

Trinket M0 Using Circuit Python

—

TechAhoy Project

What is a Tomato Timer?

- Productivity Tool Used by Engineers
 - 25 minutes of work
 - 5 minutes of rest.

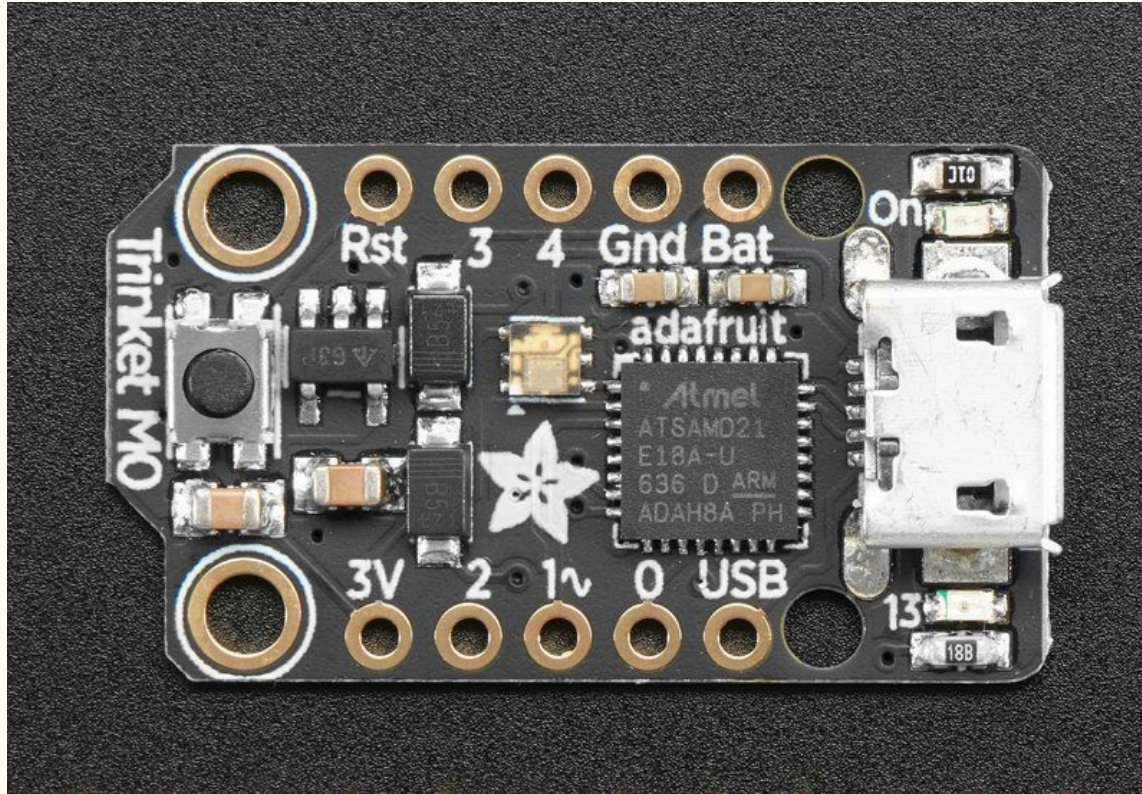


Materials

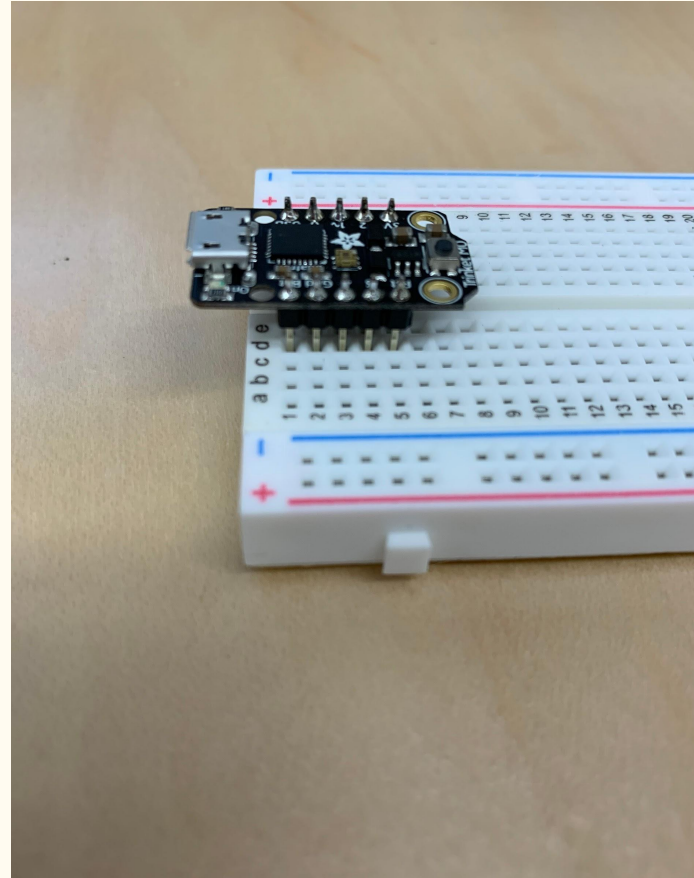
All available at TechAhoy!

- Trinket M0
- 5 LED Strip
- Breadboard (for prototyping)
- Header Pins
- Push Button Switch
- 2 Male Jumper Cables
- 3 Alligator Clips
- Micro USB Data Cable

