
```

Now when you press both A and B buttons the counter will reset to 0!

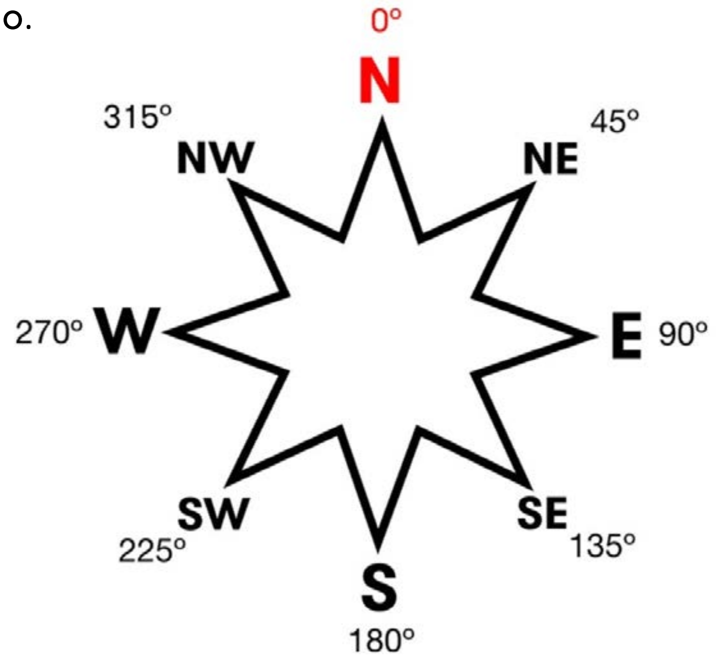
04 COMPASS NORTH

OVERVIEW

Let’s make a compass on the micro:bit!

HOW DOES IT WORK?

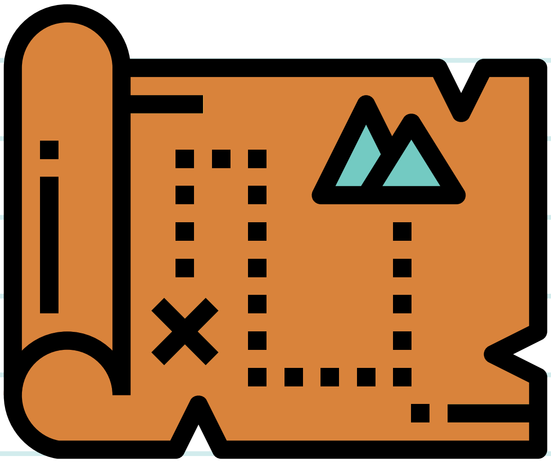
The micro:bit has a compass sensor called a magnetometer. It can sense the Earth’s magnetic field and tell us the north and south poles’ direction. When we press A, the micro:bit will take a reading from the magnetometer (compass sensor), and it will show us the numerical compass bearing on the LED display. If you are facing north, the micro:bit compass will show 0 on the screen. You can use this diagram to understand the degrees and what direction they correspond to.



Here's a diagram of the cardinal points of a compass and the measurement they correspond to in degrees.

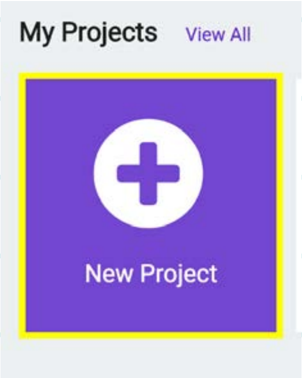
VOCABULARY

- **Compass** -- A tool for finding direction.
- **Magnetometer**-- an instrument for measuring the magnitude and direction of a magnetic field.
- **Magnetic field**-- The earth's magnetic field has two poles: the north and south.
- **Compass Heading**-- another word for the direction the compass is pointing towards. Compass directions are measured clockwise from north.

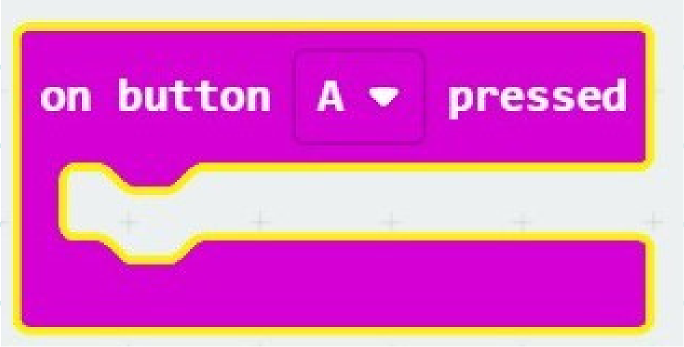


CODING THE COMPASS

1 Open the make code editor and click the plus icon to create a new project. we will give our project a name, for example: "Compass North."

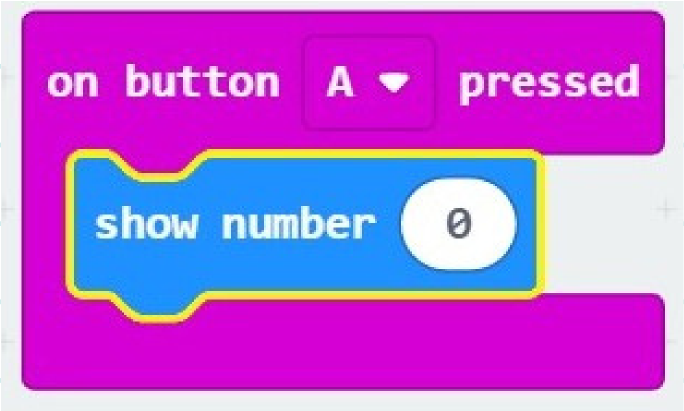


2 We want the micro:bit to take a reading when we press A. Open the “Input” tab and select the “on A button pressed” block.



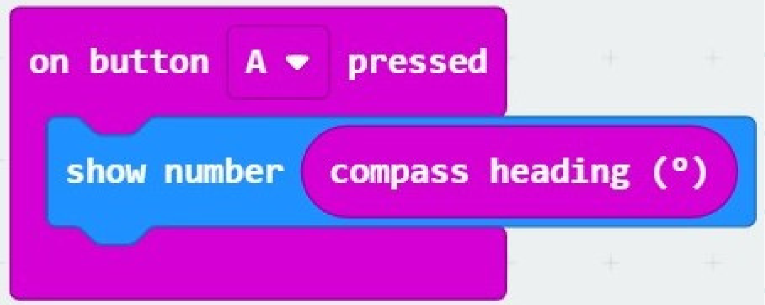
```
on button A ▾ pressed
```

3 We want the micro:bit to show us the reading it takes. Use the “show number 0” block in the blue “Basic” tab.



```
on button A ▾ pressed
    show number 0
```

4 The number we want the micro:bit to show is the compass heading. The heading tells us the micro:bit’s direction degrees. In the purple “Input” tab, select the “compass heading” bubble block.

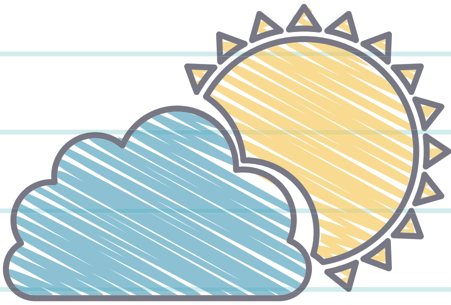


```
on button A ▾ pressed
    show number compass heading (°)
```

Now try out your compass! Use the diagram to understand the direction and the degrees they correspond to.

05

HOW COLD IS IT?



OVERVIEW

We will learn how to take a temperature readings with the enviro:bit!

HOW DOES IT WORK?

We will use the enviro:bit to take temperature readings. For this activity, we will take the enviro:bit out and take temperature readings. If it is too cold for the enviro:bit, try taking some inside temperature readings.

NON-COMPUTER ACTIVITY

Why is it important to measure temperature?

Read the comic strip titled: “bytesized” about the dangers of sea ice.

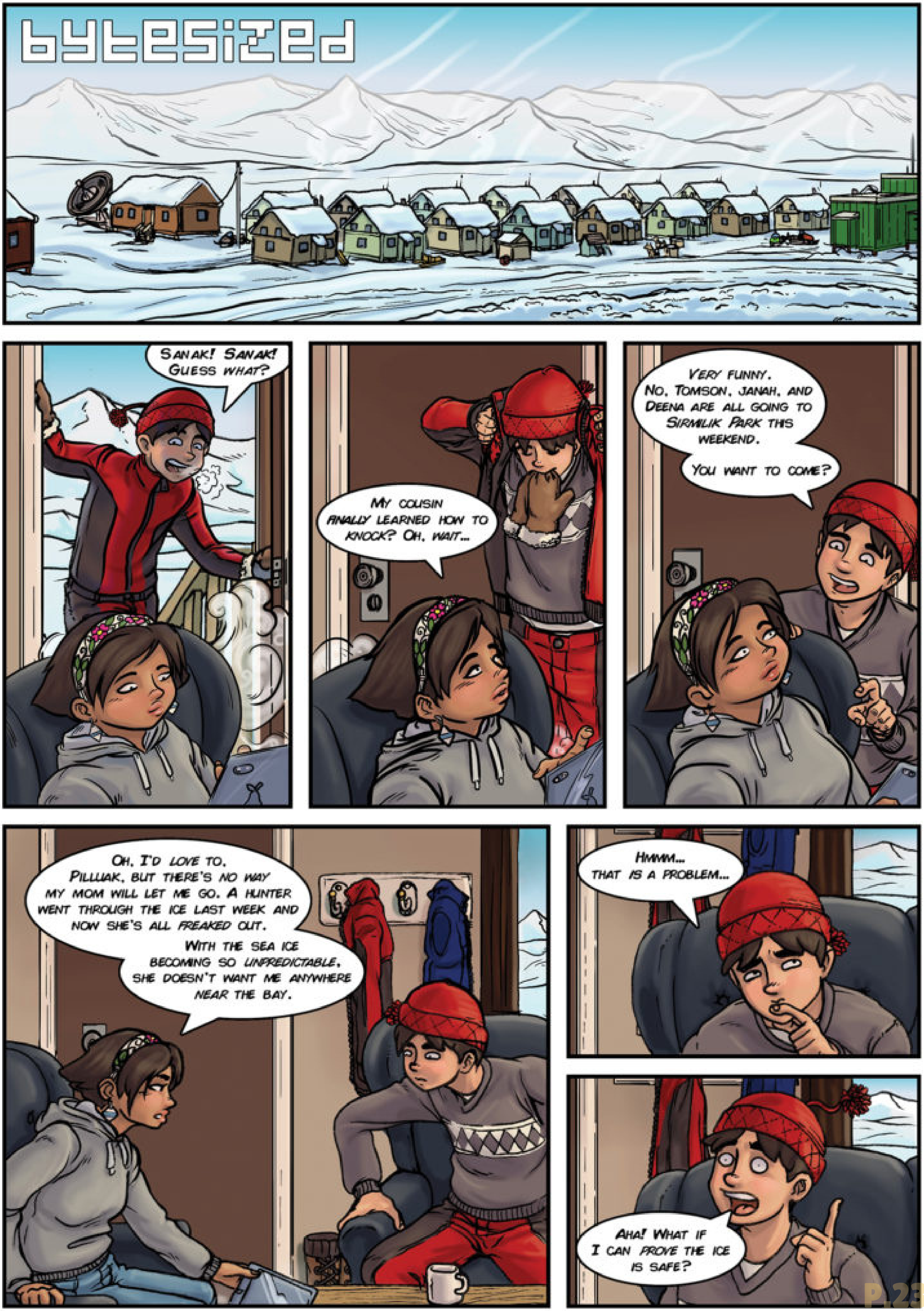
How is the climate changing?

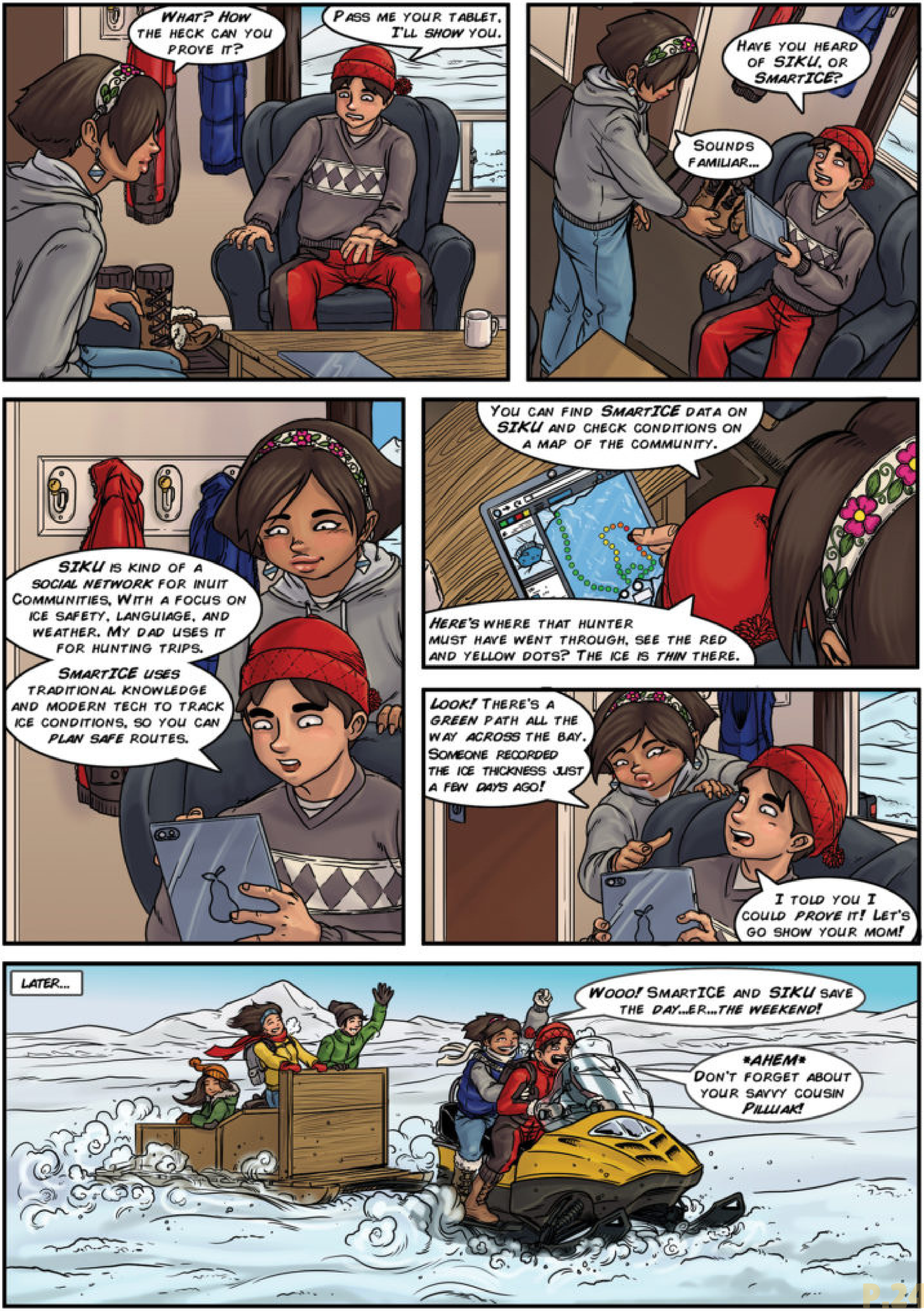
How does this affect the land?

Siku and **SmartIce** are apps that use the data we collect to inform our community about the environment. We can take our local climate measurements, such as temperature and sea ice thickness at different locations. Not only does this information help us to stay safe, over time, but it also shows us patterns and what changes are happening.

Please note the micro:bit does not work at temperatures below -25°C. Cold temperatures can break it!

Bytesized was created by Ian MacLean. Ian is a professional pixel artist/animator and illustrator who has worked in the gaming industry for 16 years. The comic can also be found on pages 28 and 29 of the Root and Stem publiication, Issue One.


bytesized
Sanak! Sanak! Guess what?
My cousin finally learned how to knock? Oh, wait...
Very funny. No, Tomson, Janah, and Deena are all going to Sirmilik Park this weekend.
You want to come?
Oh, I'd love to, Pilluak, but there's no way my mom will let me go. A hunter went through the ice last week and now she's all freaked out.
With the sea ice becoming so unpredictable, she doesn't want me anywhere near the bay.
Hmmm... that is a problem...
Aha! What if I can prove the ice is safe?

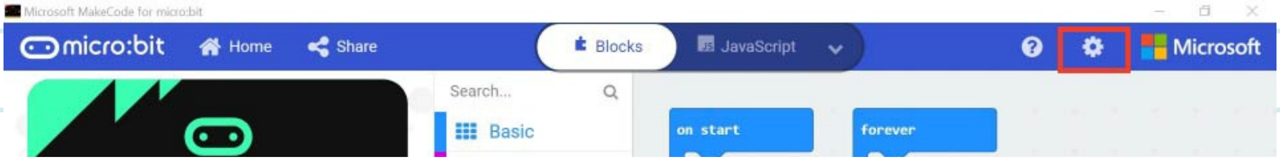

WHAT? HOW THE HECK CAN YOU PROVE IT?
PASS ME YOUR TABLET, I'LL SHOW YOU.
HAVE YOU HEARD OF SIKU, OR SMARTICE?
SOUNDS FAMILIAR...
SIKU IS KIND OF A SOCIAL NETWORK FOR INUIT COMMUNITIES, WITH A FOCUS ON ICE SAFETY, LANGUIAGE, AND WEATHER. MY DAD USES IT FOR HUNTING TRIPS.
SMARTICE USES TRADITIONAL KNOWLEDGE AND MODERN TECH TO TRACK ICE CONDITIONS, SO YOU CAN PLAN SAFE ROUTES.
YOU CAN FIND SMARTICE DATA ON SIKU AND CHECK CONDITIONS ON A MAP OF THE COMMUNITY.
HERE'S WHERE THAT HUNTER MUST HAVE WENT THROUGH, SEE THE RED AND YELLOW DOTS? THE ICE IS THIN THERE.
LOOK! THERE'S A GREEN PATH ALL THE WAY ACROSS THE BAY. SOMEONE RECORDED THE ICE THICKNESS JUST A FEW DAYS AGO!
I TOLD YOU I COULD PROVE IT! LET'S GO SHOW YOUR MOM!
LATER...
WOOO! SMARTICE AND SIKU SAVE THE DAY...ER...THE WEEKEND!
AHEM DON'T FORGET ABOUT YOUR SAVVY COUSIN PILLUAK!

COMPUTER ACTIVITY

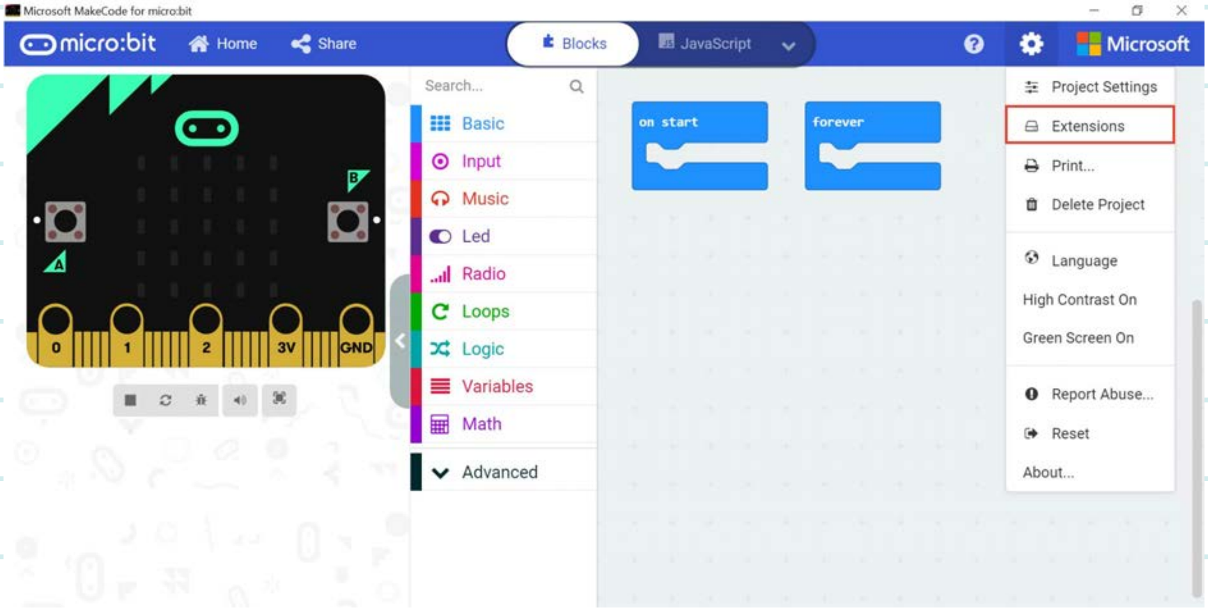
ADDING THE ENVIRO:BIT EXTENSION

! For this activity, we will need to add the enviro:bit extension to our MakeCode Editor. Open up the MakeCode editor and create a new project.

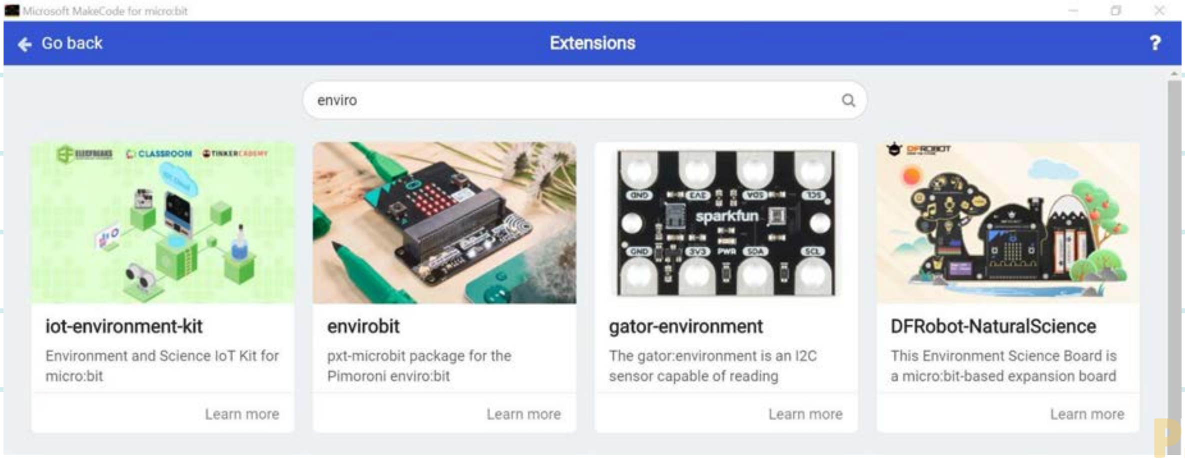
1 Click the settings button in the top right corner next to the Microsoft logo.



2 Select the second option “Extensions”



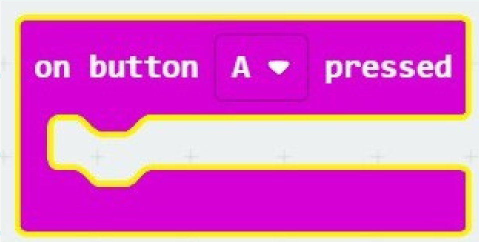
3 Type “enviro” into the search bar at the top of the screen. And select the “envirobit” extension.


Microsoft MakeCode for micro:bit
Go back
Extensions
?
enviro
iot-environment-kit
Environment and Science IoT Kit for micro:bit
Learn more
envirobit
pxt-microbit package for the Pimoroni enviro:bit
Learn more
gator-environment
The gator:environment is an I2C sensor capable of reading
Learn more
DFRobot-NaturalScience
This Environment Science Board is a micro:bit-based expansion board
Learn more

CODING THE ENVIRO:BIT

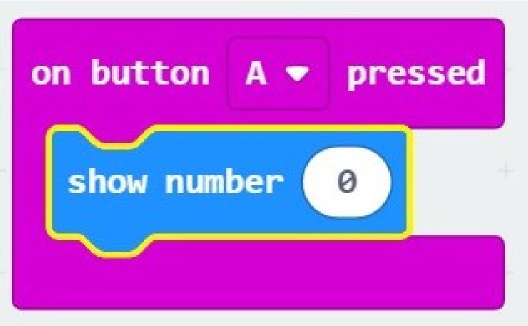
! Let’s add the code to make our enviro:bit take temperature readings. We want the enviro:bit to take a reading when we press the A button.

1 Go to the code library on the left and select the Input tab. Select the purple “on A button pressed” block and bring it to the code area.



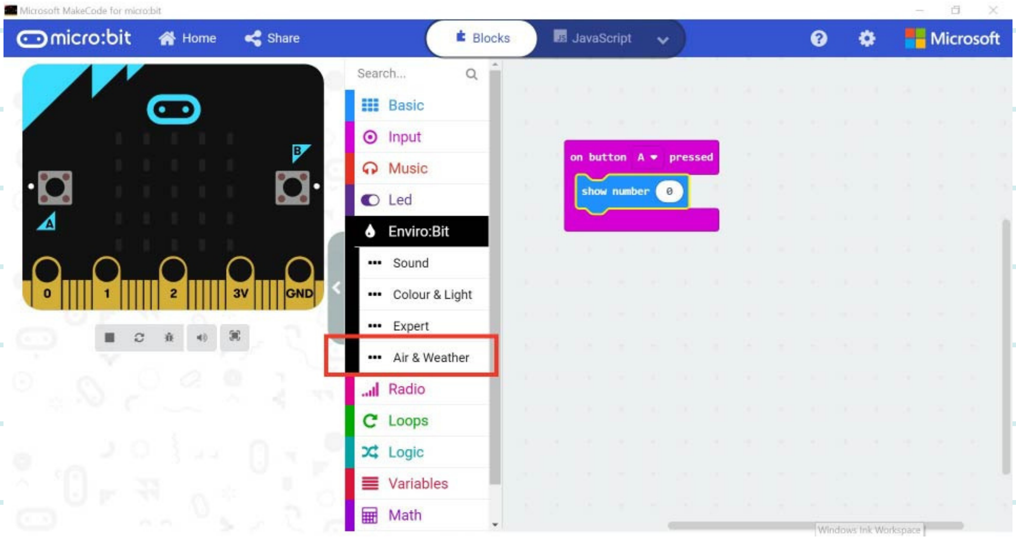
```
on button A ▾ pressed
```

2 In the “Basics” tab, select the blue “show number 0” block and drag it into the first block. When you press the A button on the micro:bit simulator, it will show the number 0 on the display.



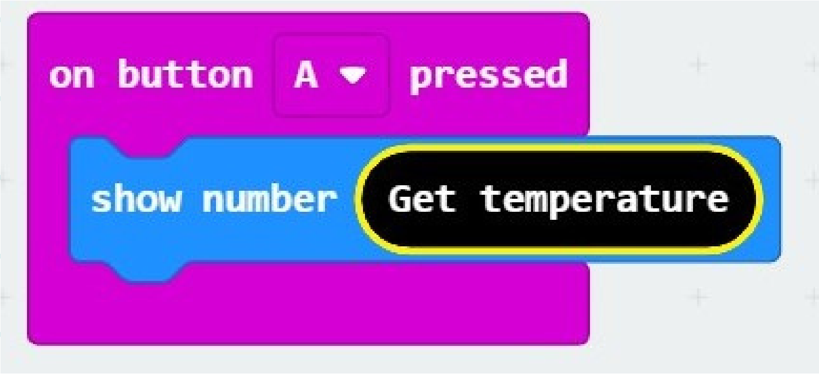
```
on button A ▾ pressed
    show number 0
```

3 When you click on the black envirobit tab, there are four options: Sound, Colour & Light, Expert, and Air & Weather. Select “Air & Weather”.



