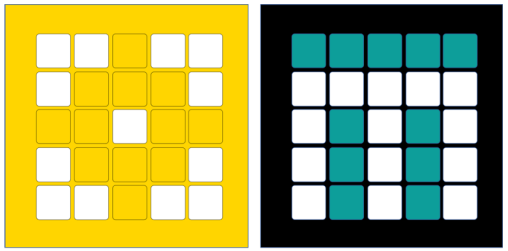
PRIME LESSONS

By the Makers of EV3Lessons



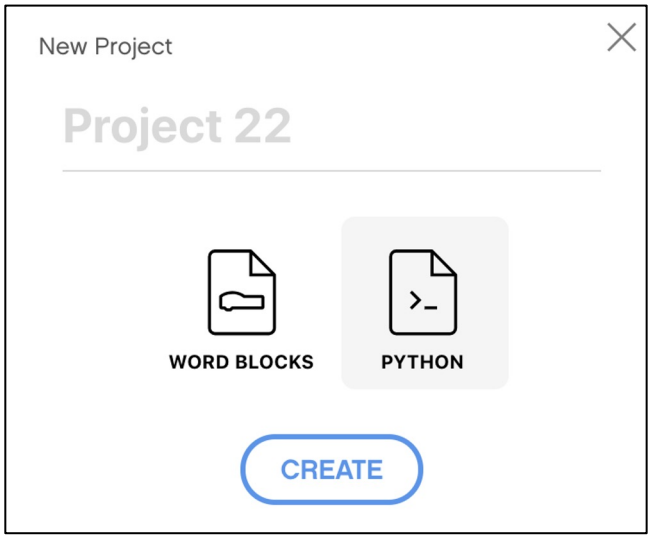
INTRODUCTION TO HUB & SOFTWARE (PYTHON)

