

Welcome to CircuitPython!

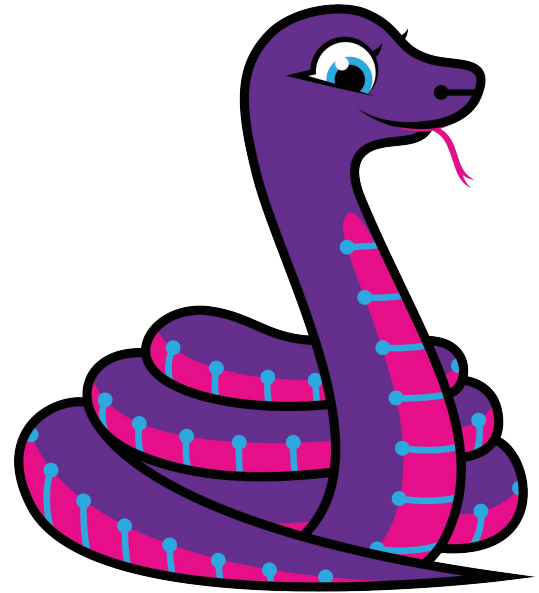
Kattni Rembor

Overview

- What is CircuitPython?
- Why would I use CircuitPython?
- The CircuitPython Community
- Getting Started
- Circuit Playground Express!

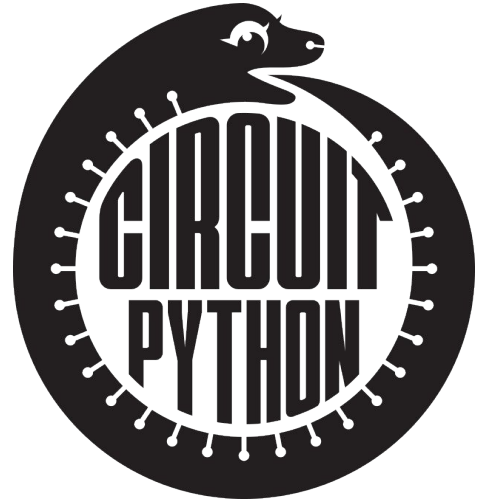
What is CircuitPython?

- Open Source
- Amazing, supportive community
- Runs on microcontrollers
- Higher level programming language
- Designed for learning
- Lowers the barrier for entry



Why would I use CircuitPython?

- New to programming
- Get your project up and running quickly
- Easily update your code - live!
- Serial console and REPL
- Solid hardware support
- It's Python!
- Open Source Software on Open Source Hardware



The Adafruit CircuitPython Community

- Everyone's welcome!
- Open Source Community
- Discord
- Forums
- GitHub

CODE + COMMUNITY = circuitpython

Content for the Workshop

- All of the content used in this workshop can be found at:

https://github.com/kattni/ChiPy_2018

Getting Started: Loading CircuitPython

- Download the latest version of CircuitPython - a **.uf2** file:
 - `adafruit-circuitpython-circuitplayground_express-3.1.1.uf2`
- Plug the board into a USB port on your computer.
- Double-tap the reset button found in the middle of the board - the ring of LEDs should turn green.
- A drive called **CPLAYBOOT** will appear in your file explorer.
- Drag the **.uf2** file you downloaded to the **CPLAYBOOT** drive.
- The drive will disconnect and a drive called **CIRCUITPY** will appear.

Getting Started: Downloading Mu

We'll be using Mu as our editor.

- On Mac or Windows, download the correct package:
 - <https://codewith.mu/en/download>
- On Linux, you must have Python 3 installed, and run:
 - `pip3 install mu-editor`
- On Linux, to install in a virtual environment:
 - `python3 -m venv .muenv && source .muenv/bin/activate`
 - `pip3 install mu-editor`

Getting Started: Serial Console Setup

Mu has the serial console built in. Open Mu to get started.

- Windows 7 and 8 require drivers - https://github.com/kattni/ChiPy_2018
- On Linux, your user must first be in the **dialout** group:
 - From the command line, run **nano /etc/group**
 - Find the line with **dialout:** and add your user to the end of the line, after a comma with no space, i.e. **dialout:foo,your_user**
 - Log out of your user and log back in.
- On Linux, to open Mu, from the command line run:
 - **mu-editor** or **python3 -m mu**

Getting Started: CircuitPython Basics

- Plug your board into USB.
- CIRCUITPY drive
- `code.py`
- Serial console and REPL

Getting Started: CircuitPython Code Basics

- `import`
- Setup
- Main loop:
 - `while True:`

Circuit Playground Express Library

- To use the CPX library, import `cpx` by typing the following at the beginning of your file:
 - `from adafruit_circuitplayground.express import cpx`

01. Blinky - The CircuitPython “Hello, world!”

```
import time
from adafruit_circuitplayground.express import cpx

while True:
    cpx.red_led = True
    time.sleep(0.5)
    cpx.red_led = False
    time.sleep(0.5)
```

Let's play with the plotter!