```
on button A ▾ pressed
    show number 0
```

4. Drag the black “Get temperature” bubble to the blue “show number 0” block.

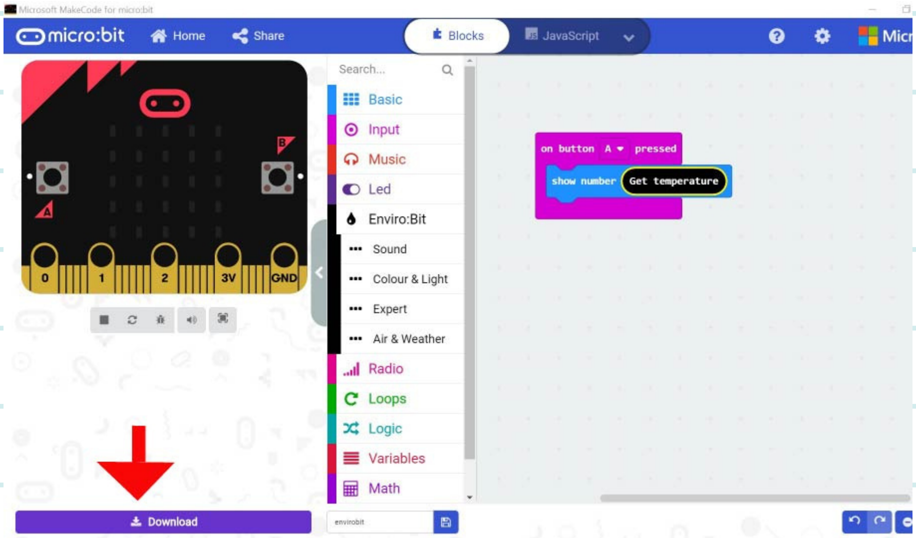


```
on button A ▾ pressed
    show number (Get temperature)
```

DOWNLOAD YOUR CODE

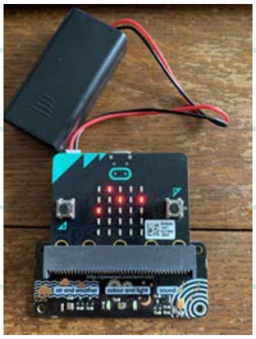
Now you can plug in your micro:bit to your computer using the USB cord and download the code to take temperature readings.

1. First, make sure its plugged into the computer, next you can click the “Download” button on the bottom left of the MakeCode editor.