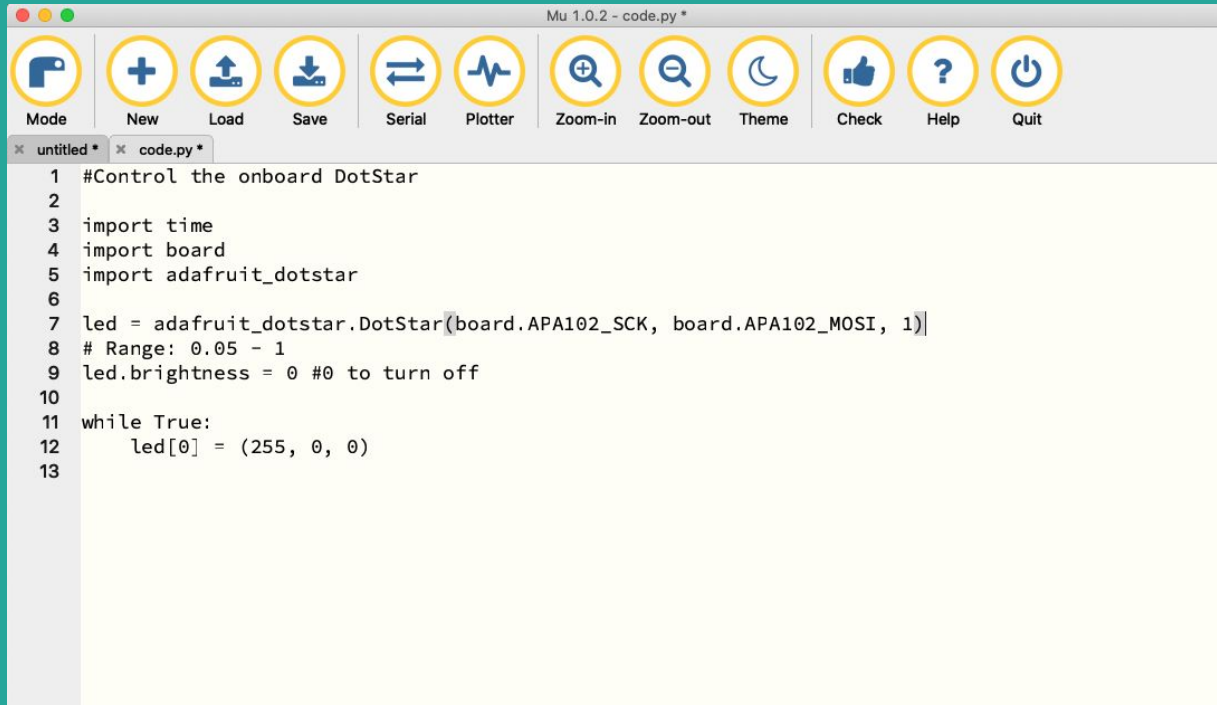
Trinket M0



Bread Board



Introduction to Software



The screenshot shows the Mu Python IDE window titled "Mu 1.0.2 - code.py *". The interface includes a toolbar with icons for Mode, New, Load, Save, Serial, Plotter, Zoom-in, Zoom-out, Theme, Check, Help, and Quit. Below the toolbar, there are two tabs: "untitled *" and "code.py *". The "code.py" tab is active, displaying the following Python code:

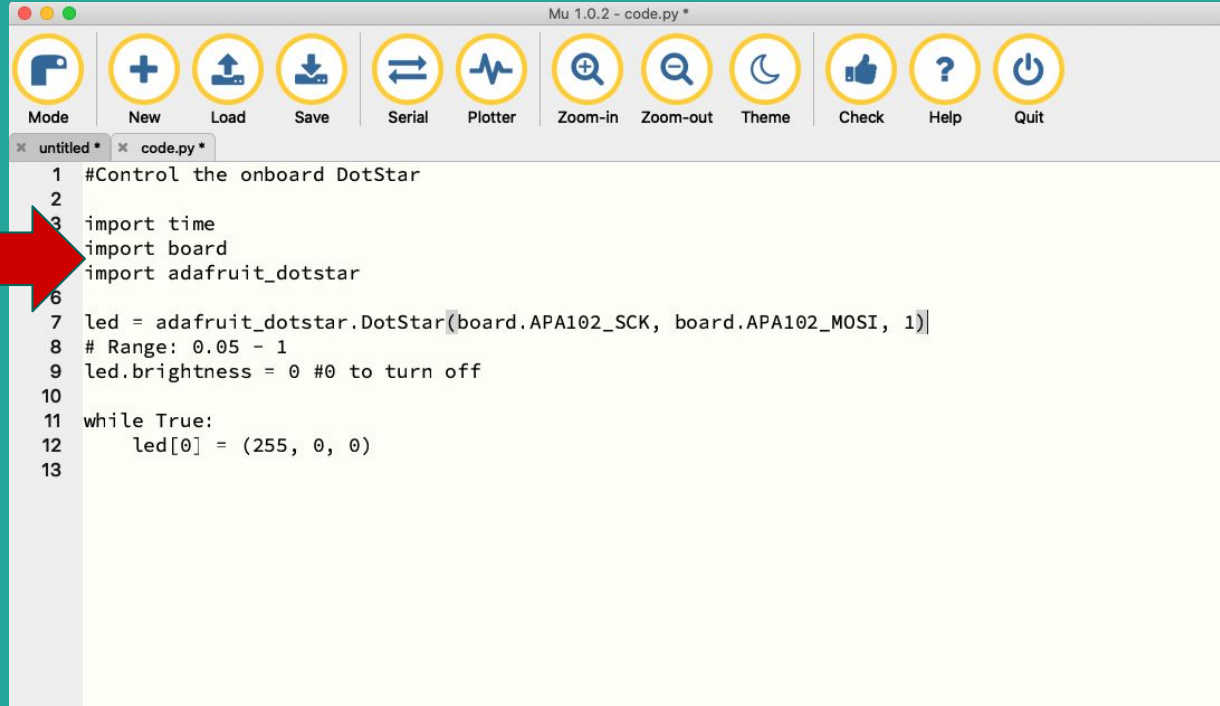
```
1 #Control the onboard DotStar
2
3 import time
4 import board
5 import adafruit_dotstar
6
7 led = adafruit_dotstar.DotStar(board.APA102_SCK, board.APA102_MOSI, 1)
8 # Range: 0.05 - 1
9 led.brightness = 0 #0 to turn off
10
11 while True:
12     led[0] = (255, 0, 0)
13
```

Turquoise: (19, 84, 78)

Coral: (100,50,31)

Fuchsia: (255, 0, 255)

