

Requirements:

- A. A computer that is running Apple Macintosh OS, or Windows OS (Operating System). The work can be done on a Linux machine, but I have no experience with this, so I cannot provide support. The software we will be using (the Mu editor) will not run on a Chromebook, or on smart phone or tablet. *You need administrator privilege on your computer to install Mu. If you install it without sufficient privilege, it will start and then close immediately, every time.*
- B. A CircuitPlayground Express and a USB cable to connect the CircuitPlayground to your computer.  
<https://www.adafruit.com/product/3333>

Workshop instructions:

1. Download and install the latest version of CircuitPython. Go to:  
<https://circuitpython.org/downloads>
2. From the menu, select the device you are using to get the CircuitPython UF2 file that will run on your device. In this case, select either CircuitPlayground Express, or CircuitPlayground Bluefruit. UF2 is a USB Flashing Format file, abbreviated UFF, or "U F squared", written and generally pronounced as UF2.
3. Click on the DOWNLOAD .UF2 NOW button.
4. Locate the downloaded .uf2 file on your computer, probably in your Downloads folder.
5. Connect the CircuitPlayground device to your computer with a USB cable.

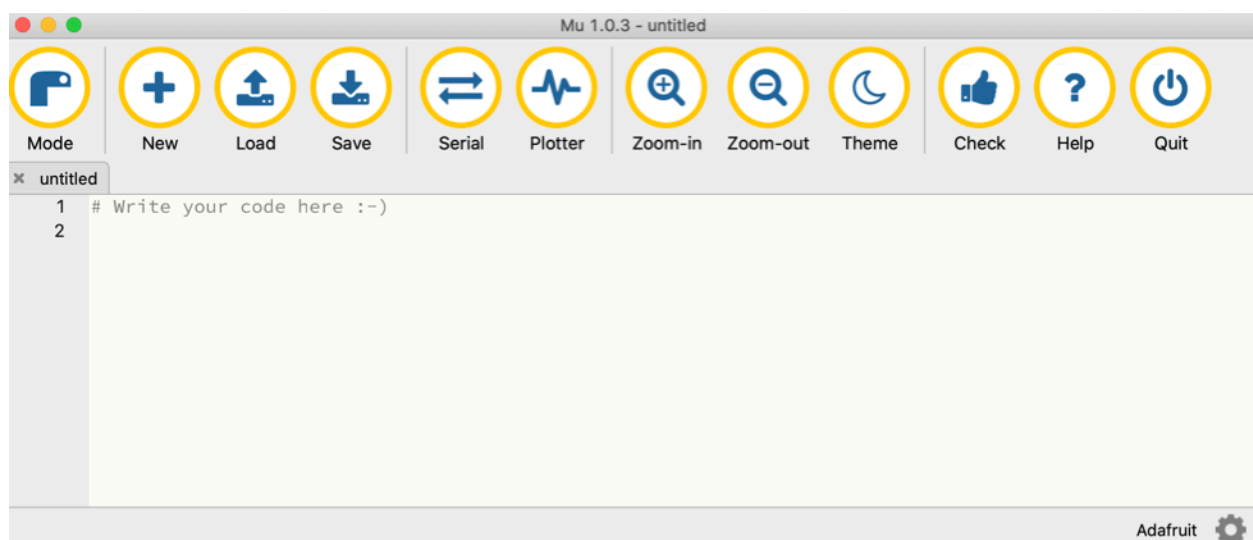
-- I've included two USB cables in the kit for this workshop, one for USB A type ports (older) and one for USB C type ports (newer).

-- If your computer asks if you want to set up a USB keyboard or mouse device, cancel that action.

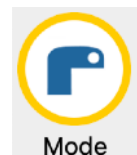
*CAUTION: Do not place the CircuitPlayground device in contact with the body of your computer. The exposed electrical connections on the CircuitPlayground could make contact with the metal of the computer, which may damage the CircuitPlayground, or damage the computer.*

-- A brand new Circuit Playground Express will light up a rainbow pattern on the lights in a sequence. This is normal behavior.

