Welcome to the labs!

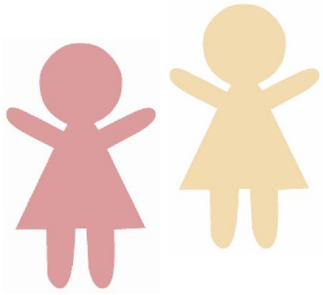
Bop It!- Micro:Bit

Who are the tutors?

Who are you?

Two Truths and a Lie

1. Get in a group of 3-5 people
2. Tell them three things about yourself:
   a. Two of these things should be true
   b. One of these things should be a lie!
3. The other group members have to guess which is the lie



Log on

Log on and jump on the GPN website

http://bit.ly/gpn-2019-4

You can see:

- These **slides** (to take a look back or go on ahead).
- A digital copy of your **workbook**.
- Help bits of text you can **copy and paste**!

There's also links to places where you can do more programming!

Tell us you're here!

Click on the
Start of Day Survey
and fill it in now!

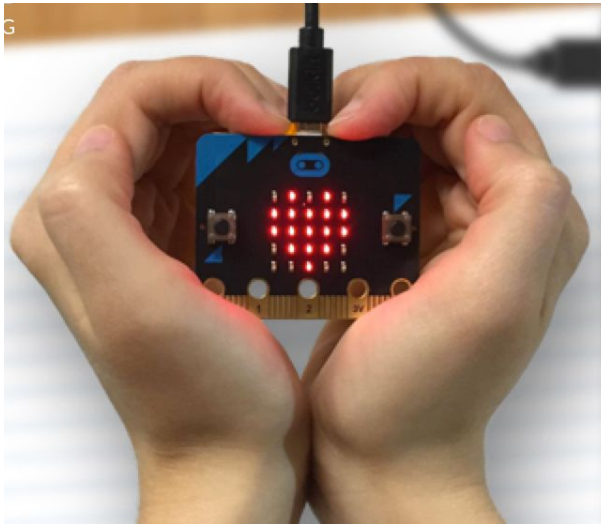
Today’s project!

Bop It! - Micro:Bit

Micro:Bits - IRL

Today we have real life MicroBits to use!

But sad you can’t keep them at the end of the day. 😢



If you want one for home (maybe for christmas or your birthday! They’re are about $25 .

Find out where to buy them at the bottom of this page
https://groklearning.com/microbit/

Micro:Bits - Digital

We also have an emulator in Grok Learning! Which you can use after the workshop! 🎉

The emulator is a fast way to test the code without downloading it.

Use it while you're still working on your code. And then try it in real life.

Using the workbook!

The workbooks will help you put your project together!

Each **Part** of the workbook is made of tasks!

Tasks - The parts of your project

Follow the tasks **in order** to make the project!

Hints - Helpers for your tasks!

Stuck on a task, we might have given you a hint to help you **figure it out**!

The hints have **unrelated** examples, or tips. **Don't copy and paste** in the code, you'll end up with something **CRAZY**!

Task 6.2: Add a blah to your code!

This has instructions on how to do a part of the project

1. **Start by doing this part**
2. **Then you can do this part**

Task 6.1: Make the thing do blah!

Make your project do blah ….

Hint

A clue, an example or some extra information to help you **figure out** the answer.

```
print('This example is not part of the project' )
```

Using the workbook!

The workbooks will help you put your project together!

Check off before you move on from a **Part**! Do some bonuses while you wait!

Checklist - Am I done yet?

Make sure you can tick off every box in this section before you go to the next Part.

Lecture Markers
This tells you you'll find out how to do things for this section during the names lecture.

Bonus Activities
Stuck waiting at a lecture marker?
Try a purple bonus. They add extra functionality to your project along the way.