IMPORTS

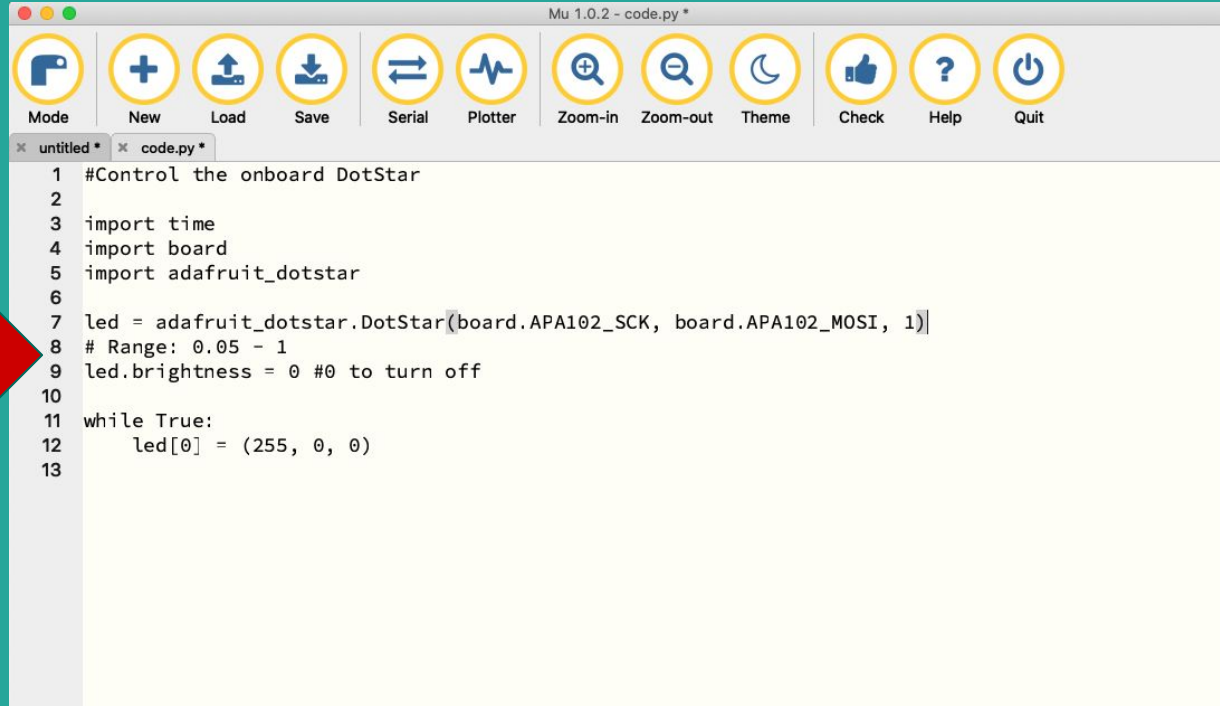


The screenshot shows the Mu Python IDE window titled "Mu 1.0.2 - code.py *". The interface includes a toolbar with icons for Mode, New, Load, Save, Serial, Plotter, Zoom-in, Zoom-out, Theme, Check, Help, and Quit. Below the toolbar, there are two tabs: "untitled *" and "code.py *". The "code.py *" tab is active, displaying the following Python code:

```
1 #Control the onboard DotStar
2
3 import time
4 import board
5 import adafruit_dotstar
6
7 led = adafruit_dotstar.DotStar(board.APA102_SCK, board.APA102_MOSI, 1)
8 # Range: 0.05 - 1
9 led.brightness = 0 #0 to turn off
10
11 while True:
12     led[0] = (255, 0, 0)
13
```

A large red arrow points to the 'import' statements on lines 3, 4, and 5 of the code.