```
on button A ▾ pressed
    show number (Get temperature)
```

2. A window will appear. Click the green download button. Now you can unplug your micro:bit from your computer. Plug in the battery pack, snap on the enviro:bit and take some temperature readings!



05 MEASURING AIR PRESSURE WITH THE ENVIROBIT

OVERVIEW

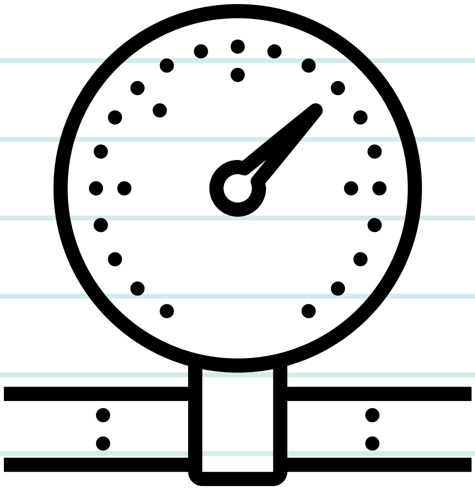
By using the enviro:bit addon with a micro:bit, you can gain access to a number of sensors that the micro:bit itself does not have. One such sensor measures the air pressure (or atmospheric pressure). We will be using Microsoft MakeCode to create a program that will display the current air pressure on the Micro:bit's display.

HOW DOES IT WORK?

By using our Qanuqtururangniq skills, we will measure air pressure. Meteorologists measure atmospheric pressure because it can help us predict the weather. We can measure Low-Pressure Systems when often we'll see cloudiness, wind and even rain! When we have High-Pressure Systems, we can expect to have dry, clear weather.

you will need:

- enviro:bit
- Ziploc/other sealable bag

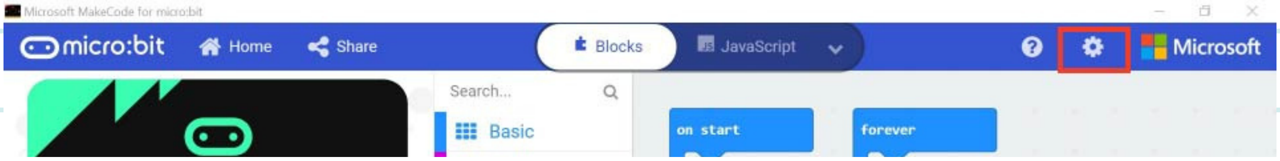


COMPUTER ACTIVITY

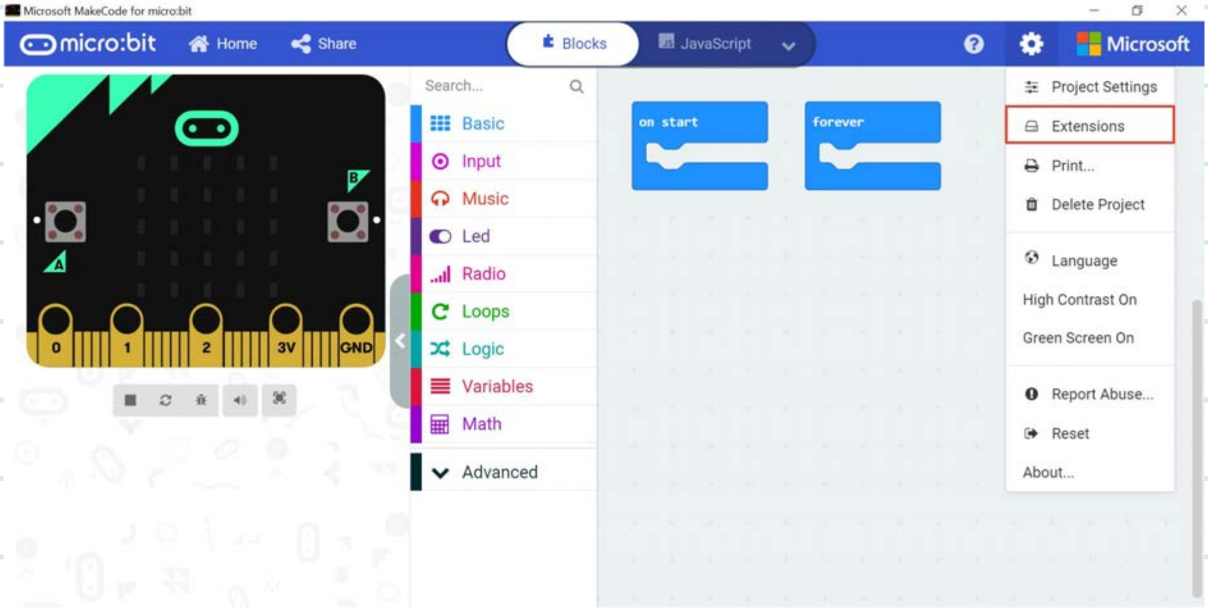
ADDING THE ENVIRO:BIT EXTENSION

! For this activity, we will need to add the enviro:bit extension to our MakeCode Editor. Open up the MakeCode editor and create a new project.

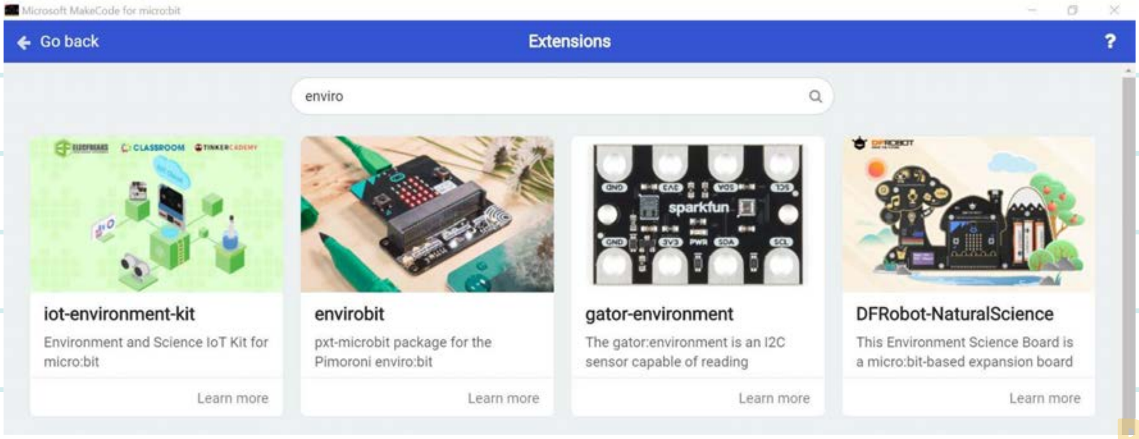
1 Click the settings button in the top right corner next to the Microsoft logo.



2 Select the second option “Extensions”



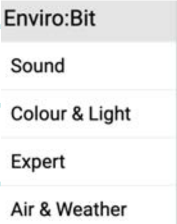
3 Type “enviro” into the search bar at the top of the screen. And select the “envirobit” card.


Microsoft MakeCode for micro:bit
Go back
Extensions
?
enviro
iot-environment-kit
Environment and Science IoT Kit for micro:bit
Learn more
envirobit
pxt-microbit package for the Pimoroni enviro:bit
Learn more
gator-environment
The gator:environment is an I2C sensor capable of reading
Learn more
DFRobot-NaturalScience
This Environment Science Board is a micro:bit-based expansion board
Learn more

05 DISPLAYING AIR PRESSURE USING THE MICRO:BIT'S LED DISPLAY

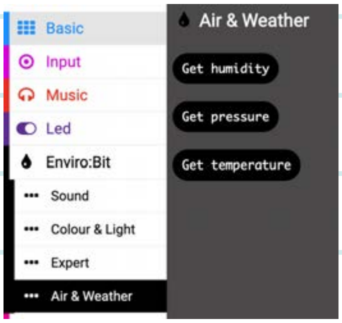
! To display the air pressure that's being measured by the enviro:bit, we need to combine the "Get pressure" block from the Enviro:Bit category with code that will display text on the micro:bit.

1 Click on the "Enviro:Bit" category. This will reveal the sensor categories on the Enviro:bit.

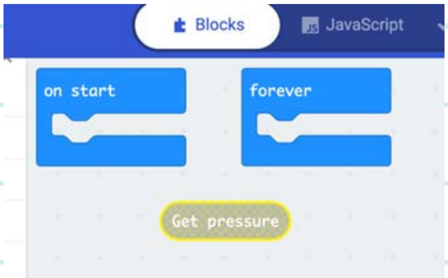


2 To read the pressure off of the sensor on the enviro:bit, you will need to use some of the blocks that can be found in the enviro:bit's "Air & Weather" category.

3 Click on "Air & Weather". This will reveal three blocks to do with measuring the environment around the enviro:bit.



4 The block that reads the air pressure is the "Get pressure" block. Drag a "Get pressure" block into the code area.



05

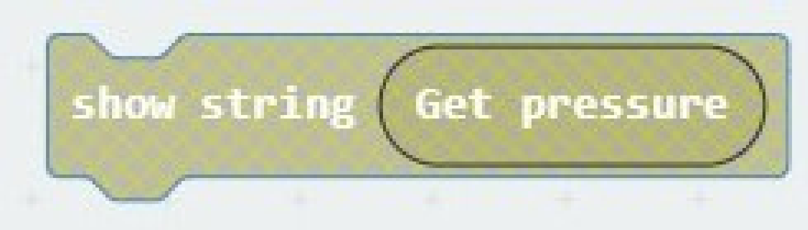
DISPLAYING AIR PRESSURE USING THE MICRO:BIT'S LED DISPLAY

! To display the air pressure that's being measured by the enviro:bit, we need to combine the "Get pressure" block with code that will display text on the micro:bit. You can find a block that does this in the "Basic" category.

1 Click on the "Basic" category. This will reveal a number of blocks that do simple or basic things on the micro:bit.

2 The block we're looking for is called "Show string" which displays a string—programmer talk for "text"—by scrolling it across the micro:bit's LED display. We can also display numbers this way. Drag a "Show string" block into the code area.

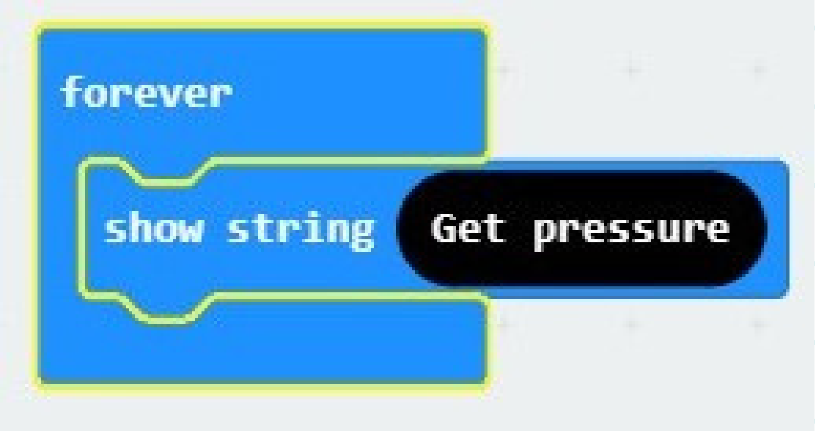
3 Did you notice that the "Show string" block has a bubble that's about the same shape as the "Get pressure" block? This is MakeCode telling us that we can put the "Get pressure" block into that space. Click and drag the "Get pressure" block in the code area into the empty bubble in the "Show string" block. If you're successful, it should look as below.



```
show string Get pressure
```

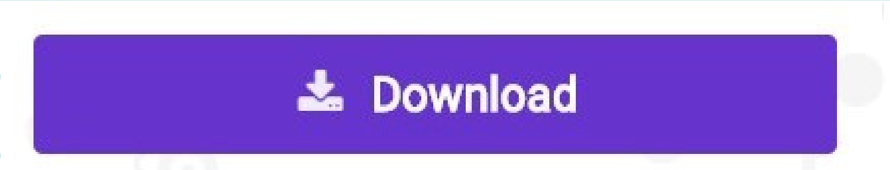
4 Click and drag the "Show string" block (now carrying the "Get pressure" block as well) into the "Forever" block. It should snap into place and should no longer have lines crossing through it. If you're successful, it should look llike the picture below.

5 Click and drag the “Show string” block (now carrying the “Get pressure” block as well) into the “Forever” block. It should snap into place and should no longer have lines crossing through it. If you’re successful, it should look like the picture below.



```
forever
  show string Get pressure
```

6 Download your program onto your micro:bit and connect your micro:bit to the enviro:bit.

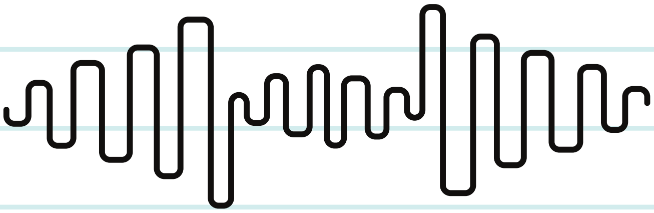


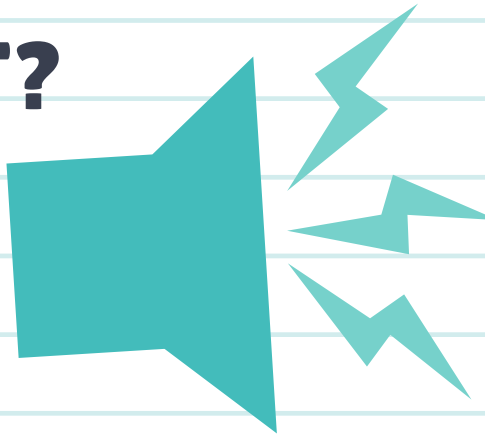
7 Connect the battery pack to the micro:bit and try measuring the air pressure where you are.
What is happening now on your micro:bit? Does a number scroll across the LED display? Is it always the same number, or does the number change? This number is the air pressure being read from the enviro:bit by your program and being displayed, as instructed, on the micro:bit’s LED display.
Try putting the micro:bit into a clear, sealable plastic bag and squeeze the bag gently (we don't want the bag to pop!) What is happening now?

Congratulations! You now have a device that can measure the current air pressure and tell you what it is!

06

HOW NOISY IS IT?





OVERVIEW

Learn how to measure the noise in your environment. Think about how noise can affect your health and wellness and the health of the animals and fish around you.

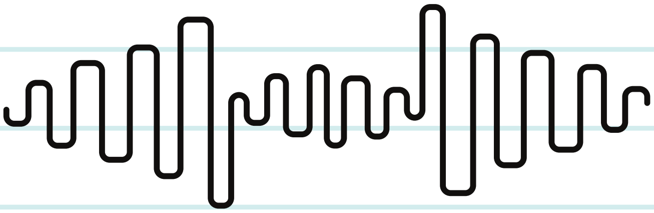
HOW DOES IT WORK?

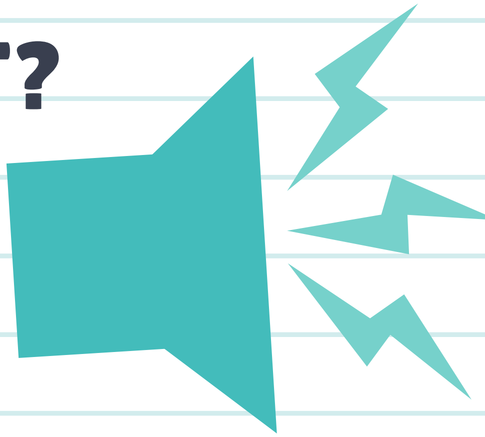
Our environment can have all sorts of sounds, especially in busy urban areas. Animals, like us, make sounds to communicate with each other. However, they can be disrupted by loud noises that are not part of their natural environment. This is a problem known as Noise Pollution. Marine mammals are especially affected by noise pollution because they use echolocation to communicate, navigate the ocean, and find food.

PILIMMAKSARNIQ AND QANUQTURURANGNIQ

To be stewards of the environment, we need to learn skills that help us understand this role. Our elders teach us that each person has an important job to learn skills so we can contribute as a community. Qanuqtururangniq is the ability to think deeply and use our knowledge that we learn from observation. We will use this teaching to observe changes in our environment. Using the micro:bit as a tool to help us measure the information, we will use our skills of observation to understand it.

HOW NOISY IS IT?





VOCABULARY

Echolocation -- A technique used by certain animals to find their way in dark environments. They emit a sound that is reflected back at them.

Qanuqtururangniq -- the ability to think deeply and use observation skills to understand our environment.

Sound sensor -- the Enviro:bit has a sound sensor built-in. You can measure the level of sound around you or make something happen with sound.

Noise -- noises are unwanted sounds that are disruptive to life.

Sound -- sounds are what we hear.

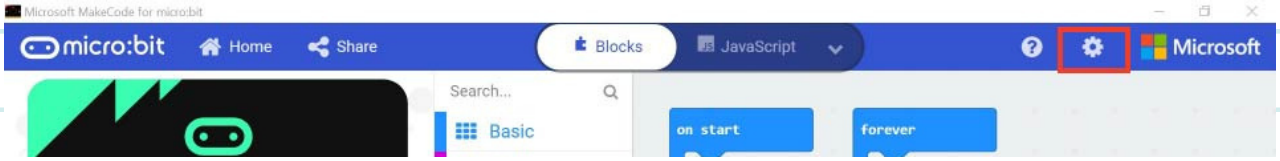
You will need an enviro:bit for this activity.

COMPUTER ACTIVITY

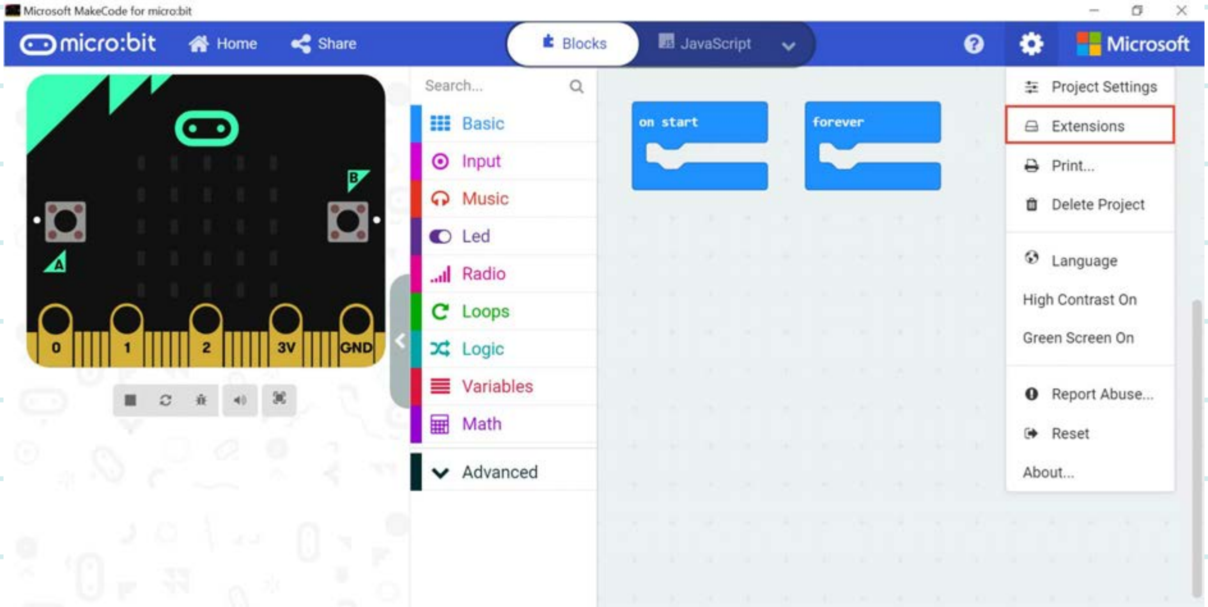
ADDING THE ENVIRO:BIT EXTENSION

! For this activity, we will need to add the enviro:bit extension to our MakeCode Editor. Open up the MakeCode editor and create a new project.

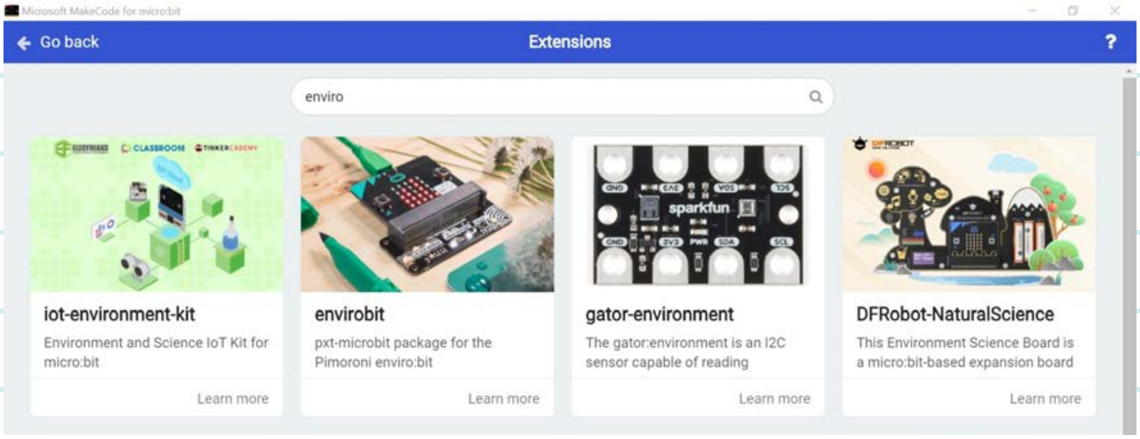
1 Click the settings button in the top right corner next to the Microsoft logo.



2 Select the second option “Extensions”



3 Type “enviro” into the search bar at the top of the screen. And select the “envirobit” card.



iot-environment-kit
Environment and Science IoT Kit for micro:bit
Learn more

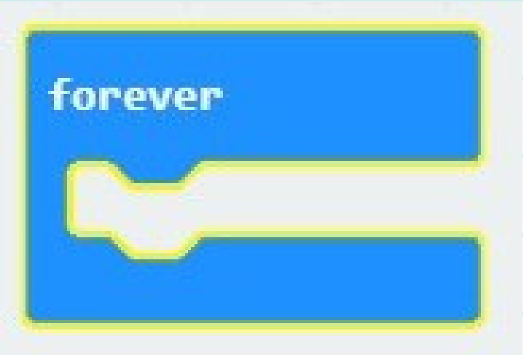
envirobit
pxt-microbit package for the Pimoroni enviro:bit
Learn more

gator-environment
The gator:environment is an I2C sensor capable of reading
Learn more

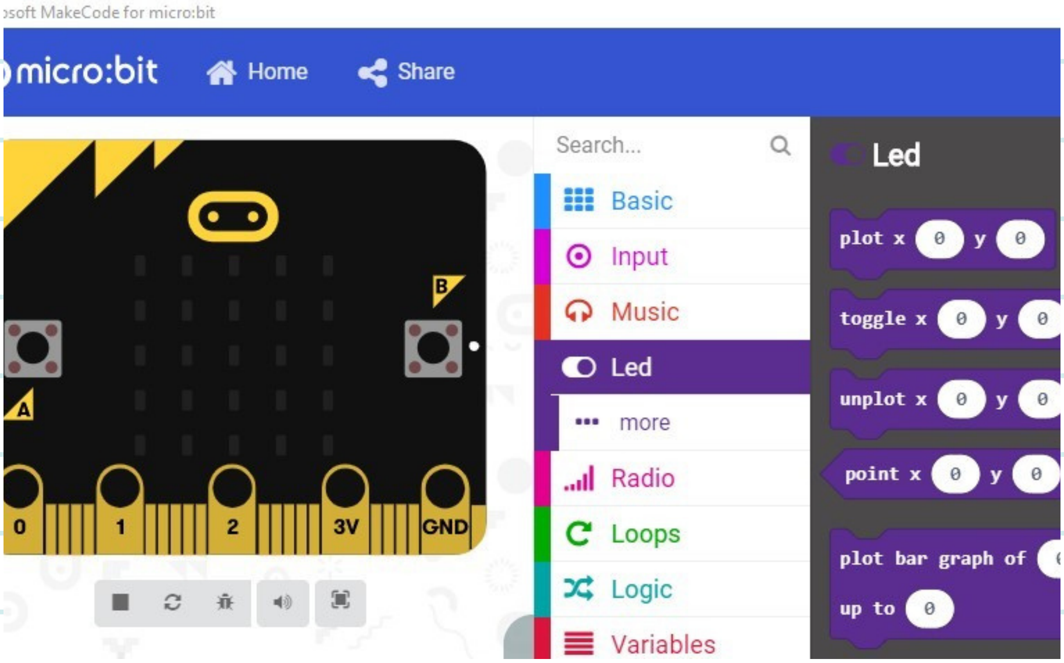
DFRobot-NaturalScience
This Environment Science Board is a micro:bit-based expansion board
Learn more

CODING THE NOISE SENSOR

1 Drag the Forever block into the middle of the code area. Computers only do what you tell them to do. This block makes the code run all the time when your micro:bit is connected to power through your computer or through the battery pack.

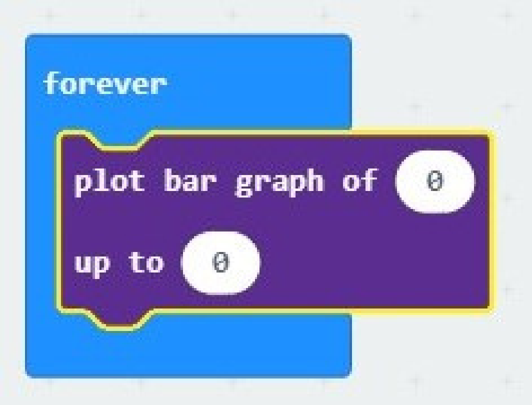


2 From the LED tab, get the plot bar graph of 0 up to 0 block. This block turns the LED display on the micro:bit into a graph display!

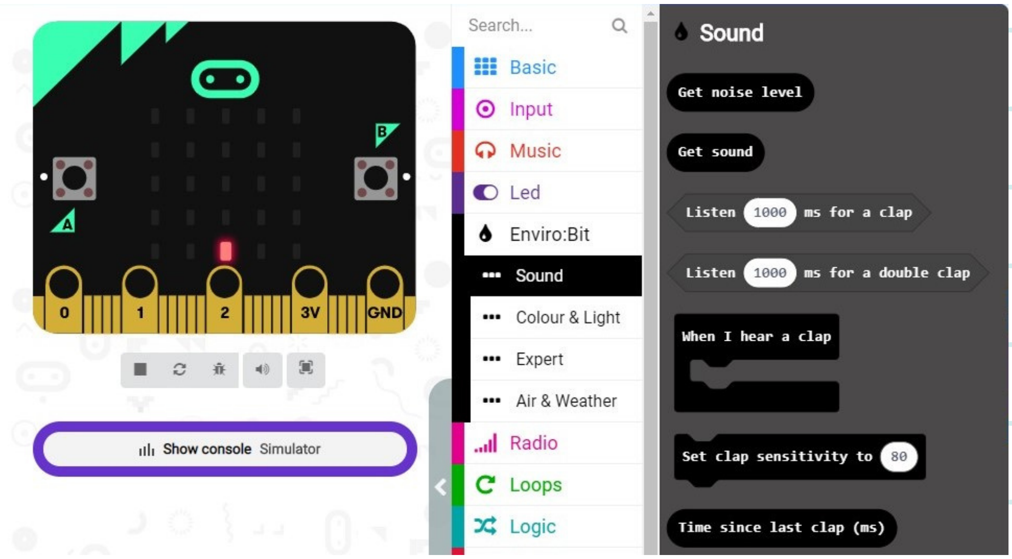


CODING THE NOISE SENSOR

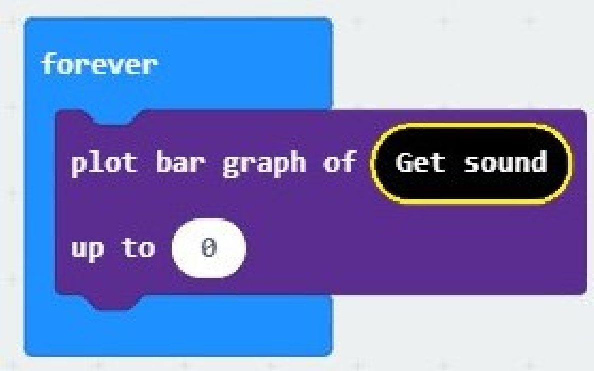
3 Drag the block into the middle of the Forever block.



4 From the Enviro:bit tab open the Sound tab.

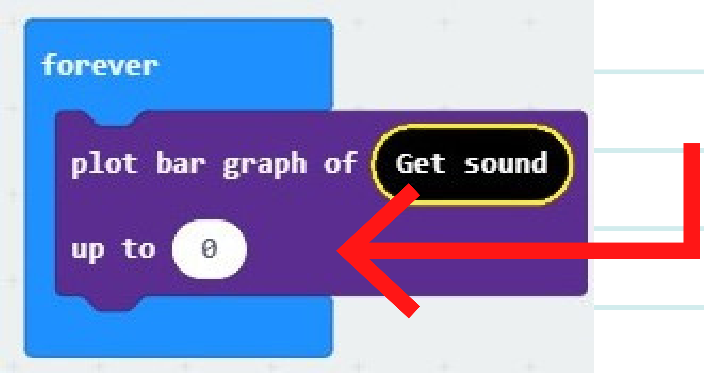


5 Drag the Get sound block into the first opening in the plot bar graph block.



CODING THE NOISE SENSOR

6 Leave the value set to 0 and the micro:bit will automatically choose the best scale to fit on the display screen.