BY SANJAY AND ARVIND SESHAN

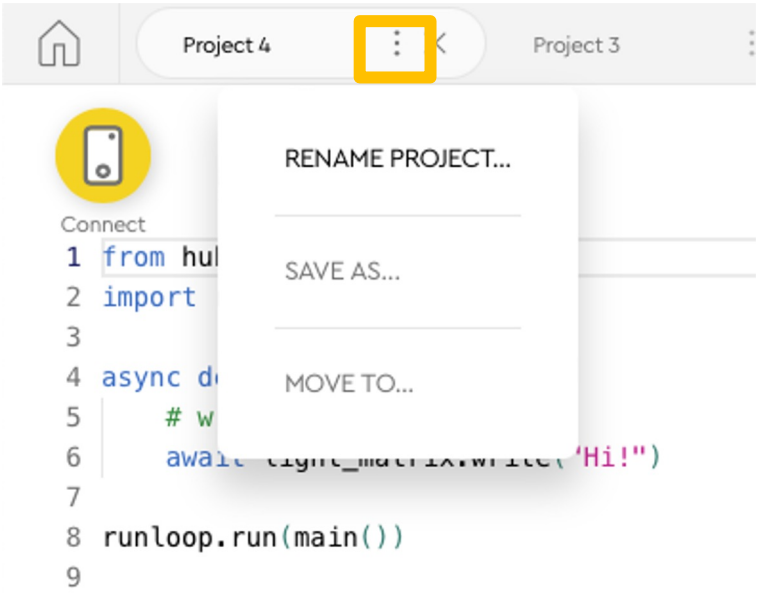
This lesson uses SPIKE 3 software

CREATING A PYTHON PROJECT

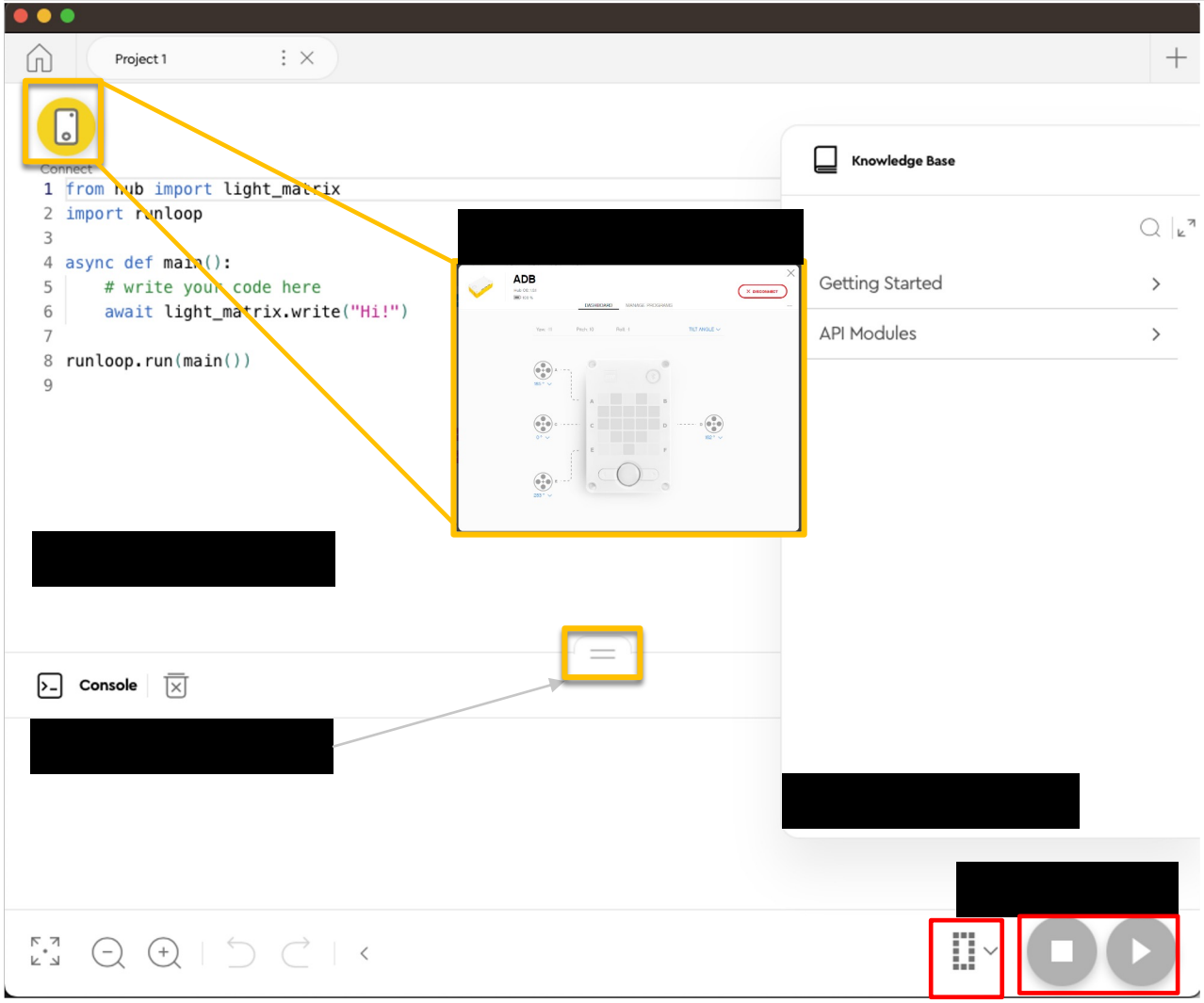
Select Python from the pop-up



Click on the three dots to change the file name



PROGRAMMING CANVAS



Knowledge Base: MicroPython Tips

Programming Canvas: The main programming canvas is where you will create each program (called 'Project')

The Connect Icon lets you access the Hub Dashboard

Stop/Play Icon lets you pick which slot to download the code to and run your code