SETUP

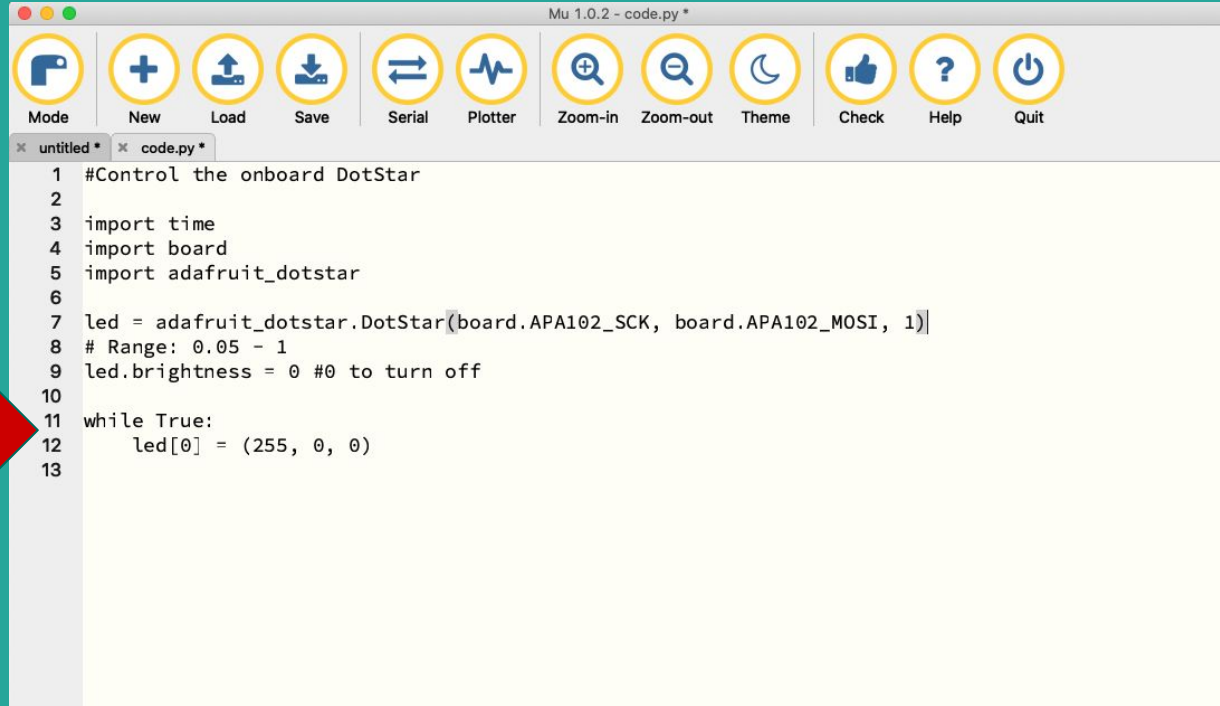


The screenshot shows the Mu Python IDE window titled "Mu 1.0.2 - code.py *". The interface includes a toolbar with icons for Mode, New, Load, Save, Serial, Plotter, Zoom-in, Zoom-out, Theme, Check, Help, and Quit. Below the toolbar, there are two tabs: "untitled" and "code.py". The "code.py" tab is active, displaying the following Python code:

```
1 #Control the onboard DotStar
2
3 import time
4 import board
5 import adafruit_dotstar
6
7 led = adafruit_dotstar.DotStar(board.APA102_SCK, board.APA102_MOSI, 1)
8 # Range: 0.05 - 1
9 led.brightness = 0 #0 to turn off
10
11 while True:
12     led[0] = (255, 0, 0)
13
```

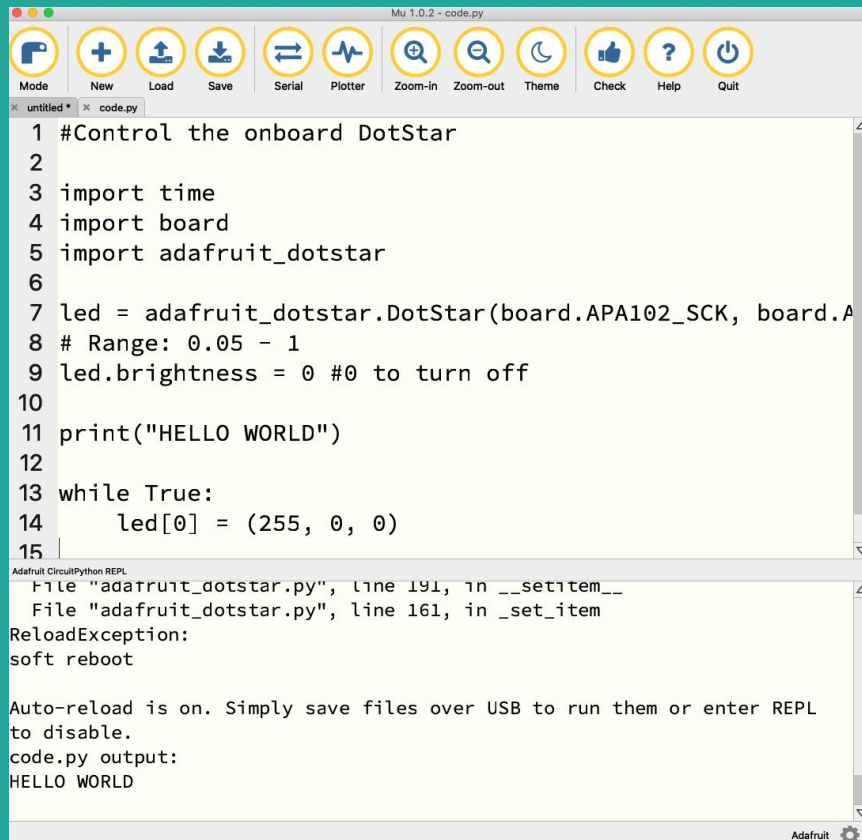
A large red arrow points to line 7 of the code, which is the line where the `led` object is instantiated.

