Unit 6: micro:bit with Python | **Skill Builder 1: The display**

In this lesson, you will write your first python programs to control the micro:bit **display** in different ways. This lesson has two parts:

Part 1: alien encounter

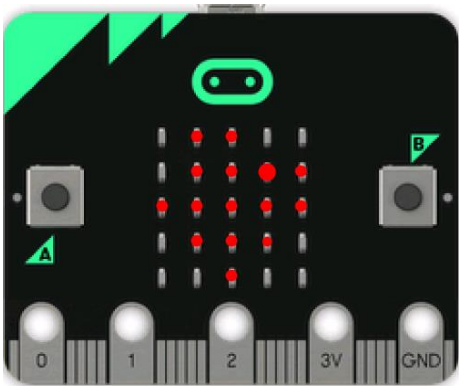
Part 2: displaying images

Objectives:

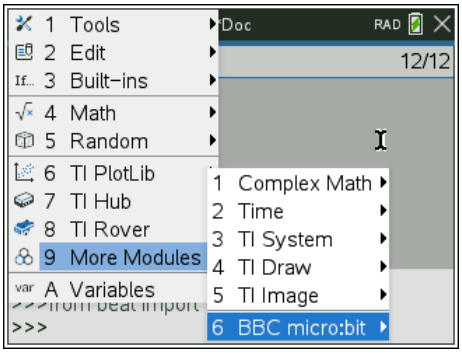
- Control the display on the micro:bit board using .**show**( ), .**scroll**( ) and .**show**(image)
- Control the speed of the display using **sleep**(*ms*)

1. Before you begin, be sure that:
   - your TI-Nspire CX II has OS **5.3** or higher
   - You are comfortable programming in python and/or you have completed Units 1 through 5.
   - your **micro:bit** is connected to your TI-Nspire CX II
   - You have followed the set-up directions and file transfers in the micro:bit **Getting Started Guide**:
     https://education.ti.com/en/teachers/microbit
     This setup process should only have to be done once, but keep informed periodically about updates/upgrades.

2. If all is good and the setup has been done correctly, your micro:bit should look like this when it has power from the TI-Nspire:
   The display on the **micro:bit** is showing the TI logo, an icon of the state of Texas with a bright spot near Dallas, the home of **Texas Instruments, Inc**.



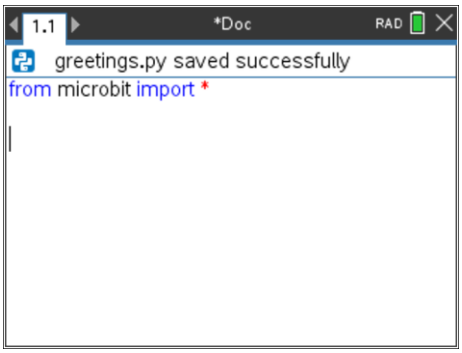
3. and finally…
   The micro:bit module is installed to your Python Library. In a Python Editor, press **[menu] > More Modules** and see that **BBC micro:bit** is listed beneath the TI modules.



Note: In OS 5.3 and above, python modules that are stored in your **Pylib** *folder on your device are shown on this menu in addition to the ti_ modules. Your list may differ from the one shown here. The modules are listed alphabetically by filemane, so* ***BBC micro:bit*** *is listed among the "m's" in the list, not the "B's".*

4. **Part 1: alien encounter:** Of course, as with every other first programming experience, you will start with displaying a message on the micro:bit display.
   Start a new TI-Nspire document and select **Add Python > New** to begin a new program (blank program), named '**greetings**'. In the Python Editor use **[menu] > More Modules > BBC micro:bit** to select the **import** statement at the top of the menu items:

   from microbit import *



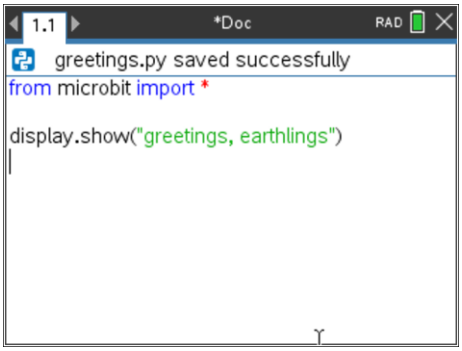
   Tip: If the message 'micro:bit not connected' ever appears, just unplug the micro:bit and plug it in again (reset).

5. To display a text message on the micro:bit display, use the statement:

   display.show(image or text**)**

   This statement is found on:

   [menu] > More Modules > BBC micro:bit > Display > Methods



The statement is inserted as **display.show(**image or text**)**, but (image or text) is just a placeholder that must be replaced with something. Inside the parentheses, replace image or text by typing your message string in quotes:

"greetings, earthlings"

When you run this program by pressing **[ctrl] [R]** you will see the letters of your message appear, one letter at a time, on the display. The lowercase letters 'e' do appear twice but you cannot distinguish two of them.