```
forever
    plot bar graph of Get sound
    up to 0
```

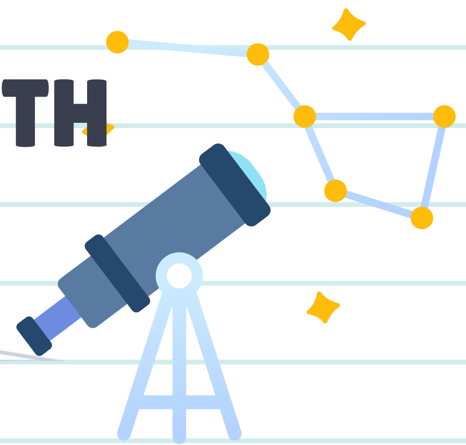
7 Connect your Enviro:bit to your Micro:bit by snapping the Enviro:bit onto the bottom edge of the Micro:bit.

8 Connect your micro:bit to your computer with the USB cord from your kit. Download (flash) your code onto your micro:bit.

9 Test your code by making sounds and see if the Enviro:bit measures the noise level and shows the graph on the Micro:bit display!

10 You can disconnect the micro:bit from your computer, plug in your battery pack, and go measure some noise levels! How noisy is your environment?

COMPASS NORTH CHALLENGE

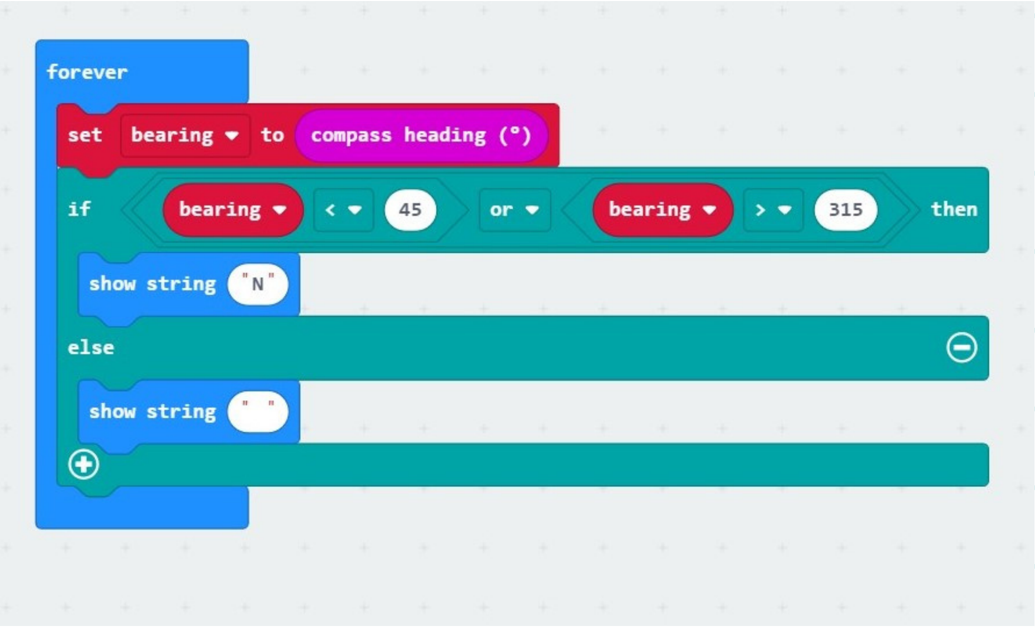


OVERVIEW

Here’s an extra activity to try once you’ve finished the first compass activity! The level of difficulty is intermediate if you would like to get into some more advanced coding. We will use the micro:bit’s magnetometer readings to create a compass that shows the cardinal directions.

HOW DOES IT WORK?

The compass that you will create will use forever loops, variables, comparative statements, and If else statements. We will be analyzing this code to make the compass.

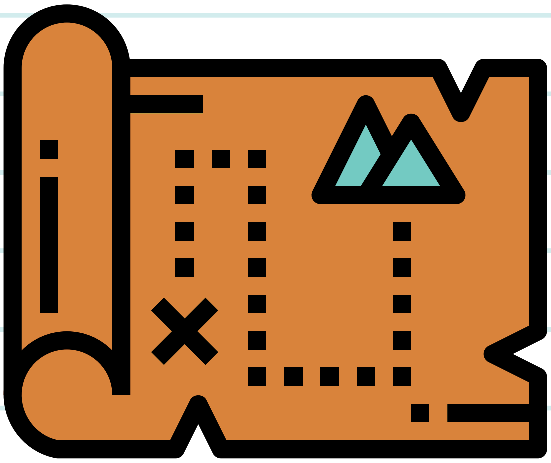


```
forever
    set bearing ▾ to compass heading (°)
    if bearing ▾ < ▾ 45 or ▾ bearing ▾ > ▾ 315 then
        show string "N"
    else
        show string " "
```

Here's the starter code! Copy the code or follow the directions below to import it into the MakeCode Editor from the USB key provided with your kit.

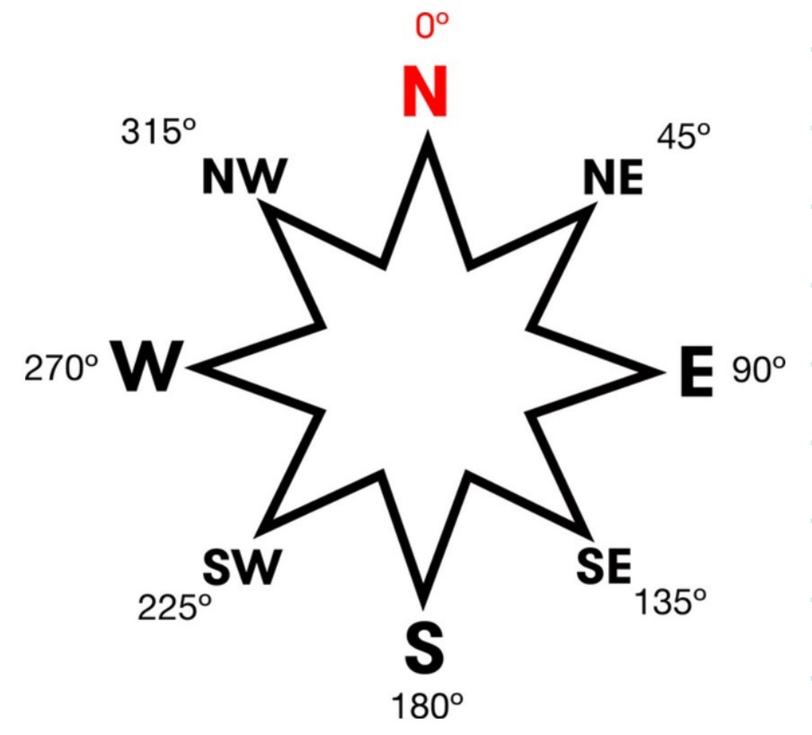
VOCABULARY

- **Compass** -- A tool for finding direction
- **Magnetometer**-- an instrument for measuring the magnitude and direction of a magnetic field
- **Compass Heading**-- The heading is another word for the direction the compass is pointing towards and is measured clockwise from north.



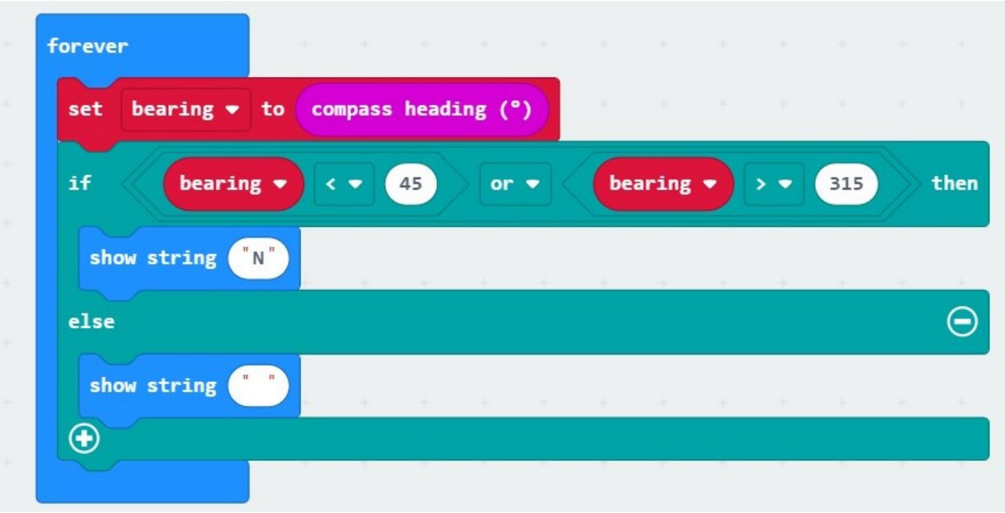
CARDINAL POINTS DIAGRAM

This diagram will help us to understand how to translate the cardinal points into numerical measurements that the micro:bit will understand. A circle is measured in degrees. A full circle is 360 degrees, with 0 degrees at the very top.



UPLOADING THE STARTER CODE

1. Open the MakeCode editor and follow the steps below to import the starter code or copy the code manually.

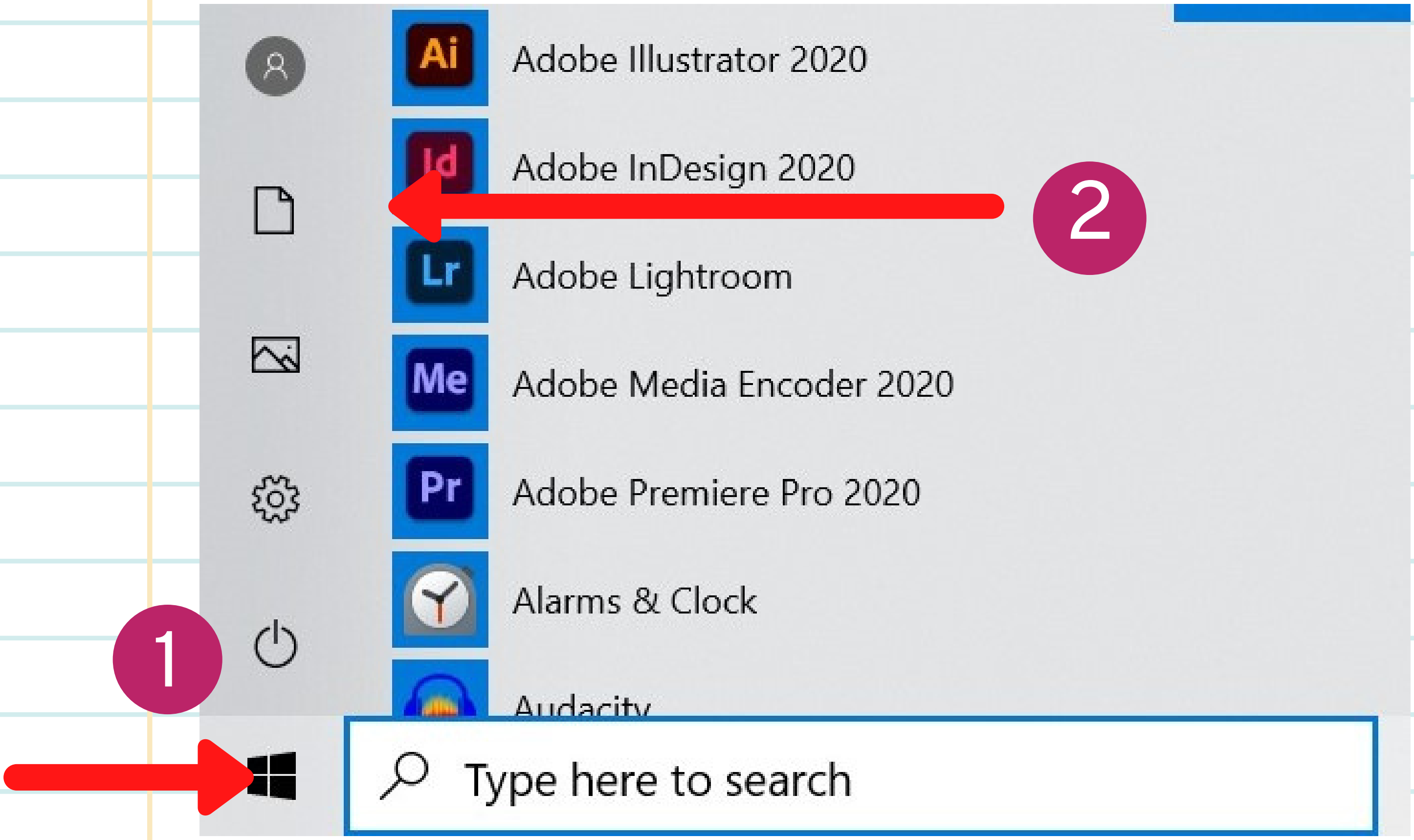