☑ **CHECKPOINT** ☑

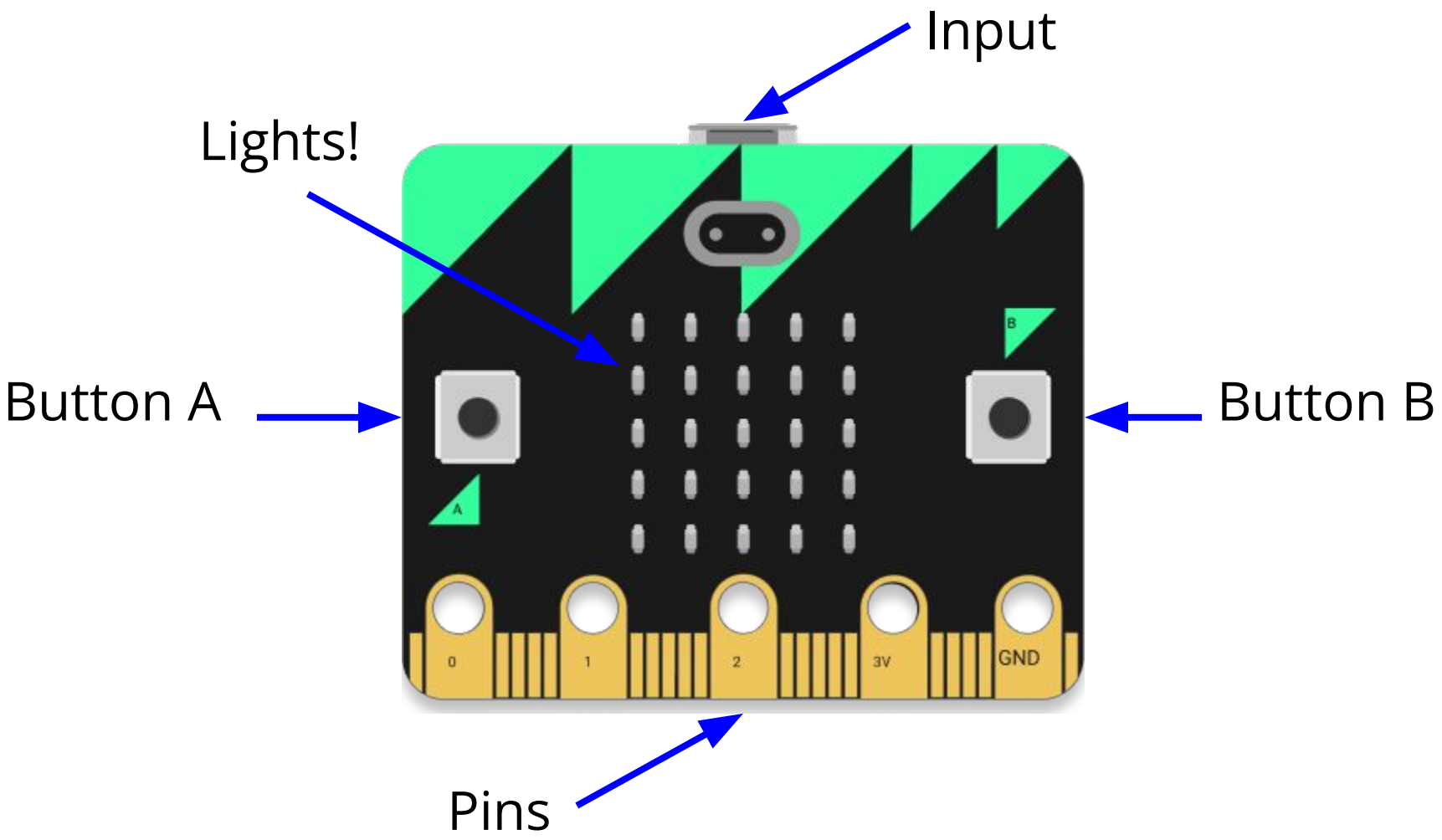
If you can tick all of these off you're ready to move the next part!
☐ Your program does blah
☐ Your program does blob

★ **BONUS 4.3: Do some extra!**

Something to try if you have spare time before the next lecture!

Intro to Micro:Bit

What is a Micro:Bit?