Console: Anything you print as well as errors will show up here

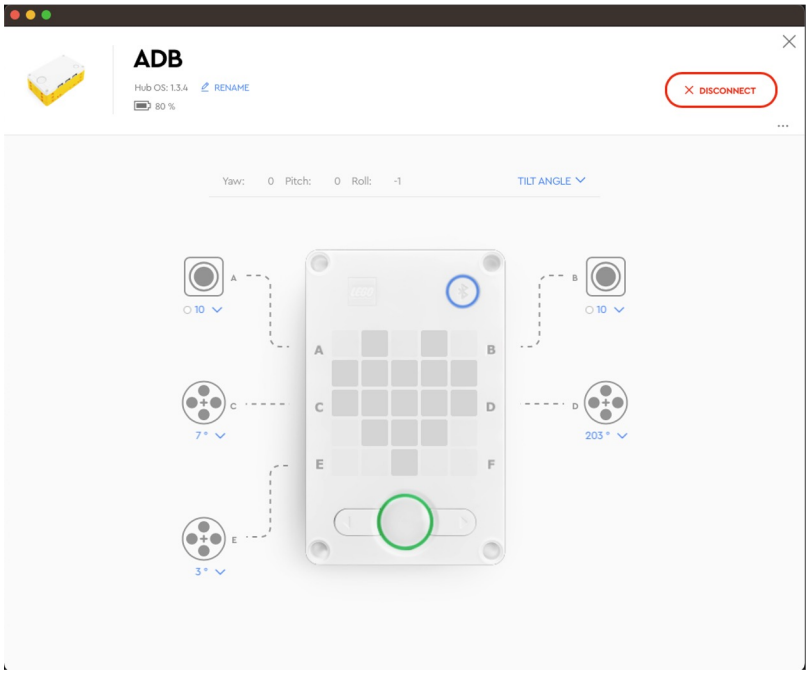
HUB DASHBOARD

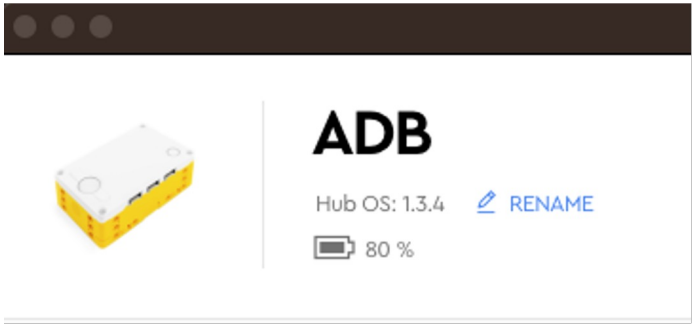
You must connect your Hub to access this section

This section is very useful for:

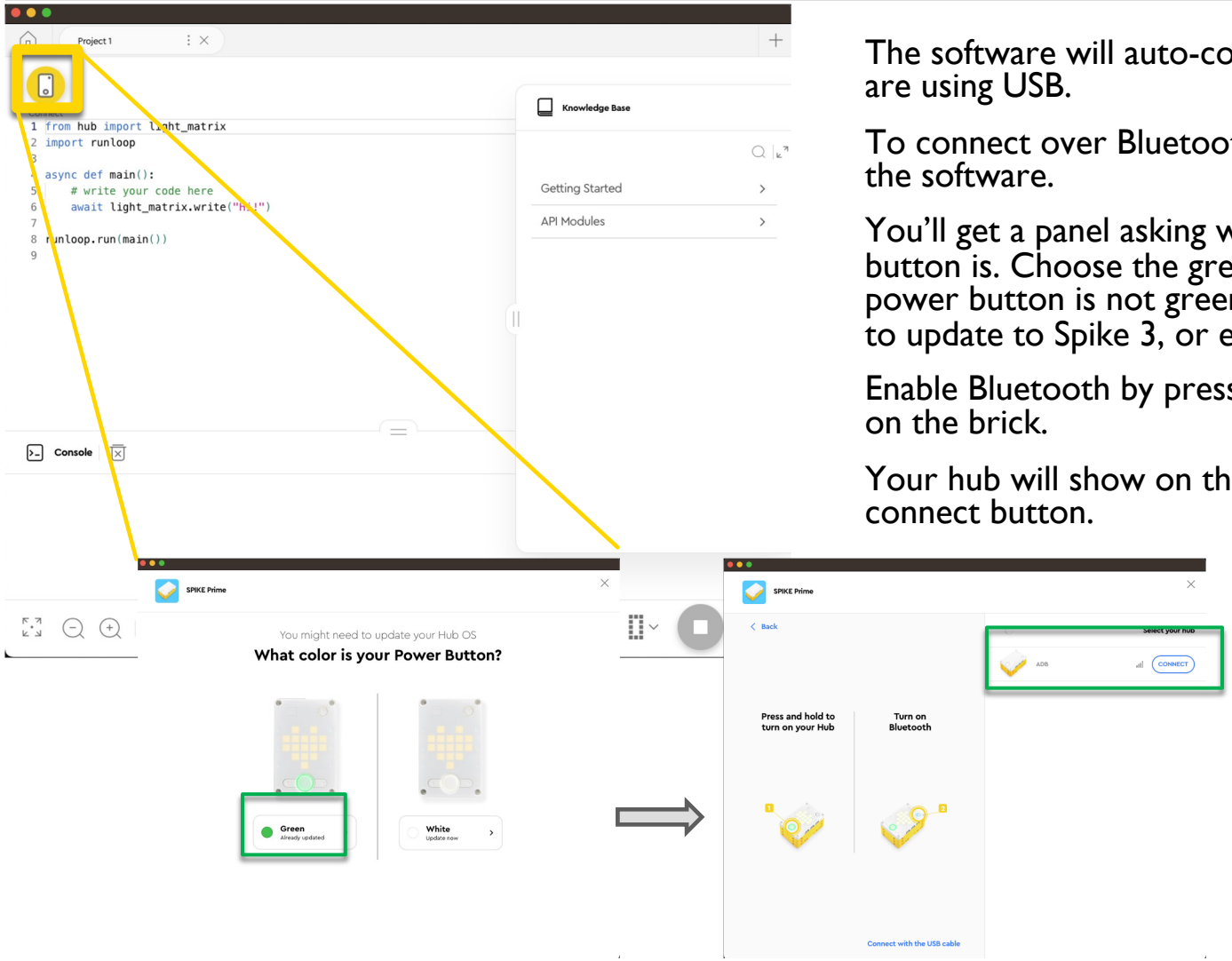
- Checking battery level
- Hub OS version
- Gyro Sensor Values
- See which motors and sensors are connected
- Get real time values from the motors and sensors