If you make a mistake…. go back to page 1.1 to edit your program, and then run the program again. Did you forget to add quotes around the text you wanted to display?

6. A better method for displaying messages is:

display.scroll(“greetings, earthlings”)

which is also found on

[menu] > More Modules > BBC micro:bit > Display > Methods

*To complete the **.scroll( )** statement you can copy/paste the string from the **.show** statement.*

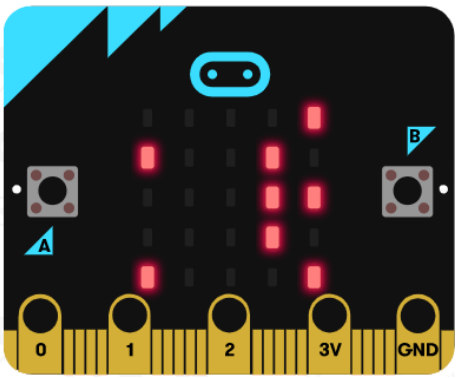
Make the previous **.show()** statement a #comment (place the cursor on that line and press **[ctrl] [T]**) to disable it and then run the program again.

Yes, you can also simply change .**show** *to* .**scroll** *by typing.*

7. The **.scroll()** method causes the message to scroll (move) from right to left like a banner across the display making it easier to read.
You can control the speed of the scrolling by adding the **delay=** parameter:

display.scroll(“greetings, earthlings”, delay = 200)

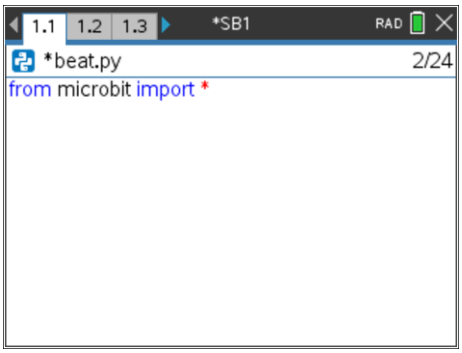
which uses a 200 millisecond (0.2 second) delay in the scrolling. Try other delay values, too.



<greetings, earthlings.gif>

8. **Part 2: Be.Still.My.Beating.Heart** press **[ctrl] [doc]** to insert a page and select **Add Python > New** to add a new Python program to your document (ours is named ‘**beat**’).
In the Python Editor, use **[menu] > More Modules > BBC micro:bit** and select the **import** statement at the top of the list:

from microbit import *



9. To display the HEART symbol on the micro:bit display, use the statement:

display.show( )

This statement is found on:

[menu] > More Modules > BBC micro:bit > Display > Methods

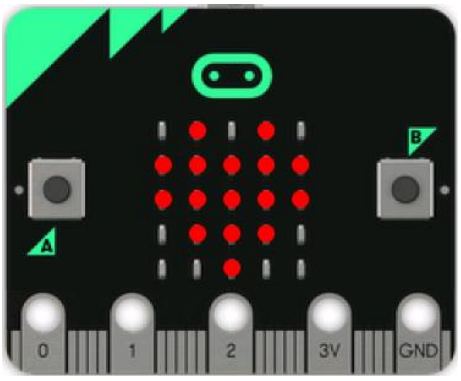
Inside the parentheses, replace the prompt by selecting:

Image.HEART

from

[menu] > More Modules > BBC micro:bit > Display > Images > Set 1> Heart

10. Run the program (press **[ctrl] [R]**) to see the HEART icon displayed on the 5x5 LED grid of the micro:bit. This display remains until something takes its place, even after the program is done.



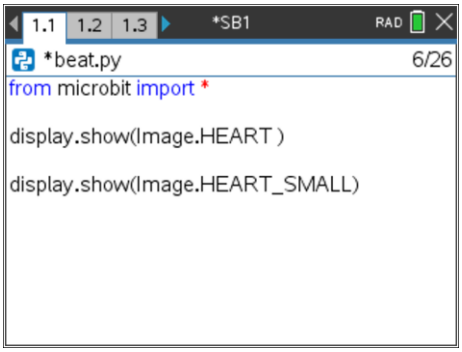
11. Go back to the Program Editor on the previous page, and add another display statement to show the small heart:

display.show(Image.HEART_SMALL)