THE LOOP



```
1 #Control the onboard DotStar
2
3 import time
4 import board
5 import adafruit_dotstar
6
7 led = adafruit_dotstar.DotStar(board.APA102_SCK, board.APA102_MOSI, 1)
8 # Range: 0.05 - 1
9 led.brightness = 0 #0 to turn off
10
11 while True:
12     led[0] = (255, 0, 0)
13
```

Hello World Program



The screenshot shows the Mu Python IDE interface. The top toolbar contains icons for Mode, New, Load, Save, Serial, Plotter, Zoom-in, Zoom-out, Theme, Check, Help, and Quit. The main editor window displays a Python script for controlling an Adafruit DotStar LED. The script imports the time module and the board and adafruit_dotstar modules. It initializes a DotStar object and sets its brightness to 0. It then prints "HELLO WORLD" and enters a while loop that sets the LED color to red (255, 0, 0) indefinitely. The bottom REPL window shows the execution output, including file reload messages and the printed "HELLO WORLD".

```
1 #Control the onboard DotStar
2
3 import time
4 import board
5 import adafruit_dotstar
6
7 led = adafruit_dotstar.DotStar(board.APA102_SCK, board.A
8 # Range: 0.05 - 1
9 led.brightness = 0 #0 to turn off
10
11 print("HELLO WORLD")
12
13 while True:
14     led[0] = (255, 0, 0)
15
```

Adafruit CircuitPython REPL

```
File "adafruit_dotstar.py", line 191, in __setitem__
File "adafruit_dotstar.py", line 161, in _set_item
ReloadException:
soft reboot

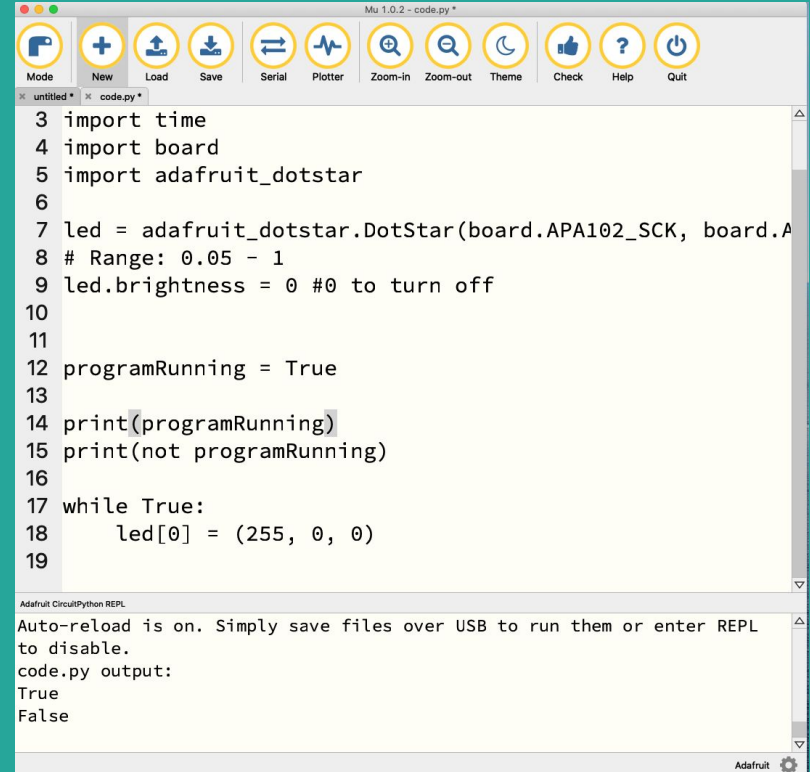
Auto-reload is on. Simply save files over USB to run them or enter REPL
to disable.
code.py output:
HELLO WORLD
```