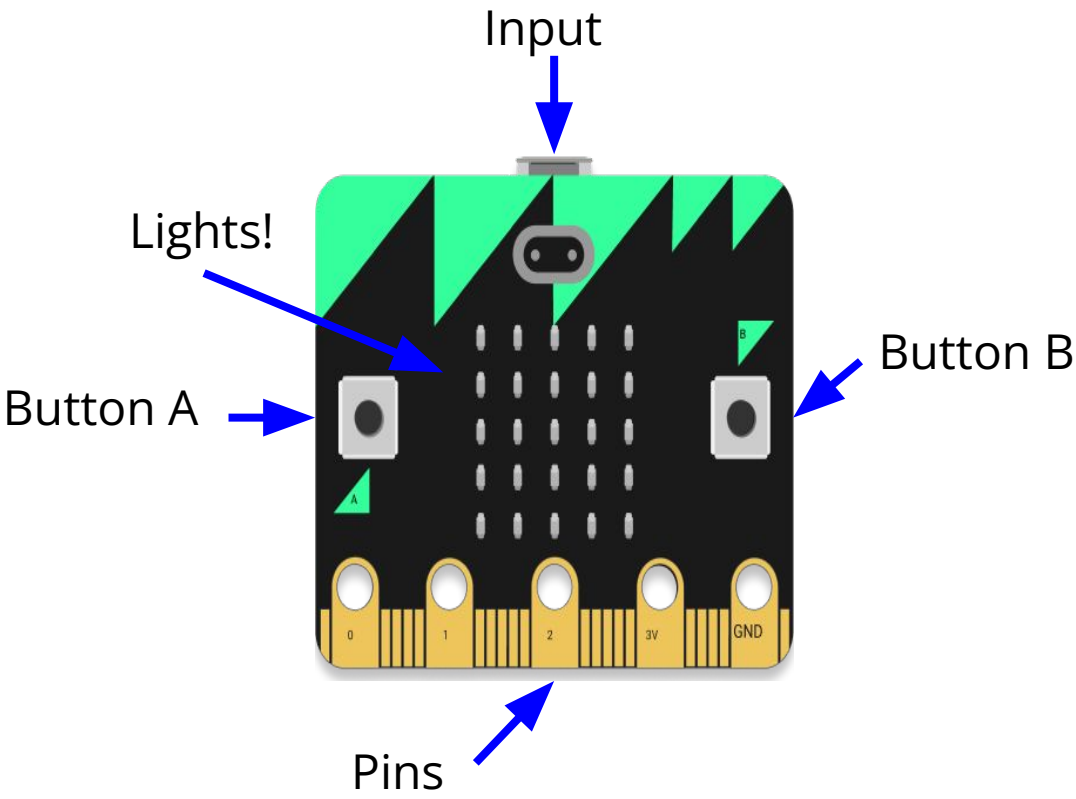
What do the different bits do?

Input: This is how we get code onto our Micro:Bit and tell it what to do!

Buttons: We can press these and tell the Micro:Bit to do different things when we do

Lights: Each of these is a little light that we can turn on. When we turn them on in different patterns we can make images!

Pins: These let us connect the MicroBit to other devices like extra buttons



How do we write code for it?

Micro:Bits use Python, which is the programming language that we usually teach here at GPN!

Because they have buttons, lights and other cool stuff we need to make sure that we tell Python that we want the extra stuff for Micro:Bits. We do this using this line of code:

```
from microbit import *
```

Always make sure this line is at the top of your code!

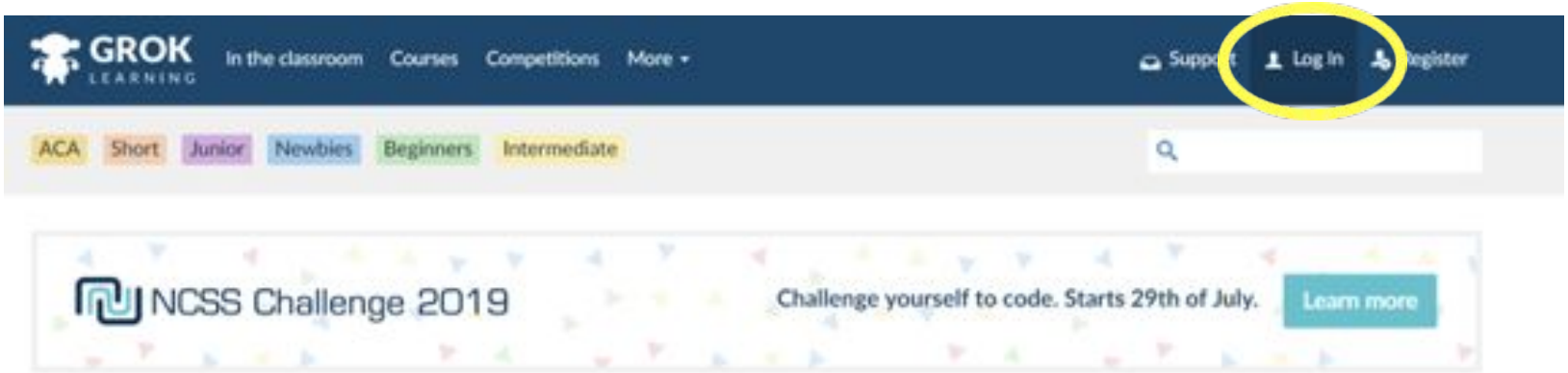
Using Grok Learning!

Today we will be using Grok Learning to program our Micro:Bits.

Grok has a great digital Micro:Bit which makes learning how to program them really easy!

Getting to Grok!

