Coding Fundamentals

```
# Add your Python code here. E
from microbit import *
while True:
    display.scroll('Hello, W
    display.show(Image.HEAR
    sleep(2000)
```

Micro:bit Python Programming

Variables and DIY Images

Overview

In this lesson, students learn to use variables to create an image on the micro: bit LEDs using Python.

Objectives

- Understand and use `display.set()` and `display.show()`
- Explain use of variables for composing images
- Create a program to display multiple custom images

Materials

- micro:bit and micro-USB cord
- Computer with access to the internet

Approx. Time Required

1-2 hours

Cyber Connections

- **Programming** – Students will program in Python.
- **Hardware and Software** – Students will utilize small electronics and learn how a computer is programmed while using microcontrollers.

CYBER.ORG
THE ACADEMIC INITIATIVE OF THE CYBER INNOVATION CENTER

This content is based upon work supported by the US Department of Homeland Security's Cybersecurity & Infrastructure Security Agency under the Cybersecurity Education Training and Assistance Program (CETAP).

Teacher Notes:

Variables and DIY Images

- Review `display.show()` by showing some of the images that are built in to the programming language.

- Preface the next section by presenting the image names (HEART, HOUSE, etc.) as variables that refer the program specifically to the images that they are named after.

- This lesson will familiarize students with the concept of a variable and teach them how to create variables. A ***variable*** is a name assigned to data to keep track of it and make it easier to use. When using any data repeatedly, it is a good idea to make it a variable.

- Show students the following code, both in the Python online editor and on the micro:bit running the program.

The colons inside the quotes for the image denote the end of the row of LEDs. They are required for the code to work.

```
from microbit import *

boat = Image(“05050:”
             “05050:”
             “05050:”
             “99999:”
             “09990”)

display.show(boat)
```

- The name of the variable created with this code is `boat`. `boat` is just a name that stores the data needed for the image. The numbers contain the light values for each pixel on the grid of LEDs. The variable name `boat` can be used in place of the image data. Each line must be enclosed in quotes, with the entire image inside parentheses.

- Have students recreate their own images from previous lessons using the same format shown above. Create a variable name. To maintain good coding practices, variable names should be simple and represent the image that they will hold.

- Once students have a firm grasp on variables by creating their own images, introduce a new tool: arrays. An array is an incredible instrument that allows multiple variables of the same type to be grouped together under one name. Imagine there are three images named `my_target`, `my_square`, and `my_arrow`. The code to group them together in an array would look like this:

```
My_images = [my_target, my_square, my_arrow]
```

- Note that all three variables are enclosed in square brackets and separated by commas. They are now all included in the array `My_images`.

- Why would you want all of your image variables under one name? You can refer to all images and use them at the same time with just the name of the array! Take a look at the following code:

```
Display.show(My_images, delay = 200)
```

- This would show all three images contained in `My_images` with a 200 millisecond delay between each one.

- Encourage students to put these concepts into practice with their own images and change the delay timer to see its effect on the speed of images.