```
forever
  set bearing ▾ to compass heading (°)
  if bearing ▾ < ▾ 45 or ▾ bearing ▾ > ▾ 315 then
    show string "N"
  else
    show string " "
```

UPLOADING THE STARTER CODE

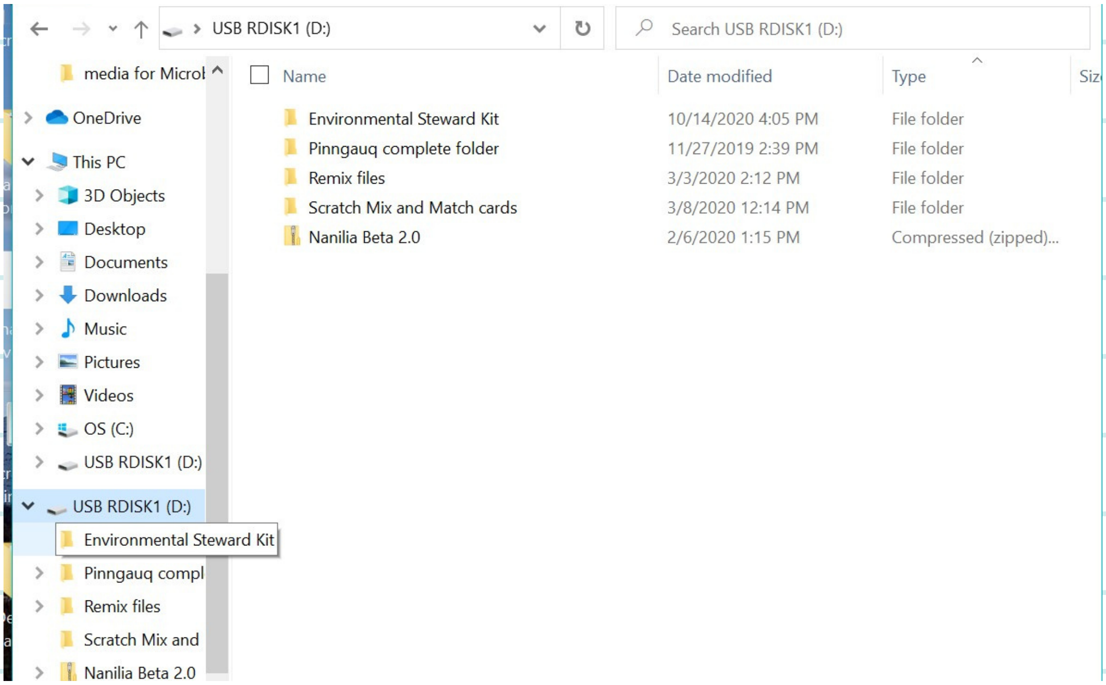
1. On your computer, click the windows icon at the bottom left of the screen. This is known as the "Start" button.

2. Click on the "Documents" icon on the left above the start button.



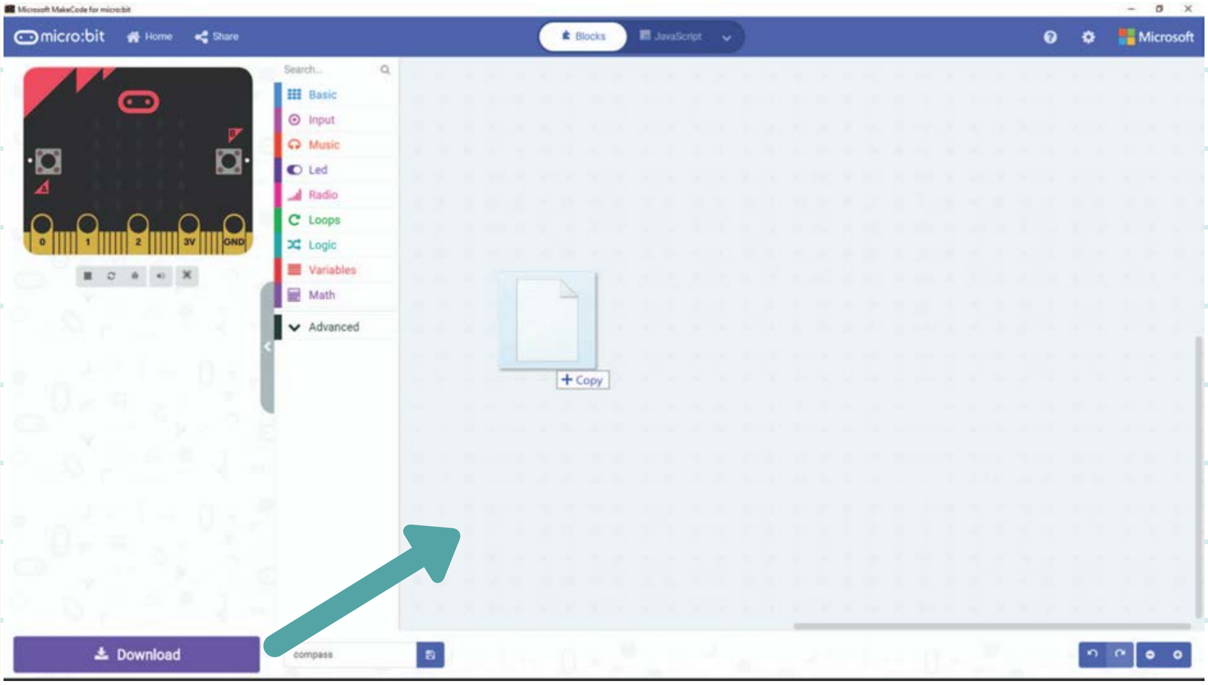
UPLOADING THE STARTER CODE

3 A window will appear. Find the location of the USB drive.



Name	Date modified	Type	Siz
Environmental Steward Kit	10/14/2020 4:05 PM	File folder	
Pinngauq complete folder	11/27/2019 2:39 PM	File folder	
Remix files	3/3/2020 2:12 PM	File folder	
Scratch Mix and Match cards	3/8/2020 12:14 PM	File folder	
Nanilia Beta 2.0	2/6/2020 1:15 PM	Compressed (zipped)...	

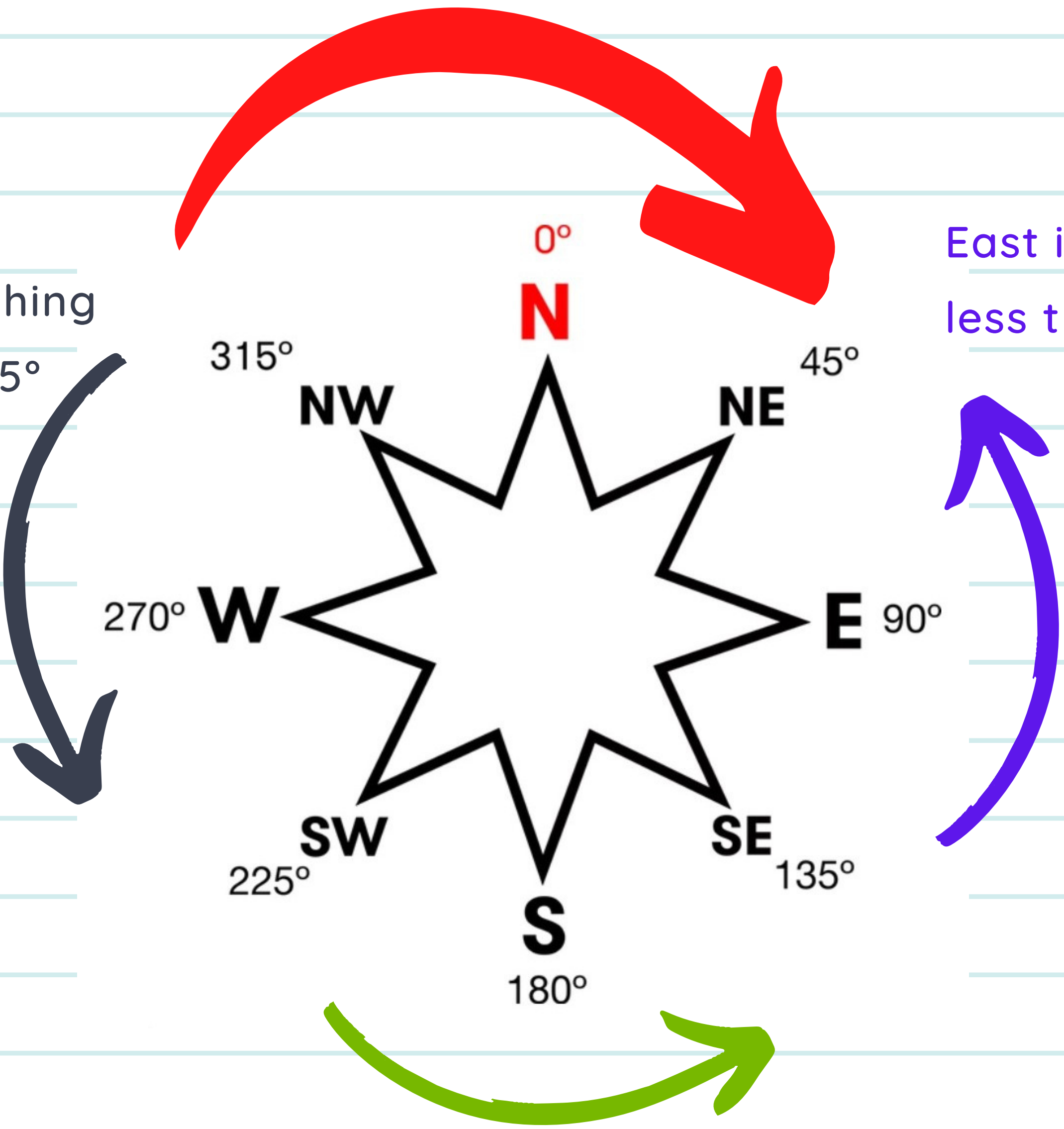
4 Click on the folder to open it. Select the "microbit-compass.hex" file and drag it into the MakeCode code area.



CARDINAL POINTS DIAGRAM

This Diagram will help us to understand how to translate the cardinal points into numerical measurements that the micro:bit will understand.

North can be described between 315° and 45°



West is anything less than 315°

East is anything less than 135°

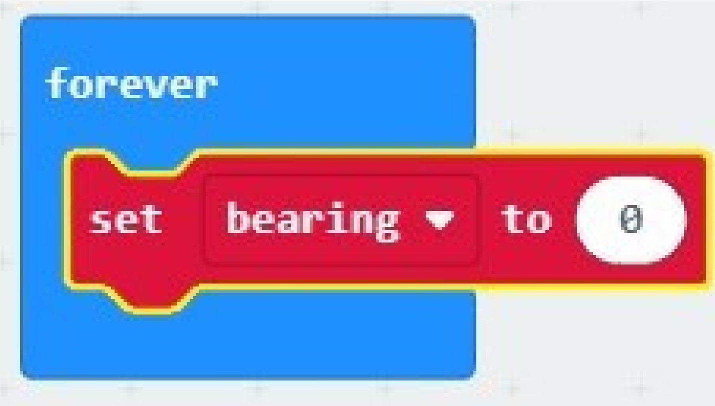
South is anything less than 225°

BREAKING DOWN THE CODE

1. The forever loop tells the micro:bit to keep taking readings of the compass heading in degrees.

   a. The compass heading input block continuously takes measurements from the magnetometer and inputs that information in degrees (°).

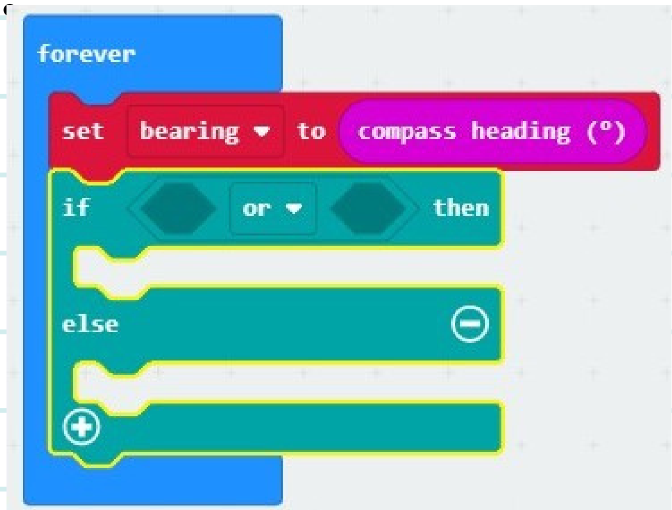
   b. The Variable block defines the “bearing” as the compass heading measurement. We use a variable block because this measurement changes as the compass moves and changes direction.



```
forever
  set bearing ▾ to 0
```

   a. The green block is an “If, then, else” statement block.

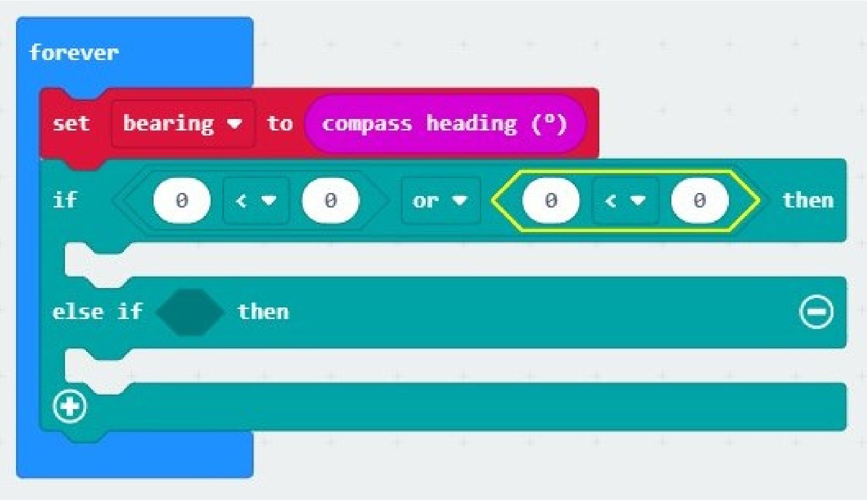
      i. Within the If-Else statement block are Boolean blocks. These blocks generate an input if one of the inputs prove to be true. For example: The micro:bit will show “N” for north if the numerical compass bearing is less than 45° or greater than 315°



```
forever
  set bearing ▾ to compass heading (°)
  if ⬢ or ▾ ⬢ then

  else ⊖

  ⊕
```

BREAKING DOWN THE CODE

i. Within the OR block are the comparative blocks. These use mathematical operations to compare two inputs. In this case, we are using greater-than or less-than operators to compare to the compass heading reading of the micro:bit.

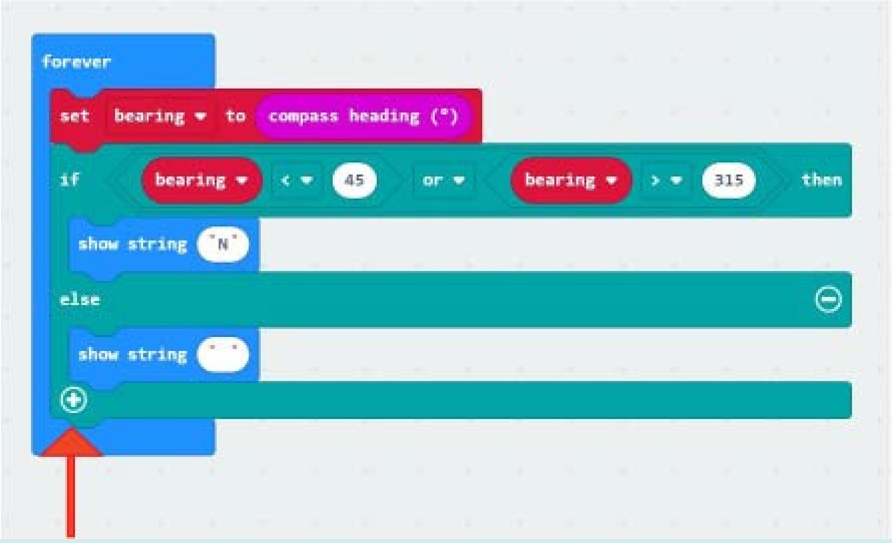