6. Find the Reset button on the center of the CircuitPlayground. Press it twice, briefly. It's a bit timing sensitive. All 10 lights should go momentarily red, and then change to green and stay green. If pressing it twice doesn't work, try pressing it just once. I know this is silly, but this is what the manufacturer suggests. In any case, you want 10 constant green lights to show. Those lights show that it is in BOOTLOADER mode, which allows us to put Circuit Python on it.
7. On your computer, there should now appear an external hard drive called CPLAYBOOT. Copy the .uf2 file, downloaded in step 3, to the CPLAYBOOT drive. The CPLAYBOOT drive will eject itself, to be replaced by a CIRCUITPY drive.
8. Download and install the Mu editor, from <https://codewith.mu>
9. Run the Mu editor. You may need to overcome a security restriction, and if you usually use a non-English language set, you will likely need to set your language settings to use English characters. Once running the Mu editor should look something like this:



10. Click on the Mode button, to see that the Mu editor is in Circuit Python mode, which it sometimes calls Adafruit Circuit Python mode. Click OK to close. (If it is not in Circuit Python mode, select that mode.)



11. Click on the Serial button. A Serial Dialogue Panel will open on the lower section of the Mu window, labelled Adafruit CircuitPython REPL. REPL is the Read, Evaluate, Print Loop. You can type individual commands in CircuitPython in this panel, and they will run on the CircuitPlayground device.

12. Click anywhere in the Serial Dialogue Panel to set the cursor in that panel.
13. Hold the ctrl key, and press the c key. We will call this ctrl-c from now on. Ctrl-c cancels whatever program is running. This is the same for Mac and Windows machines. *Do not use the Apple command key for this purpose.*
14. Press any key to enter the REPL mode. You will see a message like this, identifying which version of CircuitPython you are running, and what device you are running it on:  

```
Adafruit CircuitPython 5.3.1 on 2020-07-13; Adafruit  
CircuitPlayground Express with samd21g18  
>>>
```
15. If the Serial Dialogue Panel is unresponsive (which can happen if you press the reset button, unplug the Circuit Playground, or eject the CIRCUITPY drive), click on the Serial button to close the panel, which may be showing stale data from the previous connection. Click on the Serial button again to reopen the panel, to reestablish the connection.
16. Click the Save button. Save the file with the name **code.py** onto the CIRCUITPY device. (To access the Save As function, you must double click on the file tab. This process is not documented. If you save your code to your computer, it will not run on your CircuitPlayground, so you must double-click the filename, and Save As, selecting the CIRCUITPY drive.)
17. In the code area, where it says, `# Write your code here :-)`, write the following code:

```
print( 8 )
```

18. Then click the Save button to save the code and run it on the CircuitPlayground.
19. You should see this message appear in the Serial Dialogue Panel:

```
code.py output:  
8
```

20. Change the message to something else, to prove that you can. If you want to print characters instead of numbers, you must use quotation marks.

```
print( "hello" )
```

21. Let's program the device to print two things, with a time delay in between the two print statements.

```
import time
```

```
print( "hello" )  
time.sleep( 0.5 )  
print( "Paul" )
```

22. Let's make a program that prints those two things over and over and over as long as the CircuitPlayground has power.

```
import time
```

```
while True:  
    print( "hello" )  
    time.sleep( 0.5 )  
    print( "Paul" )
```

23. Python programs are Spaceholder Character Sensitive. To indent the three lines, I selected them all, and then pressed the Tab key. *To unindent, select some lines and press Shift-Tab.*

24. Cancel the program by clicking in the Serial Dialogue Panel, then press ctrl-c. Restart the program by pressing ctrl-d (which I think of as "decancel").

25. Blink an indicator

26. Fade brightness

27. Fade brightness over and over

28. Increase red, and use the fill method

29. Load the FancyLED library

30. Blink a color set by Hue, Saturation, Value

31. Sweep hue

32. Sweep hue collapsed steps

33. Fill one at a time

34. Sweep around the circle

35. Pattern play in groups:

You now have seen how to control Brightness, Hue, Position, and Timing, and have a hint on how to control Saturation.

What pattern would you like to play, by location, color, and time?

Try to make code that will play that pattern.