Go to groklearning.com



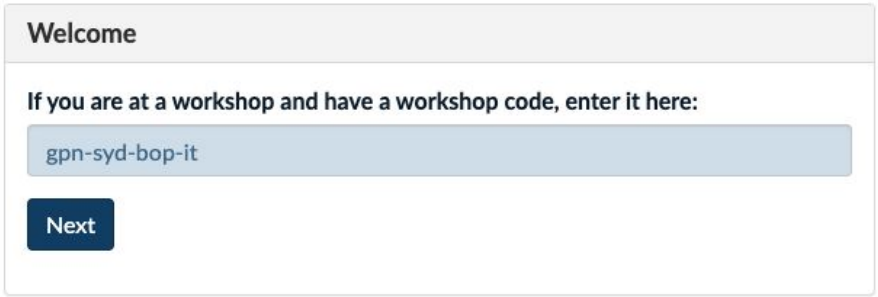
Log in with the email address you signed up to GPN with

Getting into the GPN Workshop

Next go to your profile name and click "Workshops"

You will be asked for a workshop code.

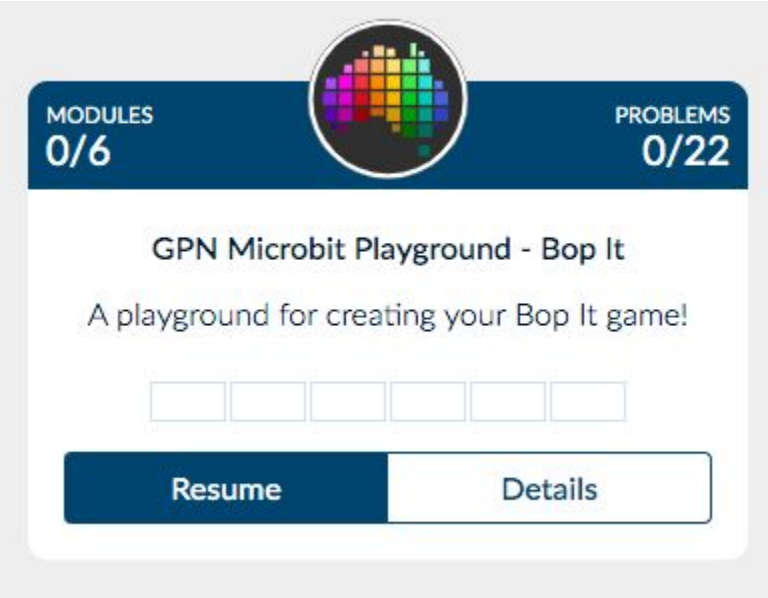
Our code is gpn-syd-bop-it



You can use your school name or Girls' Programming Network - University of Sydney as your institution

GPN MicroBit Playground

Once you're in the workshop click on the GPN MicroBit Playground



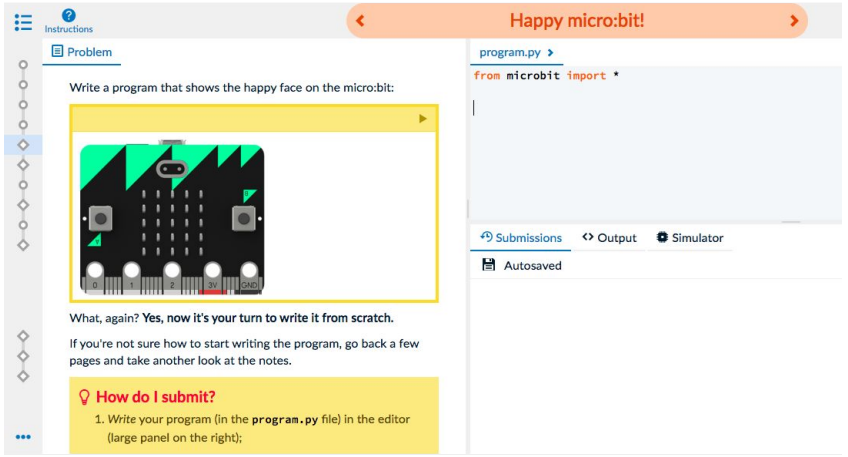
Slides and Problems

The first part of the workshop today we will be learning about Micro:Bits using Grok!

Grok has 2 different types of pages: slides and problems!

Slides look like this and teach you about the Micro:Bit

Problems look like this and they are your chance to practice what you've learned



Using a MicroBit IRL

It's fun to mess around with the Micro:Bit in Grok but it's also really fun to see your code on a MicroBit in real life!

To get your code from Grok onto your Microbit:

1. Plug your Micro:Bit into your computer
2. Click the Download button in Grok to download your code

3. Drag the downloaded .hex file onto your Micro:Bit (like you would with a usb)
4. Wait for the red light at the back to stop flashing and the code should be running!
5. If you want your code to start again from the beginning, press the "reset" button on the back

Onto the project!

Once you've done all the intro slides and problems it's time to work on our GPN Workbook of the day!