Adafruit

Boolean Variables

Boolean variables can be thought of as switches:

- On/Off
- 1/0
- True/False



The screenshot shows the Mu Python IDE interface. The top toolbar includes icons for Mode, New, Load, Save, Serial, Plotter, Zoom-in, Zoom-out, Theme, Check, Help, and Quit. The main editor window displays a Python script for controlling an LED using the Adafruit DotStar library. The script imports the necessary modules, initializes the LED, and uses a boolean variable 'programRunning' to control the LED's state. The output of the script is shown in the REPL window at the bottom.

```
3 import time
4 import board
5 import adafruit_dotstar
6
7 led = adafruit_dotstar.DotStar(board.APA102_SCK, board.A
8 # Range: 0.05 - 1
9 led.brightness = 0 #0 to turn off
10
11
12 programRunning = True
13
14 print(programRunning)
15 print(not programRunning)
16
17 while True:
18     led[0] = (255, 0, 0)
19
```

Adafruit CircuitPython REPL

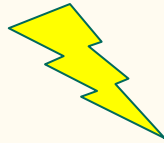
Auto-reload is on. Simply save files over USB to run them or enter REPL to disable.

code.py output:

```
True
False
```

Small Steps

When starting a new project, it is generally a good idea to start with the smallest problem.

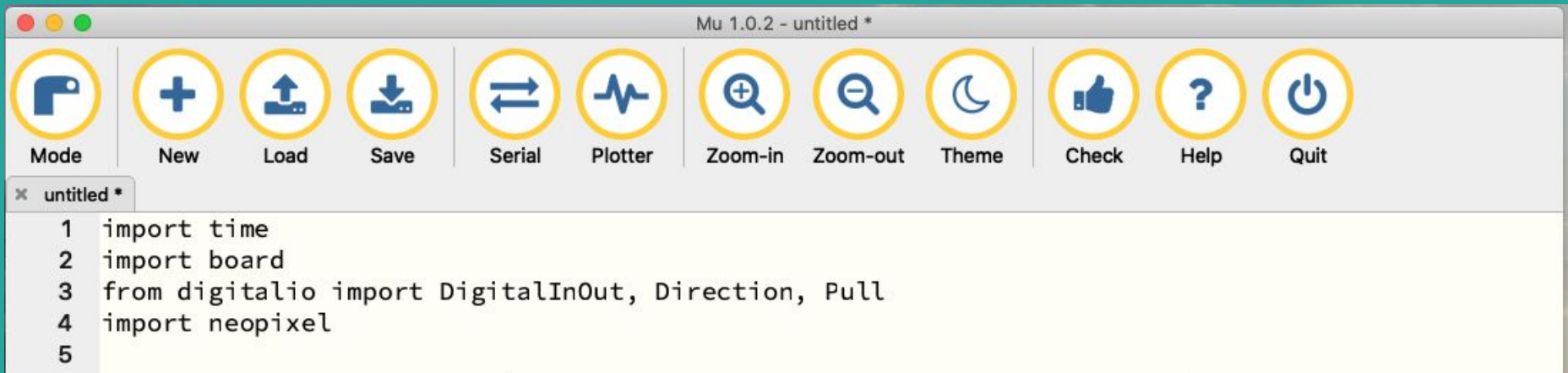


Challenge:
Blink Chip's DotStar
Using `time.sleep()`

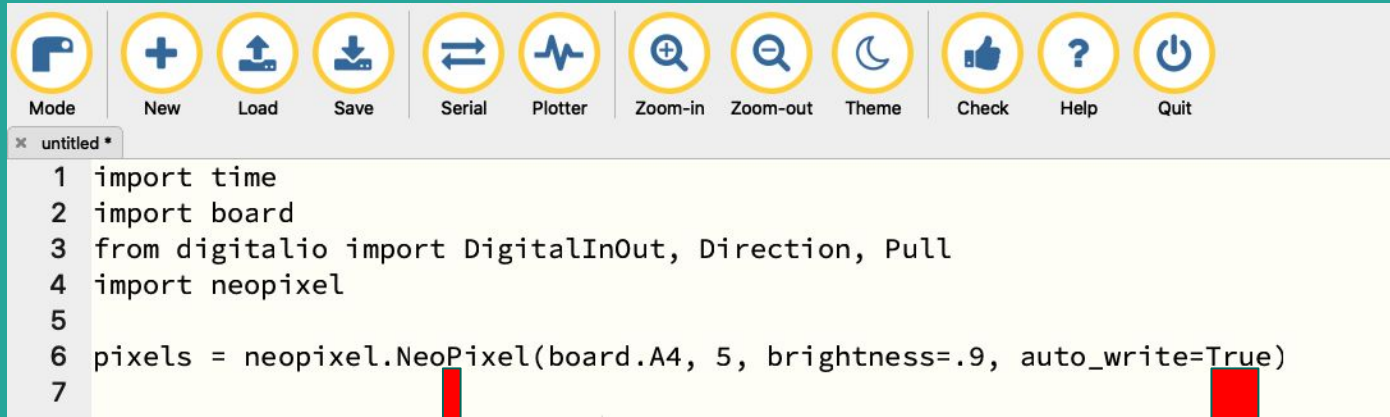
Engineering Problem

Program chip to light one LED every 5 mins. After 25 minutes, create a 5 minute resting pattern.

Important Imports



Hardware Setup



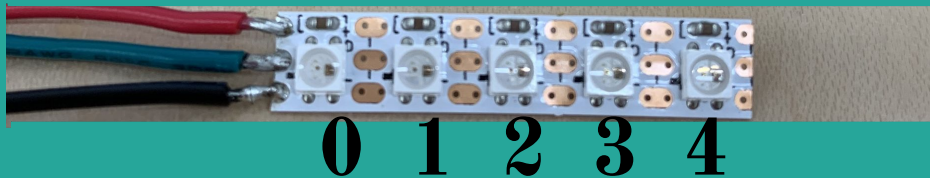