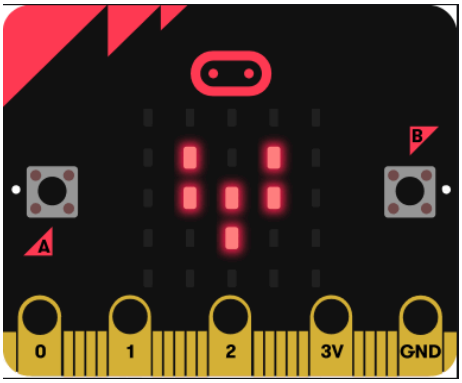
You can find this image on the same **Images** menu:

[menu] > More Modules > BBC micro:bit > Display > Images > Set 1

*Tip: you can also copy/paste the first display statement and edit (type the _***SMALL***). It must have the underscore _ and be uppercase.*



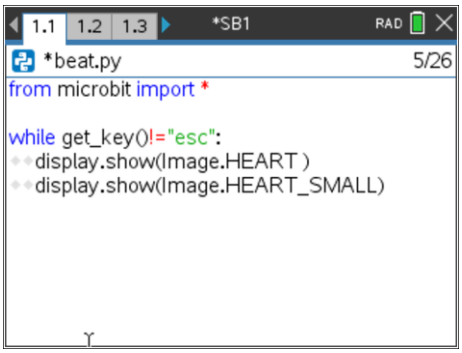
12. Run the program again. It quickly displays the large heart and then displays the small heart that looks like this.



13. **Make a loop**: To get the two hearts to blink repeatedly ('beat'), enclose the two display statements in a loop. *Before* the two display statements insert:

    while get_key() != "esc":

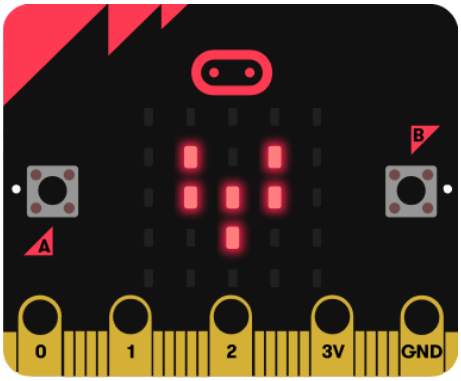
    found on **[menu] > More Modules > BBC micro:bit > Commands** and *indent* the two display statements so that they form the loop body.



    Important Tip: **Indentation** *is critical in python programs. This is how python interprets loop blocks and if blocks. If the two display statements are not indented the same number of spaces then you will see a syntax error. Use the* **[space]** *key or the* **[tab]** *key to indent the two lines the same amount. Indentation spaces are indicated in this Editor as light gray diamond symbols (♦♦) to help with proper indentation.*

14. Run your program again and watch the Beating Heart! Press the **[esc]** key to end the program.

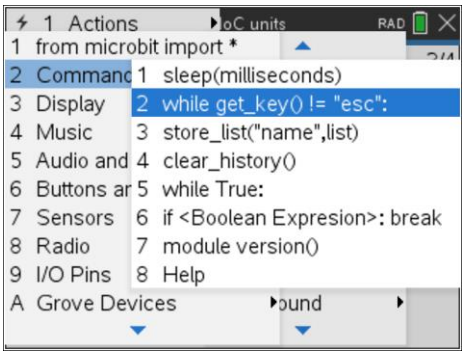
    Tip: if you ever think your program is stuck in an infinite loop press and hold the **[home/on]** key *on your TI-Nspire to 'break' the program. This could happen if you use* **while True:** *from the Commands menu improperly. These lessons avoid that type of structure.*



<beating_heart.gif>

15. *The micro:bit* **Commands** *menu contains some useful python commands that are also found on other menus. The micro:bit module imports these python commands for you.*

    You can these and any other python commands from other menus. You are not limited to just using the BBC micro:bit menu but you may need to provide the proper import commands.

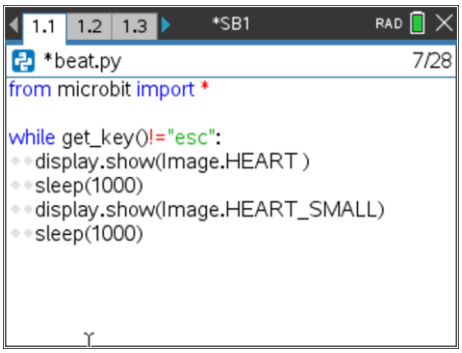


16. To control the beating heart rate, add two **sleep()** statements, one after each **display** statement:

    sleep(1000) means 1000 milliseconds or a 1 second delay.

    Also found on **[menu] > More Modules > BBC micro:bit > Commands**

    Tip: watch the indentation!



17. **Extension:** Try '**Making faces**'. Use a similar program structure as '**Beating Heart**' but use the face images found in Set 1 instead.