After the last problem there are a bunch of empty problem slides! This is where you will be writing your project code for today.

You can use the digital Micro:Bit to test your code and when you're happy with it, you can download it and put it on your real MicroBit!

The Display

Your MicroBit has a display! It is the 5 by 5 grid of little red LEDs on the front! You can do some cool stuff with the display like:

Scroll the words “Hello World” across the display

```
display.scroll(“Hello World”)
```

Show an image, like a happy face!

```
display.show(Image.HAPPY)
```

There are lots of images you can make the display show like GIRAFFE, DUCK and GHOST so have a play with them

Project Time!

Let’s get started!

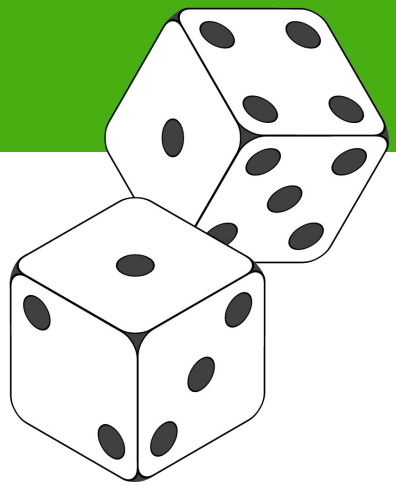
Let’s try use it in our project!

Try to do Part 0 and 1

The tutors will be around to help!

Random!

That's so random!



There's lots of things in life that are up to chance or random!

We want the computer to be random sometimes!

Python lets us **import** common bits of code people use! We're going to use the **random** module!

Using the random module

Let's choose something randomly from a list!

This is like drawing something out of a hat in a raffle!

Try this!

1. Import the random module!

```
>>> import random
```

2. Copy the shopping list into IDLE

```
>>> shopping_list = ["Bread", "Chocolate", "Ice Cream",
    "Pizza"]
```

3. Choose randomly! Try it a few times!

```
>>> random.choice(shopping_list)
```

Using the random module

You can also assign your random choice to a variable

```
>>> import random
>>> shopping_list = ["Bread", "Chocolate", "Ice Cream",
    "Pizza"]
>>> random_food = random.choice(shopping_list)
>>> print(random_food)
```

Project Time!

Raaaaaaaaaandom! Can you handle that?

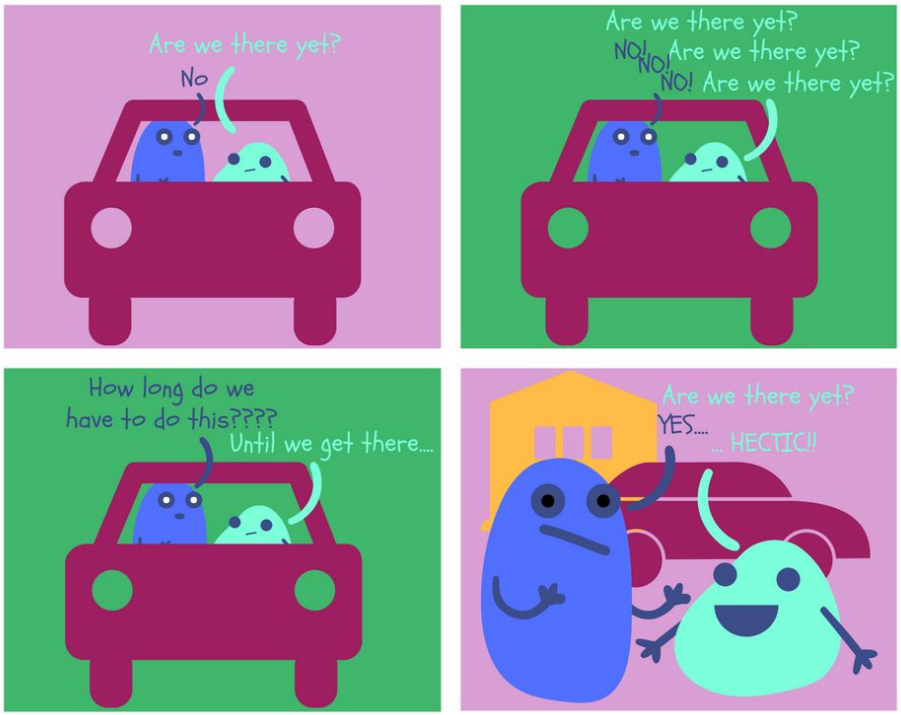
Let's try use it in our project!

Try to do Part 2

The tutors will be around to help!

While Loops

Loops



We know how to do things on repeat!

Sometimes we want to do some code on repeat!

Introducing ... `while` loops!