```
forever
  set bearing to compass heading (°)
  if 0 < 0 or 0 < 0 then
  else if then
```

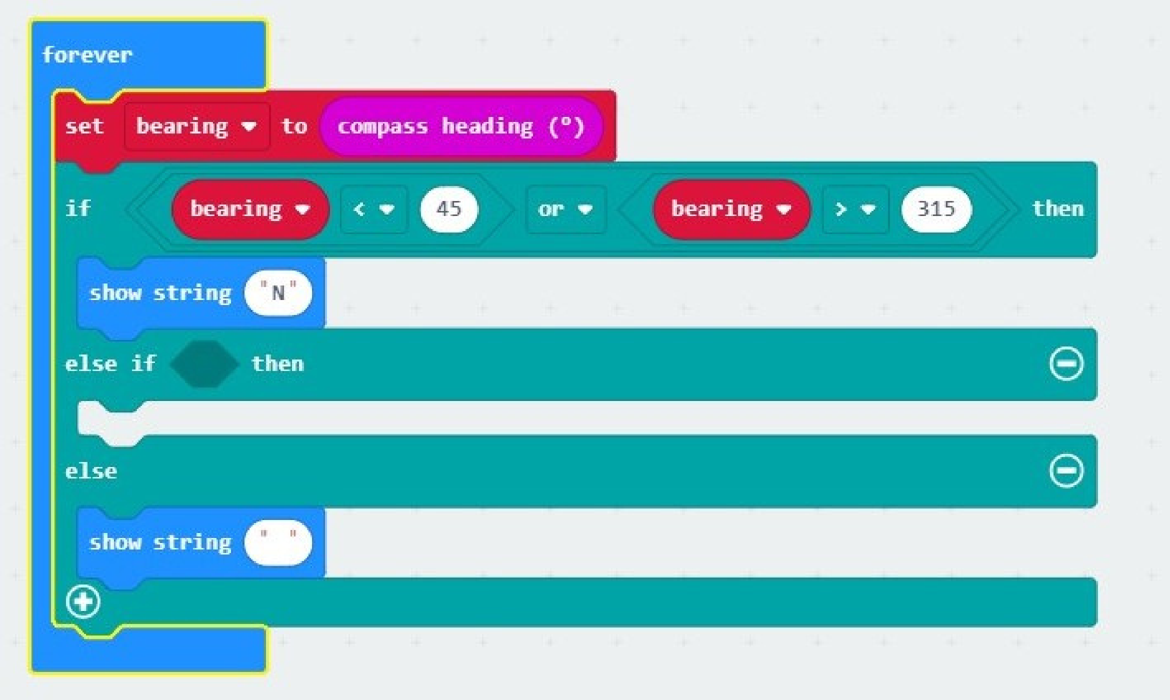
LET'S CODE!

Now we will add the code to the other cardinal points East, South, and West, using the above code and the Cardinal points diagram.

1. Click the + symbol at the bottom of the if-else statement block to add a new else-if statement.

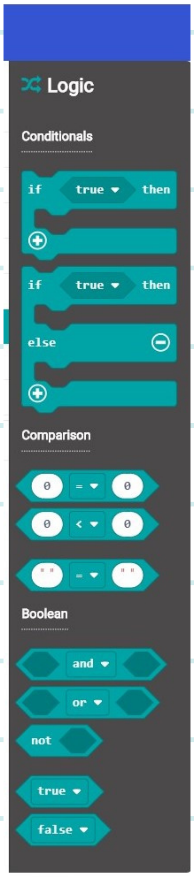


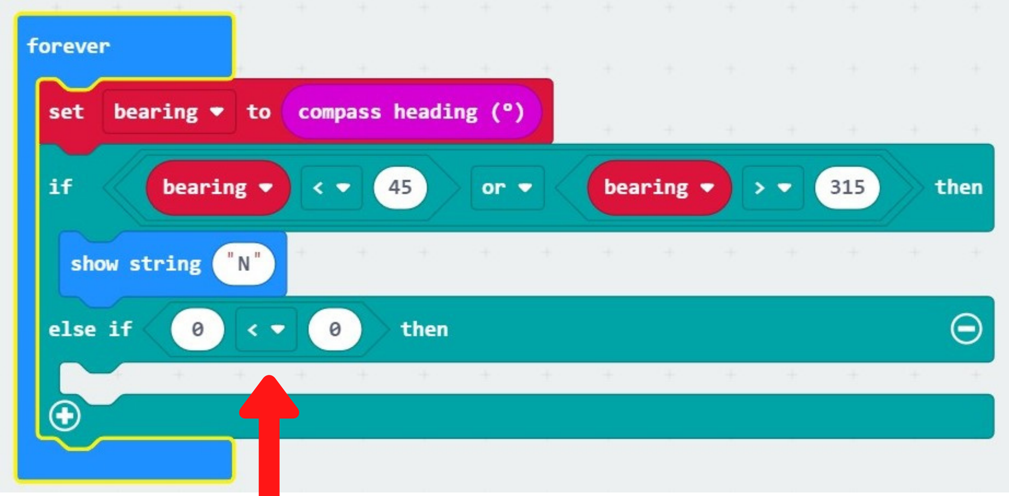
```
forever
  set bearing to compass heading (°)
  if bearing < 45 or bearing > 315 then
    show string "N"
  else
    show string ""
```



```
forever
  set bearing ▾ to compass heading (°)
  if bearing ▾ < ▾ 45 or ▾ bearing ▾ > ▾ 315 then
    show string "N"
  else if ⬡ then
  else
    show string " "
  ⊕
```

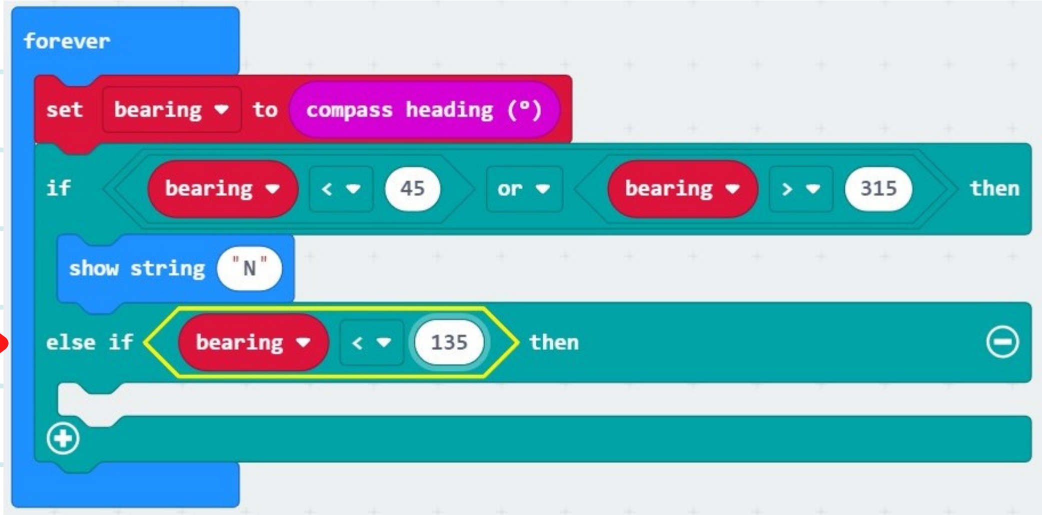
2. Notice the space in the new else-if statement. This is where we will add the comparative blocks. In the code library, select “Logic” and under the Boolean section, Select the less- than block and place it in this space.





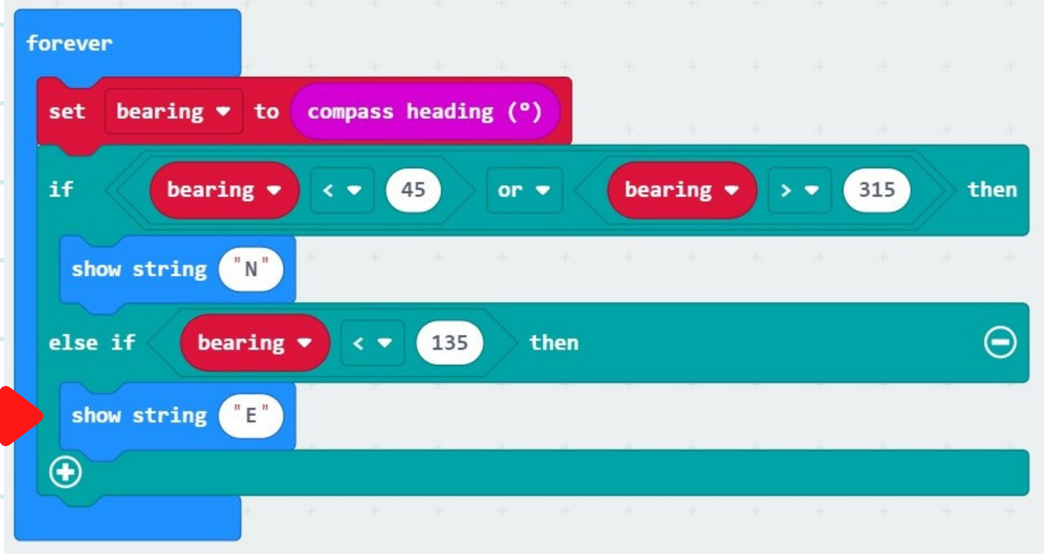
```
forever
  set bearing ▾ to compass heading (°)
  if bearing ▾ < ▾ 45 or ▾ bearing ▾ > ▾ 315 then
    show string "N"
  else if 0 < ▾ 0 then
  ⊕
```

3. Now that we have this statement in place, we can compare two inputs. The first input we want to compare is the compass bearing for East. When we look at the cardinal points diagram, East corresponds to a compass heading of 90° and a compass bearing less than 135°.



```
forever
    set bearing ▾ to compass heading (°)
    if bearing ▾ < ▾ 45 or ▾ bearing ▾ > ▾ 315 then
        show string "N"
    else if bearing ▾ < ▾ 135 then
```

4. Finally, if this statement is true, then is should show E for east. Use the “show string” block and place it below the code we just made. Enter the letter E.

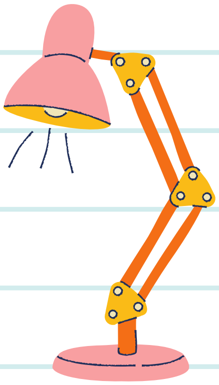


```
forever
    set bearing ▾ to compass heading (°)
    if bearing ▾ < ▾ 45 or ▾ bearing ▾ > ▾ 315 then
        show string "N"
    else if bearing ▾ < ▾ 135 then
        show string "E"
```

Code the rest of the Cardinal Points by following the Cardinal Points DIagram!

07

SIMPLE NIGHT LIGHT



OVERVIEW

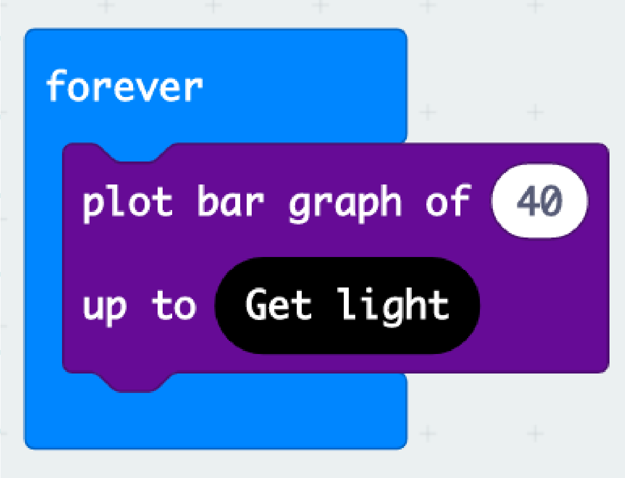
In this project, we'll use a light sensor in order to activate a night light. In dark environments, our light will get brighter. When the environment is bright, the light will dim.

BACKGROUND INFORMATION

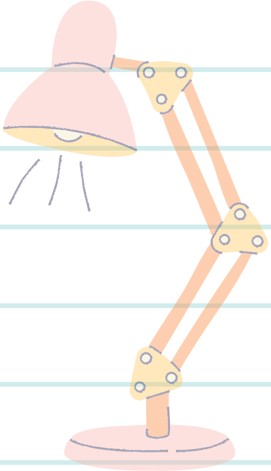
Many animals including humans have sensors in their bodies and systems that react to them. We can sense light, sound, temperature, touch, acceleration, taste, and smell. An example of a system in our bodies that responds to light is the pupils in our eyes. In bright light, our pupils get smaller to let less light into our eyes. In the dim light, our pupils become larger so that our eyes can receive more of the available light.

You will need an enviro:bit for this activity.

LET'S CODE!



```
forever
    plot bar graph of 40
    up to Get light
```



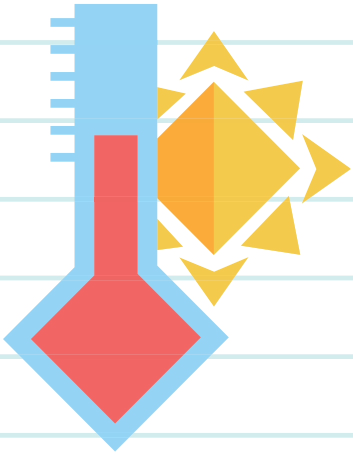
1. The ‘plot bar graph’ block uses the lights on the micro:bit screen to show a chart. We will place the ‘plot bar graph’ block inside the ‘forever’ block to make it continue graphing over and over while our program is running. You can experiment with adding different numbers here to see how the block works, but we’ll use ‘plot bar graph of 40’ for our project.

2. Click on the ‘extensions’ section and search for ‘enviro:bit’ to add the black blocks that use the extra features of our enviro:bit

3. We can use the ‘get light’ block from the ‘colour and light’ section of the micro:bit to get a number value from our enviro:bit to show how much light is reaching its light sensor. When the sensor gets no light, the number in place of the ‘get light’ block is 0, or when at maximum brightness. the value is 255. We’ll place this block after ‘up to’.

4. We’ve made a night light that will get brighter as the environment is darker, or turn off when the environment is really bright. Test it out by changing the light level reaching your micro:bit. Try holding it under a table for less light, or shining a flashlight on it for more light and see what happens!.

08 WEATHER STATION



OVERVIEW

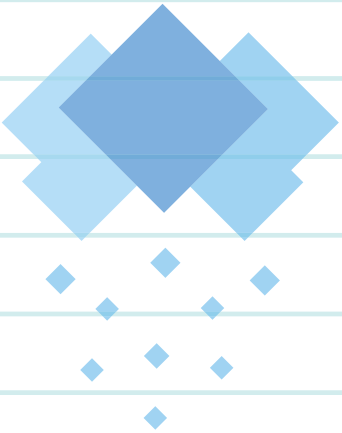
In this project, we'll combine all the weather sensors of the enviro:bit into one project.

BACKGROUND INFORMATION

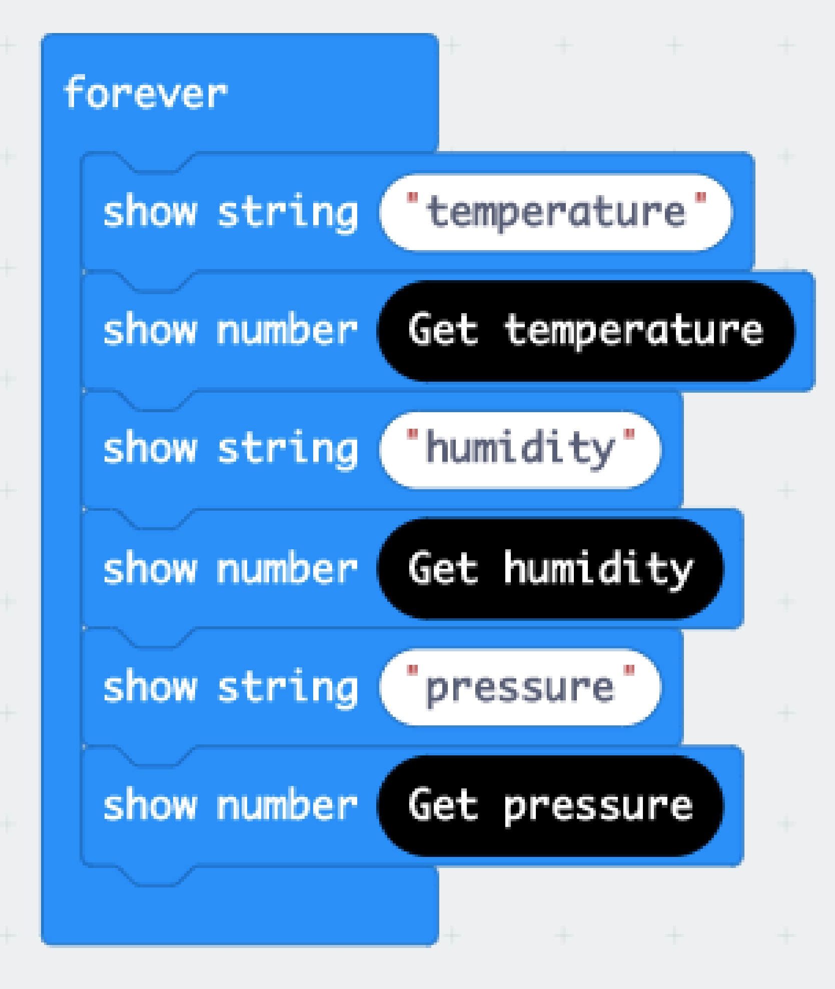
Observing our environment and measuring weather data, such as temperature, pressure and humidity can help us understand our environment. It can also really help to predict the weather and guide us towards making good decisions when planning travel, trips on the land, flights or even just deciding what clothes to wear for the day.

VOCABULARY

- **Humidity** -- The amount of moisture in the air.
- **Pressure** -- Air pressure is a measure of the weight of air above us in the atmosphere. Air pressure gets lower at high altitudes and lowers at sea level. The air pressure also changes as warm and cool air circulates in the atmosphere.



LET'S CODE!

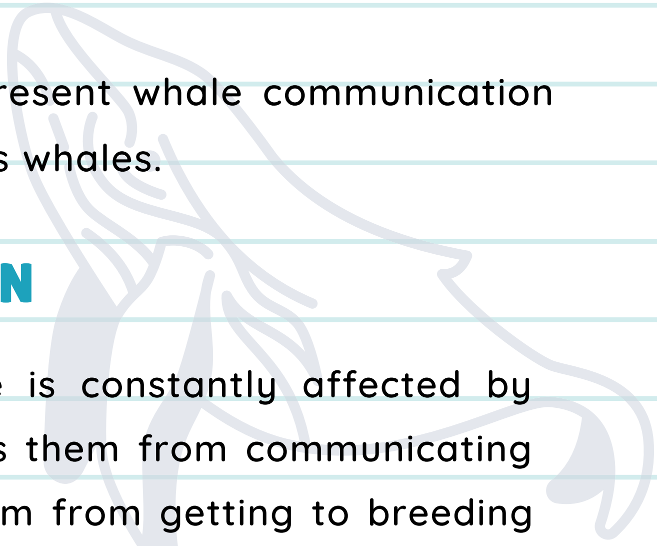