**Capitalization
Matters**

**Display colors
immediately**

WIRE THIS!

- 1) Attach alligator clips to the wires of the led strip
- 2) Connect Information Wire to Pin 4
- 3) Connect Power Wire to 3V
- 4) Connect Ground Wire to Gnd

An Array is a List of Data



`pixels[3] = (100,0,0)` makes the 4th LED red!



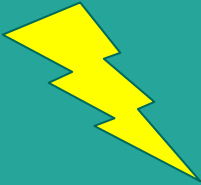
Challenge:
**Light all of the pixels in
the strip**

Introducing the **FOR LOOP**

Variable



```
14  
15 for pixel in range(0,5):  
16     print(pixel)  
17
```



Challenge:
Modify This Code to Light
Strip

Button Time

```
13 pixels = neopixel.NeoPixel(board.A4, 5, brightness=.1, auto_write=True)
14
15 button = DigitalInOut(board.A3)
16 button.direction = Direction.INPUT
17 button.pull = Pull.UP
18
```

WIRE THIS!

- 1) Have the button straddle the divider.
- 2) Connect top left leg to A3
- 3) Connect bottom right leg to Gnd

LOOP

```
13 pixels = neopixel.NeoPixel(board.A4, 5, brightness=.1, auto_write=True)
14
15 button = DigitalInOut(board.A3)
16 button.direction = Direction.INPUT
17 button.pull = Pull.UP
18
19 while True:
20     buttonUp = button.value
21     buttonDown = not buttonUp
22     print(buttonDown)
23     time.sleep(3)
24
```

Introducing the IF STATEMENT

```
19 while True:
20     buttonUp = button.value
21     buttonDown = not buttonUp
22     if buttonDown:
23         print(buttonDown)
24     time.sleep(.2)
25
26
```



CHALLENGE

Light the first led if
button is down.



CHALLENGE

Light all the pixels
when button is down.

CHALLENGE



Delay the lighting of
each led by 2 seconds.