What do you think this does?

```
i = 0
while i < 3:
    print("i is " + str(i))
    i = i + 1
```

Introducing ... `while` loops!

What do you think this does?

```
i = 0
while i < 3:
    print("i is " + str(i))
    i = i + 1
```

```
i is 0
i is 1
i is 2
>>>
```

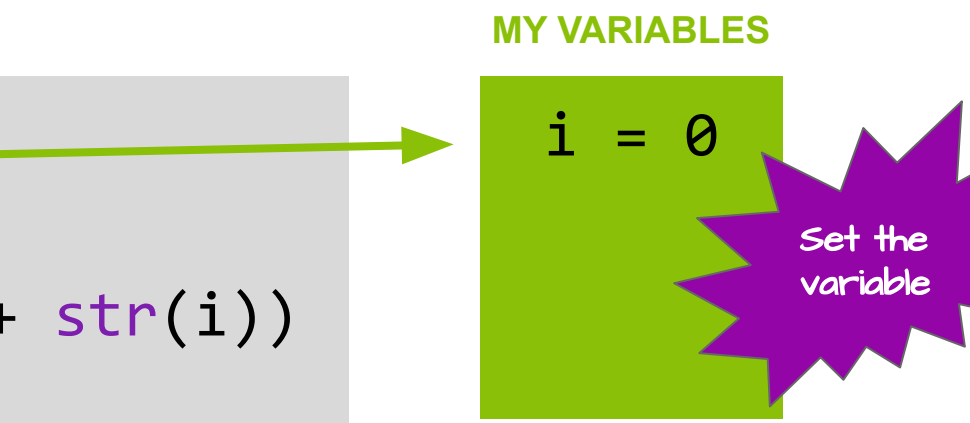
Introducing … `while` loops!

Stepping through a while loop…

Introducing … `while` loops!

One step at a time!

```
i = 0
while i < 3:
    print("i is " + str(i))
    i = i + 1
```



Introducing … `while` loops!

One step at a time!


```
i = 0
while i < 3:
    print("i is " + str(i))
    i = i + 1
```

MY VARIABLES

```
i = 0
```

Introducing … `while` loops!

One step at a time!


```
i = 0
while i < 3:
    print("i is " + str(i))
    i = i + 1
```

MY VARIABLES

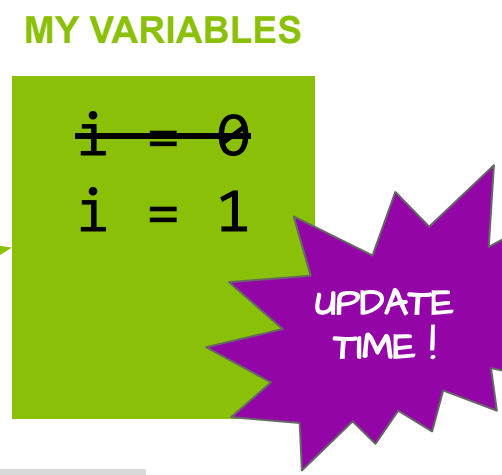
```
i = 0
```

```
i is 0
```

Introducing … `while` loops!

One step at a time!

```
i = 0
while i < 3:
    print("i is " + str(i))
    i = i + 1
```




```
i is 0
```

Introducing … `while` loops!

One step at a time!


```
i = 0
while i < 3:
    print("i is " + str(i))
    i = i + 1
```

MY VARIABLES

~~i = 0~~
i = 1

```
i is 0
```

Introducing … `while` loops!

One step at a time!


```
i = 0
while i < 3:
    print("i is " + str(i))
    i = i + 1
```

MY VARIABLES

~~i = 0~~
i = 1

```
i is 0
```

Introducing … `while` loops!

One step at a time!


```
i = 0
while i < 3:
    print("i is " + str(i))
    i = i + 1
```

MY VARIABLES

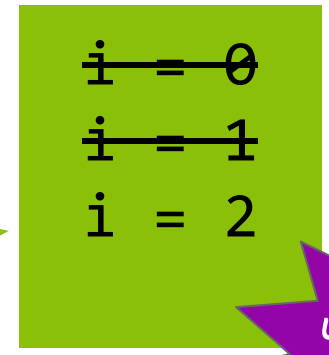
~~i = 0~~
i = 1

```
i is 0
i is 1
```

Introducing … `while` loops!

One step at a time!

```
i = 0
while i < 3:
    print("i is " + str(i))
    i = i + 1
```




```
i is 0
i is 1
```

Introducing … `while` loops!