```
forever
    show string "temperature"
    show number Get temperature
    show string "humidity"
    show number Get humidity
    show string "pressure"
    show number Get pressure
```

1. Using the ‘show string’ block allows us to make text scroll across the screen. The ‘show number’ block does the same thing with numbers. Arrange the blocks as shown above to create a scrolling display of the enviro:bit’s weather sensor values.

2. Test it out with your micro:bit to get readings of the current temperature, humidity and pressure in the environment. Can you make any of these values change?

09 FINDING HOME–WHALE GAME



OVERVIEW

Use micro:bits radio signals to represent whale communication and show how noise pollution affects whales.

BACKGROUND INFORMATION

Whales and dolphins' everyday life is constantly affected by noise pollution. Noise pollution stops them from communicating effectively which often prevents them from getting to breeding grounds or important survival areas. Sometimes noise pollution can even lead to the death of many whales and dolphins. There are multiple approaches to reducing the impact of shipping noise on whales. One method is to increase the distance from shipping traffic to common whale habitats. Another method is to make the ship quieter. This can be achieved by making quieter engines and propellers or by reducing the speed of ships in areas where whales are commonly spotted.

NON–COMPUTER ACTIVITY

1. **Pick someone in your class who would like to be directed to a specific location and blindfold the person.**
2. **Pick someone else to give directions. - the Instructor**
3. **Select the destination and ensure the path to the location is a safe one.**
4. **Now the instructor's job is to help the blindfolded person to get to their destination by just giving directions using - Forward, Backwards, Left or Right.**
5. **Once the person gets to the destination, your job is complete. Congratulations!!!**

FINDING HOME–WHALE GAME

6. Now, let’s pick a new person to be blindfolded and a couple of new instructors. Let's say three.

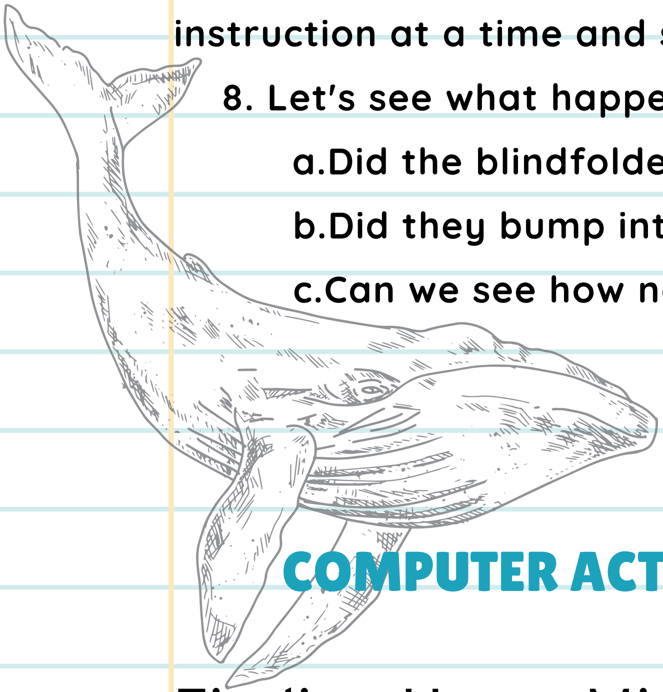
7. Only one instructor will be giving the correct directions. The other two will be giving incorrect instructions. Sometimes one instruction at a time and sometimes multiple instructions.

8. Let's see what happens.

   a.Did the blindfolded person make it to their destination?

   b.Did they bump into things?

   c.Can we see how noise pollution affects sea creatures?



This exercise requires the use of three participants with their micro:bits.

COMPUTER ACTIVITY

Finding Home Micro:bit Radio Game

In this Activity, we will use the micro:bits to demonstrate how noise pollution from shipping can affect whales and their ability to communicate with each other and to navigate.

This exercise requires the use of three participants with their micro:bits. This means we will need three micro:bits with 3 different sets of code. One set of code will be for the Whale, one set of code is for sending the correct Directions to the whale, and one set of code will disrupt the directions and mimic the interruptions caused by shipping and noise pollution.

FINDING HOME–WHALE GAME

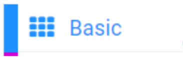
- The goal of the game is to direct the whale to a specified location. So there will be a "whale" micro:bit that will be receiving instructions from the "directions" microbit and the "disruptions" microbit.
- The "directions" and the "disruptions" players will secretly share with each other the location they want the "whale" to end.
- They will then select a neutral location for the "whale" to start.
- Then finally using their micro:bits they will send messages using the buttons to direct the "whale". It is important to pause after each sent instruction so that the "whale" gets time to follow orders that are sent.

This exercise requires the use of three participants with their micro:bits.

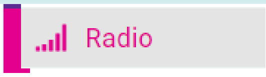
COMPUTER ACTIVITY

Whale Code - Microbit 1

1.The first thing we want to do is to create a radio group. This way, we can ensure our micro:bits are listening to the correct signals.

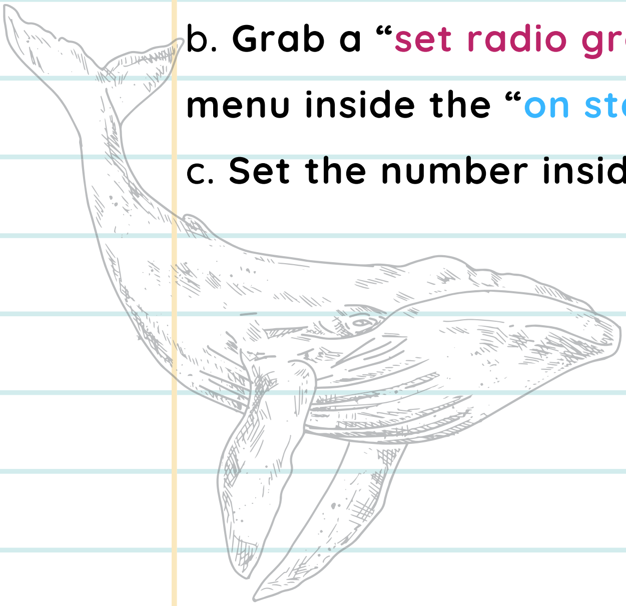
a. Grab an "on start" block from the "Basic" 

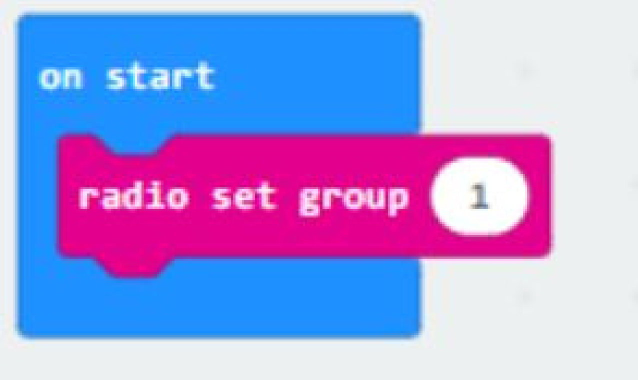
menu onto the workspace. If you already have one in your workspace, you can skip this step.

b. Grab a "set radio group" block from the "Radio" 

menu inside the "on start" block.

c. Set the number inside the group to "1"





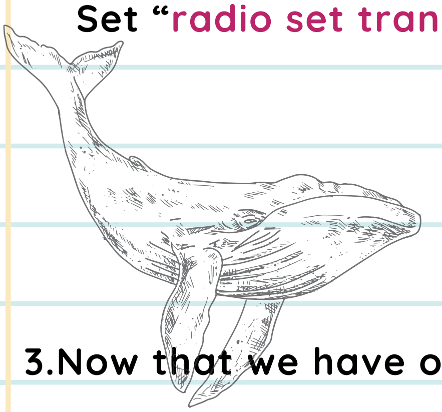
```
on start
  radio set group 1
```

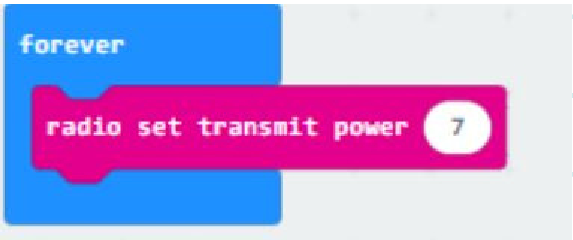
2.Next, we are going to set the radio signal strength.This ensures we can receive our radio transmissions from a fair distance away.

a. Grab a “forever” block from the “Basic” menu and drop it into the workspace.

b. Grab a “radio set transmit power” from the “more” ··· more menu under the “Radio” menu and drop it inside the “forever” block.

Set “radio set transmit power” to “7”.





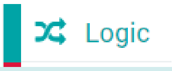
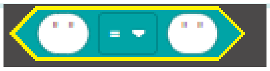
```
forever
  radio set transmit power 7
```

3.Now that we have our radio groups and transmit power setup, we are going to move on to receiving radio signals and interpreting them. For this exercise, we are going to be using micro:bits to send number messages via radio signal. When we receive those messages, we are going to check to see what number we receive in the radio message. Once we verify what the number is, we are going to light up the LEDs on our micro:bits so it can show us a directional arrow.

a. From the “Radio” menu, grab a “on radio receive receivedNumber” and drop it into your workspace.

b. From the “Logic” ⤮ Logic menu, grab the conditional “if <true> then, else” code block and drop it into the “on radio receive receivedNumber” block.

i. This block of code is going to check to see if certain conditions are met and if they are true, it’s going to run the code we put inside of it.

c. From the “Logic”  menu, grab the comparison equal block  and drop it in the spot that says “true” on the “if <true> then, else” code block.

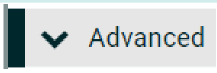
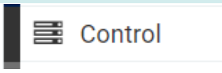
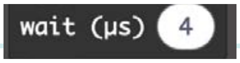
d. Next we are going to click on the “receivedNumber” part of our “on radio receive receivedNumber” and drag it into the first blank space of our comparison equal block.

e. Type the number “0” into the second blank space of our comparison equal block.

f. From the “Basic” menu, grab a “show leds” code block and drop it in the space between the “if” and “else” of the “if <true> then, else” code block.

g. Draw an arrow pointing to the left by clicking on the squares in the “show leds” code block.

h. Now we will add two blocks of code to clear our screen after displaying our arrow.

i.Click the “Advanced”  menu, then the “Control”  menu and grab a “wait (µs)” code block and drop it under the “show leds” code block. 

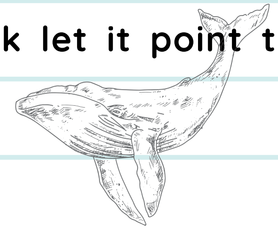
ii.Type the value “10000” into it. This is about 1 second.

iii.From the “Basic” menu, grab a “clear screen” code block and drag it under the “wait (µs)” code block.

iv.Here we are reading to see if the “receivedNumber” = “0” then we are going to show an arrow pointing left on our micro:bit, wait for about a second and then clear our screen.

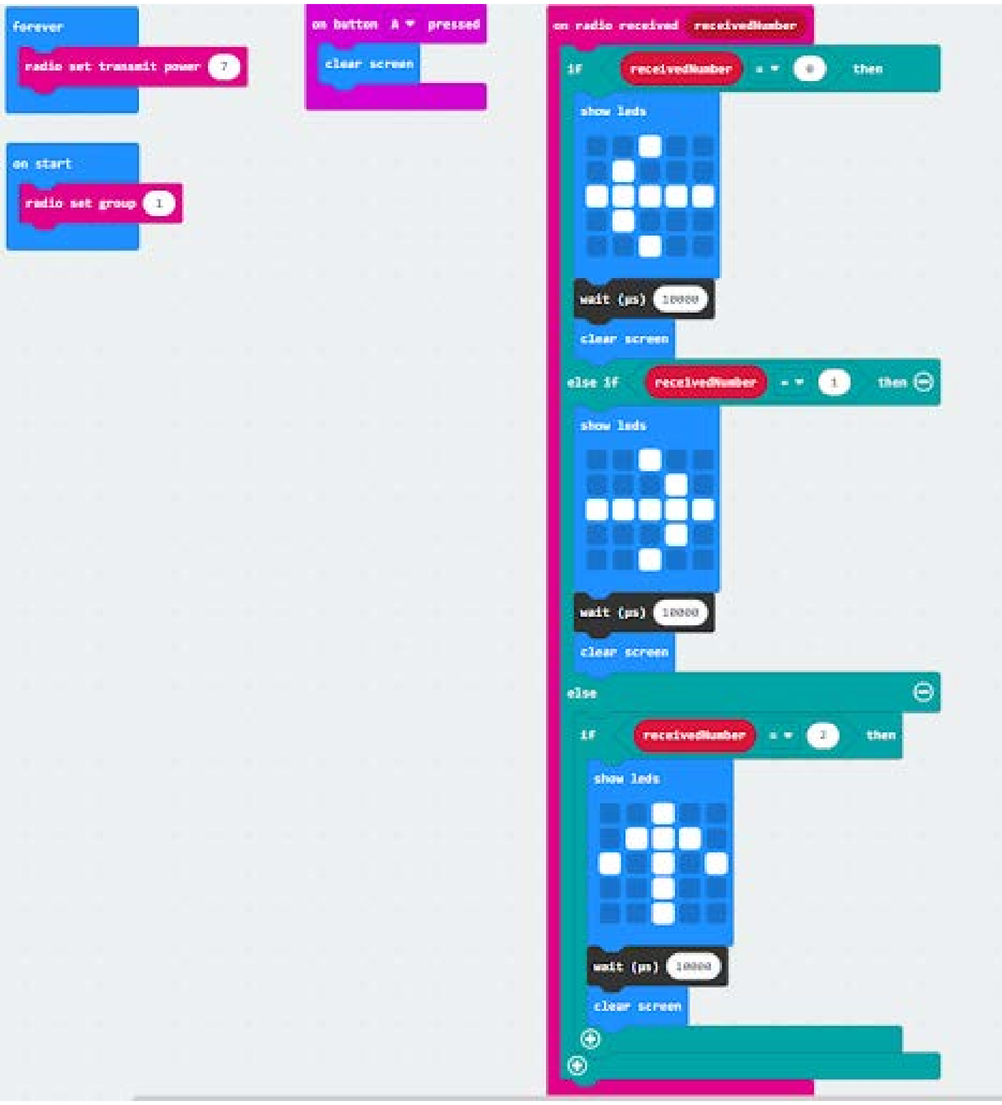
i. Click the “+” symbol under the “else” from the “if <true> then, else” code block so that we can get the “else if” conditional to show up.

j. Repeat steps “3-c” to “3-h-iii” and in our second spot for our comparison equal block, type the number “1” instead of “0” and when drawing your arrow in the “show leds” code block let it point to the right this time.



k. Now for our final conditional, we are going to grab an "if <true> then" conditional code block from the "Logic" code menu and drop it in the spot between "else" and the "+" symbol of our "if <true> then, else" code block.

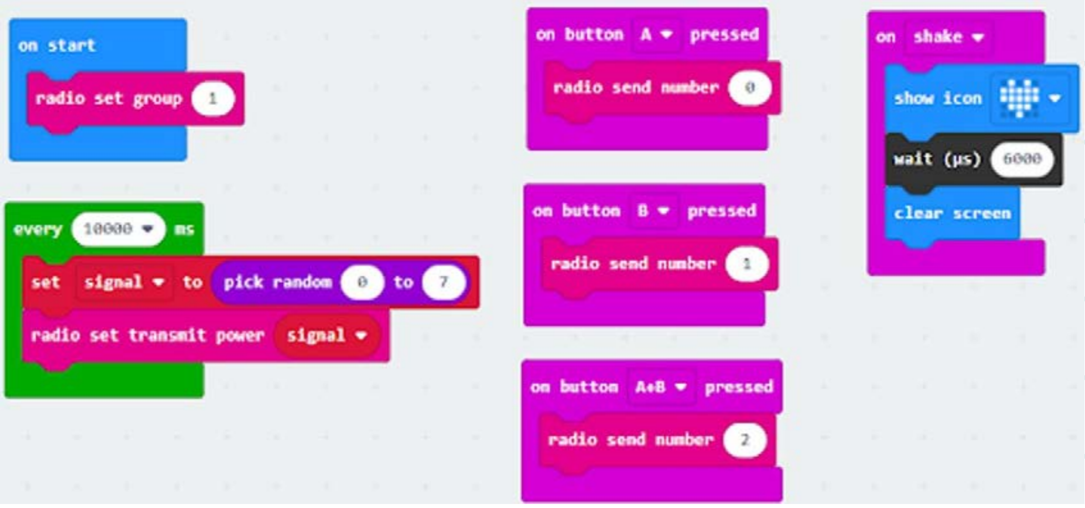
l. We are going to again repeat steps "3-c" to "3-h-iii" and this time, in the second blank spot of our comparison equal block, we are going to type "2" when drawing your arrow in the "show leds" code block. Let it point up this time.



