

Microbit Lesson 1: Getting started

Video

- [Microbit lesson 1 video](#)

Introducing



The Microbit

- Small board that can be easily programmed. Contains 5x5 grid of LEDs, 2 buttons, accelerometer, compass, touch sensors, temperature sensor, bluetooth radio

Python

- Popular and versatile programming language that is quite easy to learn.

Mu Editor

- The software on your laptop we will use. Look for **mu Editor** in your programs menu. Set mode to **BBC micro:bit** when prompted.

Connecting your Microbit

Your first program

```
from microbit import *  
  
# Display 'Hello world!' then the happy emoji  
display.scroll("Hello world!")  
display.show(Image.HAPPY)
```

(Demo)

Syntax of your first program

Features of your first program

- Statements are executed in order, top down - unless otherwise directed
- Importance of casing, consistent spelling and correct symbols

- The import statement - every microbit program will need this. Loads the Microbit functionality. Python has "core" functionality available to every program. If we want to use something not available by default, we can use import statements to tell Python to load it.
- Comments - every program should include comments.
- Scroll text - note the parenthesis and quote marks
- Images - there is quite a list available

Tips for correcting your programs (debugging)

- Run your code everytime you make a change.
- Read the error message.
- Google the error message.
- Guess and check - really it's ok to have a guess!
- Comment out code.
- Ask a peer for help - learn from each other!
- Ask the teacher.

When asking for help...

- Explain what you're trying to do.
- Show the code that's giving the error.
- Show the entire stack trace including the error message.
- Explain 2-3 things that you've tried already and why they didn't work.

Your second program

We can use the `sleep()` command to create a pause in our program as the following demonstrates...

```
from microbit import *  
  
display.show(Image.HAPPY)  
sleep(2000)  
display.show(Image.HEART)  
sleep(2000)  
display.show(Image.YES)  
sleep(2000)
```

Your third program

We aren't just restricted to the images already created for us, we can create our own with the `Image()` command.

For instance, if I woke up in the morning feeling sad, then felt neutral, then felt happy, I could show this with a Microbit. The neutral face doesn't currently exist, so I need to create it myself as follows...

```
from microbit import *  
  
display.show(Image.SAD)
```

```
sleep(2000)
display.show(Image("00000:09090:00000:99999:00000"))
sleep(2000)
display.show(Image.HAPPY)
sleep(2000)
```

The `Image()` command works by providing a brightness code for each individual LED on the board. `0` indicates LED off, `9` indicates LED full brightness on. We specify the five values for the five LEDs in each row, and use a `:` as a row separator.

Activity: Create an animated art piece

With time remaining create an animated art piece on the microbit.

You can use any mix of the commands shown so far.

For example:

- Stick person moving across screen
- Stick person jumping and landing
- However your creativity inspires...?