One step at a time!


```
i = 0
while i < 3:
    print("i is " + str(i))
    i = i + 1
```

MY VARIABLES

~~i = 0~~

~~i = 1~~

i = 2

```
i is 0
i is 1
```

Introducing … `while` loops!

One step at a time!


```
i = 0
while i < 3:
    print("i is " + str(i))
    i = i + 1
```

MY VARIABLES

~~i = 0~~

~~i = 1~~

i = 2

```
i is 0
i is 1
```

Introducing … `while` loops!

One step at a time!


```
i = 0
while i < 3:
    print("i is " + str(i))
    i = i + 1
```

```
i is 0
i is 1
i is 2
```

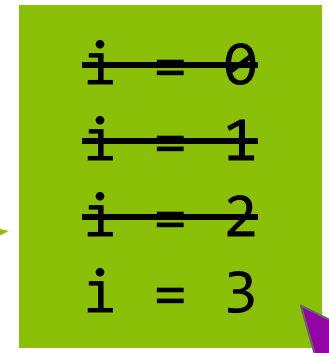
MY VARIABLES

~~i = 0~~
~~i = 1~~
i = 2

Introducing … `while` loops!

One step at a time!

```
i = 0
while i < 3:
    print("i is " + str(i))
    i = i + 1
```




```
i is 0
i is 1
i is 2
```

Introducing … `while` loops!

One step at a time!


```
i = 0
while i < 3:
    print("i is " + str(i))
    i = i + 1
```

MY VARIABLES

~~i = 0~~
~~i = 1~~
~~i = 2~~
i = 3

```
i is 0
i is 1
i is 2
```

Introducing ... `while` loops!

One step at a time!