```
forever
    radio set transmit power 7

on start
    radio set group 1

on button A pressed
    clear screen

on radio received receivedNumber
    if receivedNumber = 0 then
        show leds
        wait (µs) 10000
        clear screen
    else if receivedNumber = 1 then
        show leds
        wait (µs) 10000
        clear screen
    else
        if receivedNumber = 2 then
            show leds
            wait (µs) 10000
            clear screen
```

Congratulations!!! We have finished coding the "Whale" part of our game. We will now move on to the part of the code where we send instructions to our listening whale. This next step is a lot simpler and does not require as much coding.

START A NEW PROJECT AND GIVE IT A NAME
DIRECTIONS CODE -- MICRO:BIT 2



```
on start
    radio set group 1

every 10000 ms
    set signal to pick random 0 to 7
    radio set transmit power signal

on button A pressed
    radio send number 0

on button B pressed
    radio send number 1

on button A+B pressed
    radio send number 2

on shake
    show icon ♥
    wait (µs) 6000
    clear screen
```

1. The first thing we want to do is to create a radio group. This way we can ensure our micro:bits are listening to the correct signals.
    a. Grab an "on start" block from the "Basic" menu onto the workspace. If you already have one in your workspace you can skip this step.
    b. Grab a "set radio group" block from the "Radio" menu inside the "on start" block.
    c. Set the number inside the group to "1"
2. Next, we are going to set the radio signal strength. In our whale code, we set the strength to "7", but this time we are going to set our strength to a random number to accommodate for noise pollution and message interruptions. Sometimes our signal strength is going to be high and our whale will always get our instructions, but sometimes it might get interrupted and our whale might get the wrong information or no information at all.
    a. From the "Loops" ⟳ Loops menu, grab an "every" code block and drop it into the workspace.
    b. Next, we are going to create a variable, this is a container that is going to store some information that will change from time to time.

c. To create a variable, click the “Variables” , then click the “Make a Variable..” option, this will give you a pop-up menu where you can name your variable. We will name our variable “signal” so that we know this is where we are storing/saving our signal strength, then click “Ok”.

d. If your “Variables" menu was empty before, it should now have three new code blocks. If it wasn’t empty, you should now see an additional code block named “signal” along with any other variables that were there before.

e. Now we are going to set our “signal” variable to a random number. From the “Variables” menu drag and drop the “set signal to” into our “every” code block. If you don’t see a “set signal to” code block, this probably means you already had some variables created. Just drag the “set xxxx to” into the “every” code block and click the little white triangle and select the signal.

f. Next, we are going to go to the “Math” Math menu and drag and drop the “pick random (0) to (10)” code block and drop it into the number spot of our “set signal to” code block.

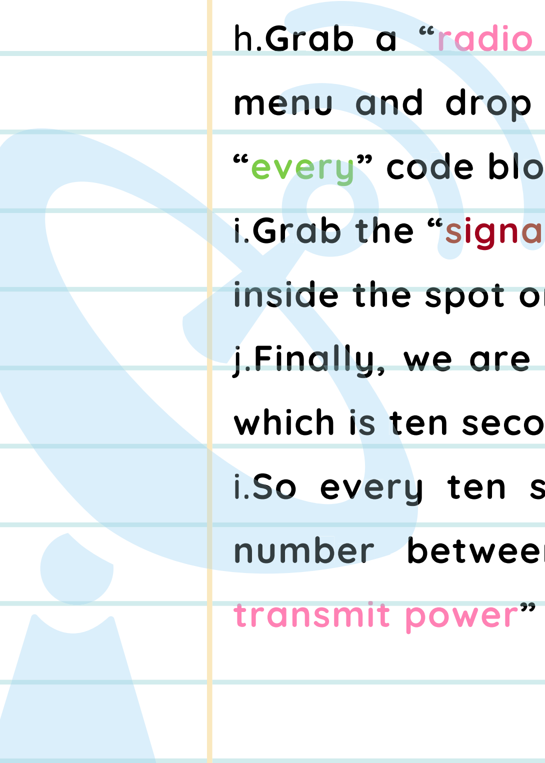
g.Our radio transmit power can only go as high as “7”, so we are going to set our “pick random (0) to (10)” to “0 - 7”.

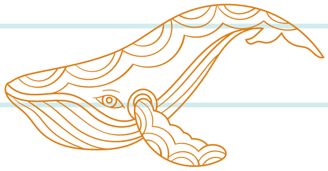
h.Grab a “radio set transmit power” code block from the “more” menu and drop it under the “set signal to” code block inside the “every” code block.

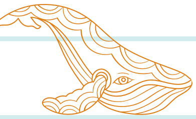
i.Grab the “signal” code block from the “Variables” menu and drop it inside the spot on the “radio set transmit power” code block.

j.Finally, we are going to set our “every” code block to “10000” ms, which is ten seconds.

i.So every ten seconds, we are setting our “signal” to a random number between zero and seven, then setting our “radio set transmit power” to that number.







3.Now let's send our various signals to our whale.

a. Select the "Input" ⊙ Input menu and drag and drop the "on button A pressed" code block 3 times into our workspace.

b. Click the little triangle next to the "A" on the "on button A pressed" code block and select "B" on one of the blocks and another select the "A+B" option.

c. We should now have three different blocks. "on button A pressed", "on button B pressed" and "on button A+B pressed"

4.From the "Radio" menu, grab a "radio send number" code block three times to the workspace.

a. Set the numbers in the spots to "0", "1" and "2" on the "radio send number" spots.

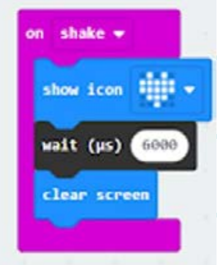
b. Drag the "radio send number" with the "0" value and drop it into the "on button A pressed", the one with the "1" value into the "on button B pressed" and the one with the "2" value into the "on button A+B pressed" code blocks.

c. Now when we press the respective button on the micro:bit, it will send those numbers to our whale micro:bit.

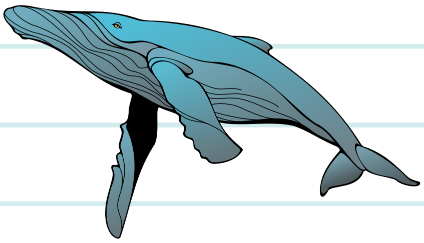
5. So that we don't confuse the directions code with the disruptions code, let's create a shake function so we know which micro:bit does what.

a. Select the "Input" menu and drag and drop the "on Shake" code block into our workspace. ⊙ Input

b. From our "Basic" menu select the "show icon" code block and drop it into our on Shake" code block. Select an icon you like. I chose the heart icon.



```
on shake ▾
  show icon ▾
  wait (µs) 6000
  clear screen
```

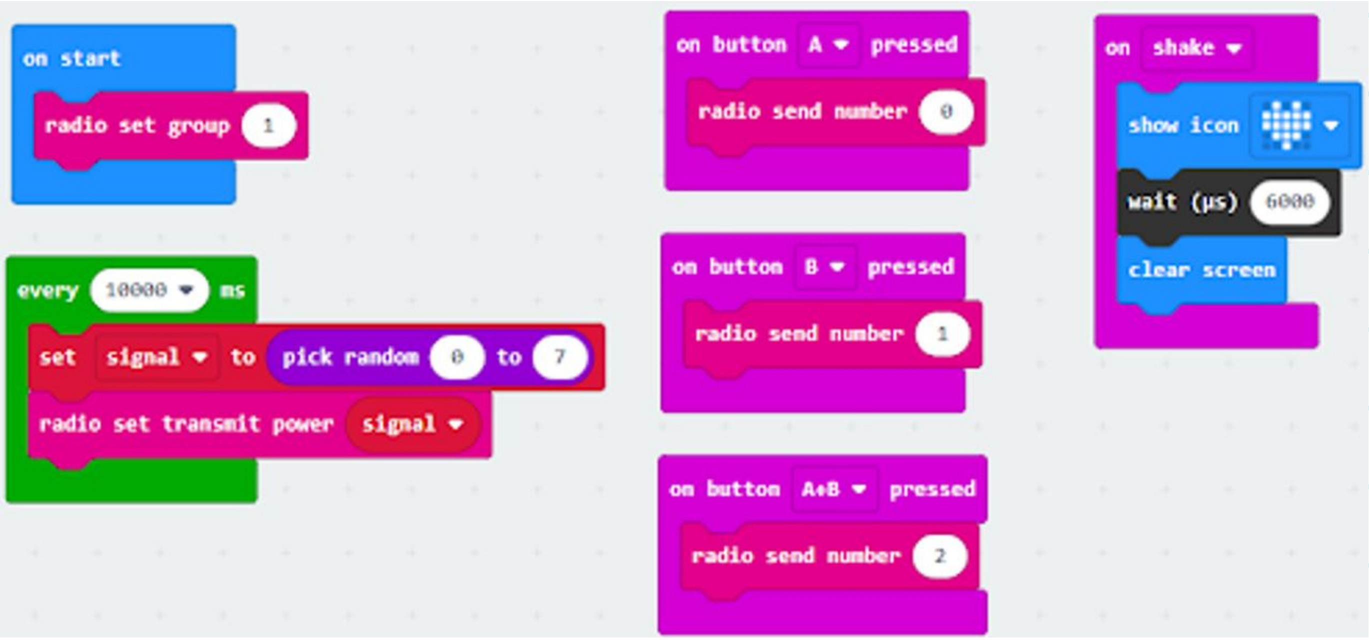


c. Click the “Advance” dropdown from the menu and you will get a few more options. There you should be able to see “Control”

d. In the “Control” menu grab the “wait” code block into our “on Shake” code block

e. Below our “show icon” code block, set your wait value to “6000”.

f. From the “Basic” menu drag the “clear screen” code block and drop it into our “on Shake” code block below our “wait” code block.

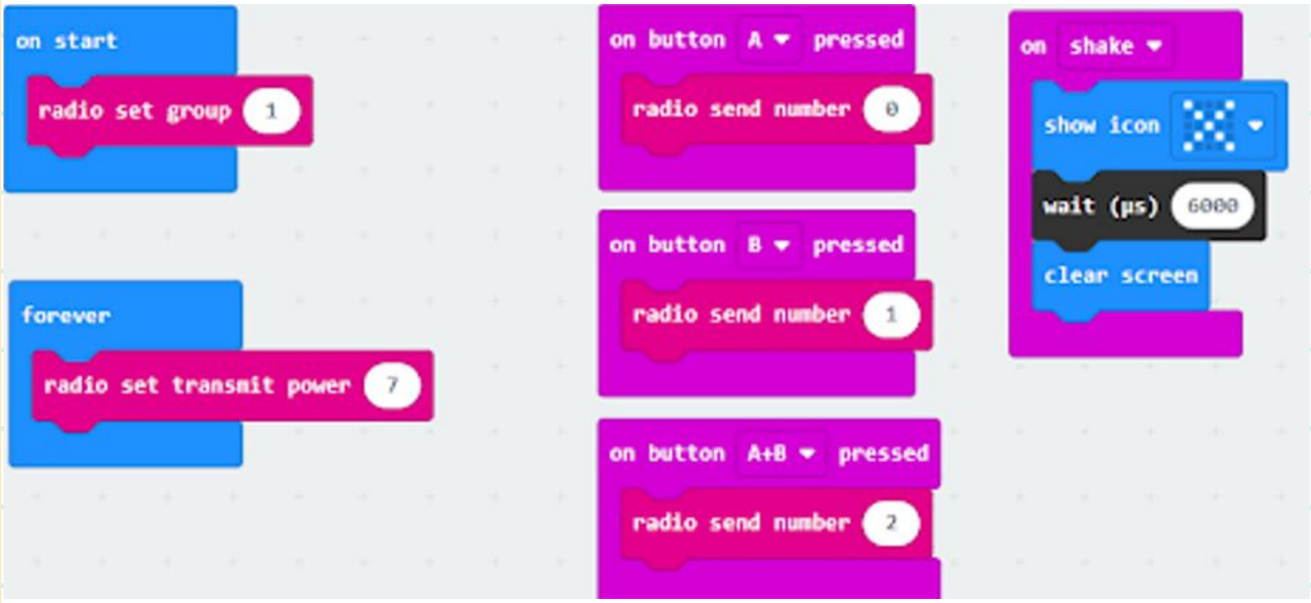


```
on start
    radio set group 1

every 10000 ms
    set signal to pick random 0 to 7
    radio set transmit power signal

on button A pressed
    radio send number 0

on button B pressed
    radio send number 1

on button A+B pressed
    radio send number 2

on shake
    show icon
    wait (µs) 6000
    clear screen
```

Fantastic! You have now coded micro:bit 2 to send the directions to the whale!

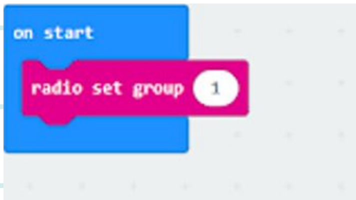
START A NEW PROJECT AND GIVE IT A NAME
INTERRUPTIONS CODE –– MICRO:BIT 3

This is how we will code the micro:bit that will be interrupting and disrupting the correct directions.



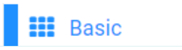
```
on start
    radio set group 1

forever
    radio set transmit power 7

on button A pressed
    radio send number 0

on button B pressed
    radio send number 1

on button A+B pressed
    radio send number 2

on shake
    show icon
    wait (µs) 6000
    clear screen
```

1. We are going to follow the same design as the code we wrote for the Directions Micro:bit 2, but without the fluctuation of the radio signal strength.
2. The first thing we want to do is to create a radio group. This way, we can ensure our micro:bits are listening to the correct signals.
    a. Grab an “on start” block from the “Basic” menu onto the workspace. If you already have one in your workspace, you can skip this step. ▦ Basic
    b. Grab a “set radio group” block from the “Radio” menu inside the “on start” block. .ıl Radio
    c. Set the number inside the group to “1”



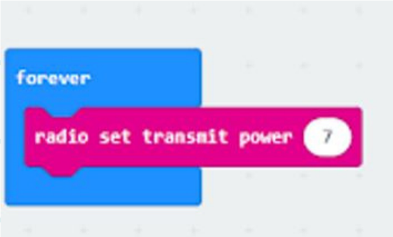
```
on start
    radio set group 1
```

3. Next, we are going to set the radio signal strength. This ensures we can receive our radio transmissions from a fair distance away.

a. Grab a “forever” block from the “Basic” menu and drop it into the workspace. 

b. Grab a “radio set transmit power” from the “more” menu under the “Radio” menu and drop it inside the “forever” block.

c. Set “radio set transmit power” to “7”.



```
forever
    radio set transmit power 7
```

4. Now let's send our various signals to disrupt our whale.

a. Select the “Input” menu and drag and drop the “on button A pressed” code block 3 times into our workspace.

b. Click the little triangle next to the “A” on the “on button A pressed” code block and select “B” on one of the blocks and another select the “A+B” option.

c. We should now have three different blocks. “on button A pressed”, “on button B pressed” and “on button A+B pressed”

5. From the “Radio” menu, grab a “radio send number” code block three times to the workspace.

a. Set the numbers in the spots to “0”, “1” and “2” on the “radio send number” spots.

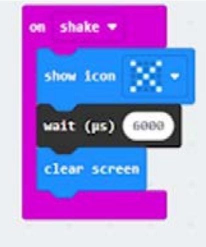
b. Drag the “radio send number” with the “0” value and drop it into the “on button A pressed”, the one with the “1” value into the “on button B pressed” and the one with the “2” value into the “on button A+B pressed” code blocks.

c. Now when we press the respective button on the micro:bit, it will send those numbers to our whale micro:bit.

6. So that we don't get confused by which microbit is sending the directions and which micro:bit is sending the interruptions, let's show a different icon that will be triggered by the shake function.

a. Select the "Input" menu and drag and drop the "on Shake" code block into our workspace. ⦿ Input

b. From our "Basic" ▦ Basic menu select the "show icon" code block and drop it into our on Shake" code block. Select an icon you want to represent the interruptions and disruptions. I've chosen an "X".



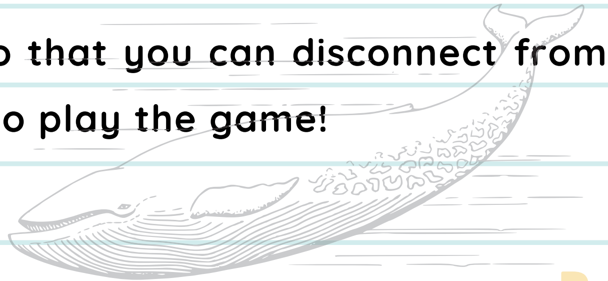
```
on shake ▾
    show icon ▾
    wait (µs) 6000
    clear screen
```

7. Click the "Advance" dropdown from the menu and you will get a few more options. There you should be able to see "Control"

a. In the "Control" menu grab the "wait" code block into our "on Shake" code block below our "show icon" code block, set your wait value to "6000".

b. From the "Basic" menu drag the "clear screen" code block and drop it into our "on Shake" code block below our "wait" code block.

Terrific! You have now coded the micro:bit that is interrupting and disrupting the flow of correct directions going to the whale! Make sure you have downloaded the code onto each of the 3 micro:bits. Connect each micro:bit to a battery pack so that you can disconnect from the computer. You are now ready to play the game! Try to get the whale safely home!



Congratulations!

You have learned how to use technology to help you learn about and take care of the land.

You have earned your position as Environmental Steward.

We hope you have enjoyed coding the micro:bit and using the sensors on the enviro:bit to observe your environment.

Look for more micro:bit activities at Pinnguaq.com.

PINNGUAQ

CONGRATULATIONS!