## Lesson 3 – Images, Variables and Functions

## Getting Started

Create a simple image on the LEDs by turning them ON or OFF and also adjusting the brightness of each LED.

The import command enables you to import program code to control the micro:bit. The \* means import all the modules, for example, LEDs, acceleration, buttons etc.

Enter the code shown below and download it to your micro:bit. It creates a simple image of a boat. Notice that it uses the import on the first line and then we create a variable named boat to hold the details of the image.

from microbit import \*

boat = Image("05050:"

"05050:"

"05050:"

"99999:"

"09990")

display.show(boat)

The numbers correspond to what you want the LED to do. A ‘0’ means the LED is off, ‘9’ is the LED ON with full brightness. The function display.show() is used to display the image with the boat variable passed to the function. Play around with the various settings.

## Creating an Animation

By creating variables containing different images and then using the display.show() function we can build up a simple animation. These are a series of images displayed one after the other which create the illusion that the image is moving. Consider the simple program to create a single dot moving across the bottom of the micro:bit.

from microbit import \*

from time import sleep

dot1 = Image("00000:"

"00000:"

"00000:"

"00000:"

"50000")

dot2 = Image("00000:"

"00000:"

"00000:"

"00000:"

"05000")

dot3 = Image("00000:"

"00000:"

"00000:"

"00000:"

"00500")

display.show(dot1)

sleep(0.5)

display.show(dot2)

sleep(0.5)

display.show(dot3)

sleep(0.5)

First, the variables **dot1, dot2, dot3** are created to store each array of LED images. Then we use the function display.show() to turn the LEDs ON and OFF. The next line adds a small pause before loading the displaying the next LED image.

* Can you complete the program?
* Now create your own animation.

## Success Criteria

1. Complete the example dot program so that the ‘dot’ moves across the bottom
2. Adjust the LED brightness
3. Create your own animation

## Pro-tip

Remember the image is built using an array or list, you need to ensure that the correct number of entries are made. A simple 5 × 5 grid, one to represent each LED. You can use these grids to plan your animation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

At the end of each line remember to include the : symbol

Close the Array with the close brackets **)**

## Test time

Download your animation program to your micro:bit and test it. Does it play as you expected? Maybe you need to adjust the timings. Show another student or your teacher, what do they think?

## Stretch Tasks

Create a more complex animation. Can you make a winking eye, a mouth talking, a person walking? How about combining two micro:bits and creating an animation looks like it moves between the two micro:bits.

## Final Thoughts

In this micro:bit project we have covered:

* Importing
* Images
* Functions as commands (arguments)
* Variables

Variables and functions are essential parts of computer programming and you will notice that you use them throughout this course. Perhaps keep a tally of the number of times you come across them.