3 IS NOT less than 3 !

```
i = 0
while i < 3:
    print("i is " + str(i))
    i = i + 1
```

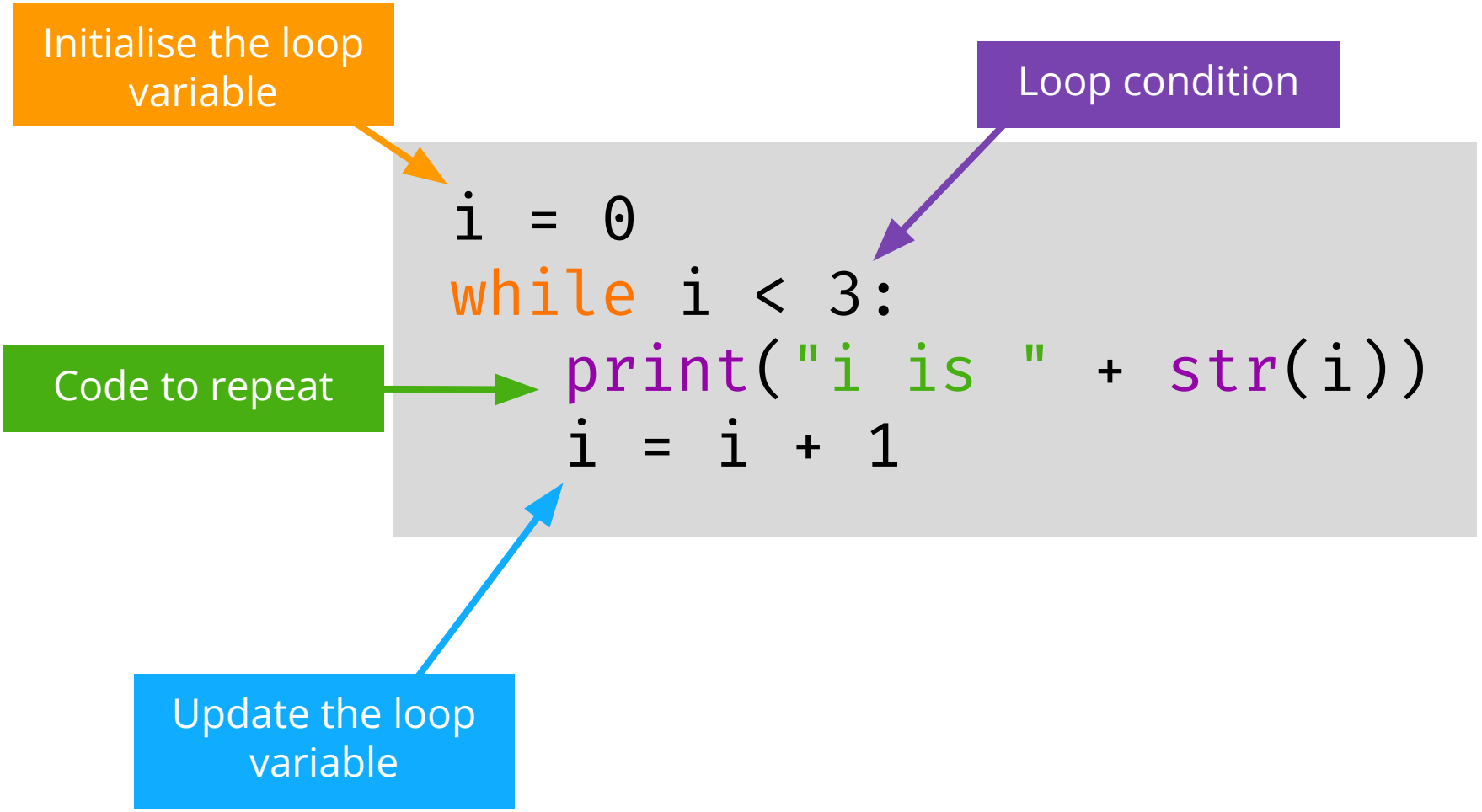
MY VARIABLES

~~i = 0~~
~~i = 1~~
~~i = 2~~
i = 3

We are are done with this loop!

```
i is 0
i is 1
i is 2
```

Introducing ... `while` loops!



What happens when.....

What happens if we forget to update the loop variable?

```
i = 0
while i < 3:
    print("i is " + str(i))
```

What happens when.....

What happens if we forget to update the loop variable?

```
i = 0
while i < 3:
    print("i is " + str(i))
```

```
i is 0
i is 0
i is 0
i is 0
i is 0
i is 0
i is 0
i is 0
i is 0
i is 0
i is 0
i is 0
i is 0
```

Infinite loop!

Sometimes we want our loop to go forever!

So we set a condition that is always True!

We can even just write True!

```
while True:
    print("Are we there yet?")
```

Give me a break!

But what if I wanna get out of a loop early?

That's when we use the break keyword!

```
number = 0
while number != 42 :
    number = input("Guess a number: ")

    if number = "I give up":
        print("The number was 42")
        break

    number = int(number)
```

Continuing on

How about if I wanna skip the rest of the loop body and loop again? We use continue for that!

```
number = 0
while number != 42 :
    number = input("Guess a number: ")

    if not number.isnumeric():
        print("That's not a number!")
        print("Try again")
        continue

    number = int(number)
```

Running Time

Sometimes you want to time things. Like, for example, if you wanted to put a time limit on a game and see how many points you can get in 30 seconds!

To figure out how long the Micro:Bit program has been running (in milliseconds) you can use this command:

```
time = running_time()
```

What would running_time() be after 4 seconds?

4000

What about after 10 and a half second?

10,000

Project Time!

while we're here:

Try to do Part 4!

The tutors will be around to help!

Micro:Bit Buttons

Buttons!

Your Micro:Bit has 2 buttons: Button A and Button B

You can use this code to check whether or not a button is pressed:

```
button_a.is_pressed()

button_b.is_pressed()
```

The statement will be **TRUE** if the button is being pressed at that time and it will be **FALSE** if it is *not* being pressed

Buttons!

What do you think this code does?

```
if button_a.is_pressed():
    display.show(Image.HAPPY)

if button_b.is_pressed():
    display.show(Image.SAD)
```

If ***button a*** is pressed when the Micro:Bit gets to this line of code then what happens?

The Micro:Bit shows a Happy face

If ***button b*** is pressed when the Micro:Bit gets to this line of code then what happens

The Micro:Bit shows a Sad face

What do you think happens if *both* button a AND button b are being pressed?

Micro:Bit Radio

Radio

Your Micro:Bit can send messages to other Micro:Bits using radio waves!

It only takes a few lines of code to make this work!

1. We have to tell the Micro:Bit that we want to use the radio:

```
import radio
```

2. We need to turn the Radio on:

```
radio.on()
```

3. We need to send a message:

```
radio.send("Hello World")
```

4. We want to receive a message:

```
message = radio.receive()
```

Radio Groups

We need to set our radio to communicate on a certain group, otherwise all our Micro:Bits will try to talk to each other! This will get confusing for the Micro:Bit.

After you turn the radio on, set the group channel!

```
radio.config(group=100)
```

Your tutors will give you a group number to use.

Radio Example

What :do you think this code does?

Micro:Bit 1

```
import radio

radio.on()
radio.config(group=100)

while True:
    if button_a.is_pressed():
        radio.send("Hello!")

    if button_b.is_pressed():
        radio.send("World!")
```

Micro:Bit 2

```
import radio

radio.on()
radio.config(group=100)

while True:
    message = radio.receive()
    if message:
        display.scroll(message)
```

Why do you think it's important to check the message?

Tell us what you think!

Click on the

End of Day Form

and fill it in now!