02. Light Sensor

```
import time
from adafruit_circuitplayground.express import cpx

while True:
    print("Light level:", cpx.light)
    print((cpx.light,))
    time.sleep(1)
```

03. Accelerometer

```
import time
from adafruit_circuitplayground.express import cpx

while True:
    x, y, z = cpx.acceleration
    print((x, y, z))
    time.sleep(0.5)
```


Let's take a look at some other inputs and sensors!

04. Button A

```
from adafruit_circuitplayground.express import cpx
```

```
while True:
```

```
    if cpx.button_a:
```

```
        cpx.red_led = True
```

05. Button B

```
from adafruit_circuitplayground.express import cpx
```

```
while True:
```

```
    if cpx.button_b:
```

```
        cpx.red_led = True
```

```
    else:
```

```
        cpx.red_led = False
```

06. Slide switch

```
from adafruit_circuitplayground.express import cpx

while True:
    # Left returns True. Right returns False.
    cpx.red_led = cpx.switch
```

07. Touch on A1

```
import time
from adafruit_circuitplayground.express import cpx

while True:
    if cpx.touch_A1:
        print("Touched A1!")
        time.sleep(0.1)
```

08. Touch on All Touch Pads

```
import time
from adafruit_circuitplayground.express import cpx

while True:
    if cpx.touch_A1:
        print("Touched A1!")
    if cpx.touch_A2:
        print("Touched A2!")
    if cpx.touch_A3:
        print("Touched A3!")
    ... # This means there's more code here!
    time.sleep(0.1)
```

09. Play Tone

```
from adafruit_circuitplayground.express import cpx
```

```
cpx.play_tone(262, 1)
```

```
cpx.play_tone(294, 1)
```

10. Start and Stop Tone

```
from adafruit_circuitplayground.express import cpx

while True:
    if cpx.button_a:
        cpx.start_tone(262)
    elif cpx.button_b:
        cpx.start_tone(294)
    else:
        cpx.stop_tone()
```


11. NeoPixel One

```
from adafruit_circuitplayground.express import cpx  
  
while True:  
    cpx.pixels[1] = (0, 50, 0)
```

12. All NeoPixels

```
from adafruit_circuitplayground.express import cpx
```

```
cpx.pixels.brightness = 0.3
```

```
while True:
```

```
    cpx.pixels.fill((255, 0, 0))
```

Now let's start combining the concepts we've learned!

13. Buttons and NeoPixels

```
from adafruit_circuitplayground.express import cpx
```

```
cpx.pixels.brightness = 0.3
```

```
while True:
```

```
    if cpx.button_a:
```

```
        cpx.pixels[0:5] = [(255, 0, 0)] * 5
```

```
    elif cpx.button_b:
```

```
        cpx.pixels[5:10] = [(0, 255, 0)] * 5
```

```
    else:
```

```
        cpx.pixels.fill((0, 0, 0))
```

14. Touch and Fill NeoPixels

```
import time
from adafruit_circuitplayground.express import cpx

cpx.pixels.brightness = 0.3

while True:
    if cpx.touch_A1:
        print("Touched A1!")
        cpx.pixels.fill((255, 0, 0))
    if cpx.touch_A2:
        print("Touched A2!")
        cpx.pixels.fill((210, 45, 0))
    ...
    time.sleep(0.1)
```

15. Light Up Touch Tone Piano!

```
while True:
    if cpx.switch:
        print("Slide switch off!")
        cpx.pixels.fill((0, 0, 0))
        cpx.stop_tone()
        continue
    if cpx.touch_A1:
        print("Touched 1!")
        cpx.pixels.fill((255, 0, 0))
        cpx.start_tone(262)
    elif cpx.touch_A2:
        ...
    else:
        cpx.stop_tone()
        cpx.pixels.fill((0, 0, 0))
```

Bonus CPX code included on GitHub

- Playing a wav file
- Shake
- Tap
- Rainbow Accelerometer
- Light Meter
- Sound Meter
- Potentiometer and servo (requires peripherals)

Thank You!

Find me:

kattni@adafruit.com

@kattni on Discord

<https://github.com/kattni/>

More Circuit Playground Express:

<https://adafru.it/adafruit-cpx>

<https://adafru.it/cp-on-cpx-made-easy>

More CircuitPython:

<https://adafru.it/cpy-welcome>

<https://adafru.it/discord>

<https://github.com/adafruit/circuitpython/>

<https://circuitpython.readthedocs.io>

