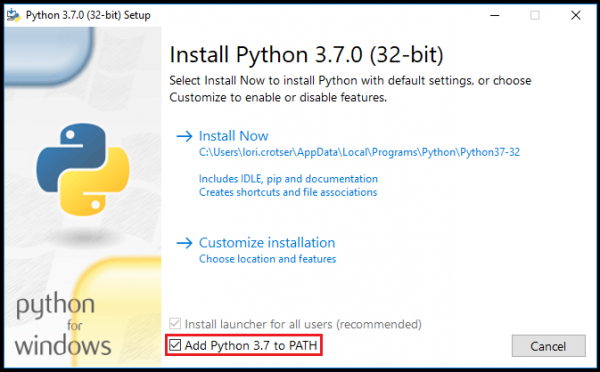
Install Python on Your Computer

In order to upload MicroPython programs to the ESP32, we will use a tool called [ampy](https://github.com/adafruit/ampy), which is a Python script that can read and write files to a MicroPython board over a serial connection. Because ampy is a Python script that runs on a host computer, we will need to install Python.

Instructions for installing Python can be found at [python.org](https://www.python.org/). Download and run the installer for your OS, or use your OS's package manager to install Python (if you are on Linux).

If you are installing on Windows, make sure the "Add Python <x.x> to PATH" checkbox is selected.

[](https://cdn.sparkfun.com/assets/learn_tutorials/6/5/9/AddPythonToPath.png)

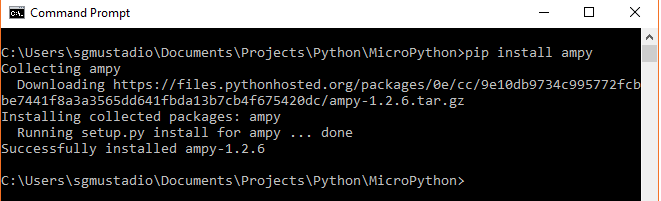
For more information on using Python on windows, see [this link](https://docs.python.org/3.3/using/windows.html).

Install ampy

Once you have installed Python, open a terminal for your host computer, and enter the following command:

COPY CODEpip install adafruit-ampy

You should see a printout showing that ampy was downloaded and installed.



REPL (Hello, World!)

One of the easiest ways to interact with MicroPython running on the ESP32 is to enter commands via the REPL over a serial connection. This can be extremely helpful if you want to test out various parts of code before writing them down in a (slightly more permanent) text document that is then uploaded to the ESP32.

To start, connect the ESP32 Thing to your computer using a [USB micro-B cable](https://www.sparkfun.com/products/10215). If you are on Windows, open the *Device Manager* and find out which COM port your ESP32 Thing is associated with (for example, COM7).

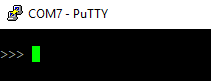
On macOS, it will likely be /dev/cu.usbserial-<some letters/numbers>,

On Linux, it’s probably something like /dev/ttyUSB0.

Open your Serial Terminal program of choice and connect to the ESP32 using the following connection details:

* **Speed**: 115200 bits per second
* **Data Bits**: 8
* **Parity**: None
* **Stop Bits**: 1

Once you have a connection, press enter once to get a REPL command prompt (>>>).



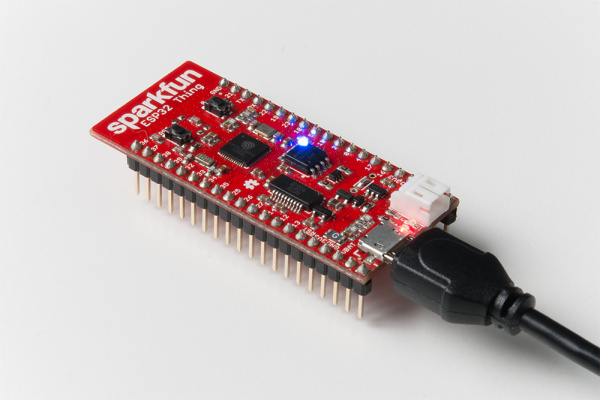
When you see that command prompt, enter the following commands one at a time:

COPY CODEimport machine

led = machine.Pin(5, machine.Pin.OUT)

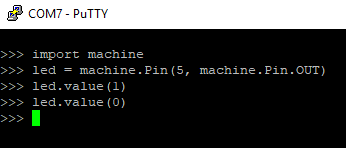
led.value(1)

This should turn on the LED found on the ESP32 Thing (the blue LED located by pin 5).

[](https://cdn.sparkfun.com/assets/learn_tutorials/6/5/9/MicroPython_Tutorial-06.jpg)

Type the following command to turn it off:

COPY CODEled.value(0)



Code to Note

The first thing we typed was import machine. [machine](http://docs.micropython.org/en/v1.7/esp8266/library/machine.html) is the MicroPython module that gives us control of the various general purpose input/output (GPIO) pins and special functions, such as I2C and interrupts. We use the keyword import to make all of the functions in the machine module available for us to use.

We then create a machine.Pin object and store it in the label led. When defining a machine.Pin object, we supply a number of arguments, including which pin number we want to use (the pin numbers can be found in white numbers on the side of the ESP32 Thing) and how we want to use that pin (e.g. as an input or output).

Because we set the pin as an output (as given by machine.Pin.OUT), we can drive the pin to logic low (0 V) or logic high (3.3 V) by setting its value to 0 (logic low) or 1 (logic high). To drive the pin high, we write led.value(1). This has the effect of turning on the LED.

Similarly, we can turn the LED off with led.value(0), which sets the pin to 0 V.