You can also rename your Hub in this panel by clicking on the blue RENAME link next to the Hub OS version number





CONNECTING TO BRICK



The software will auto-connect to the brick if you are using USB.

To connect over Bluetooth, click the connect icon in the software.

You'll get a panel asking what color your power button is. Choose the green power button. If your power button is not green, choose White if you want to update to Spike 3, or exit the app.

Enable Bluetooth by pressing the Bluetooth button on the brick.

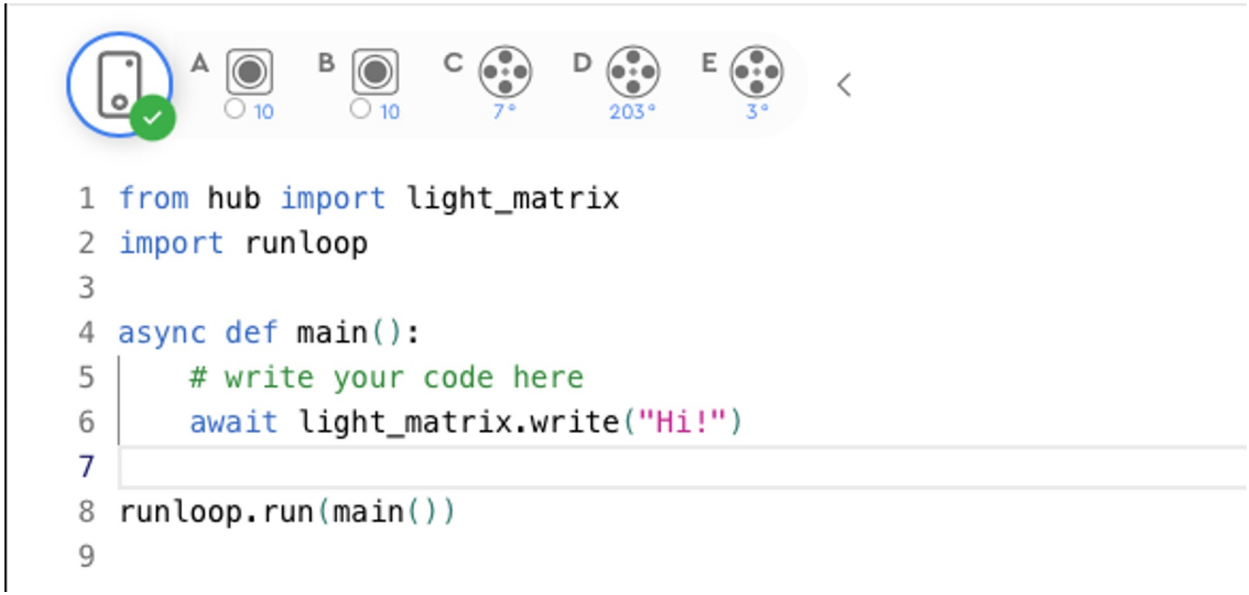
Your hub will show on the right side. Click the connect button.

DEFAULT PYTHON CODE

All Python programs will begin with the following code by default

You will typically keep lines 2, 4 and 8: The runloop import, the main function, and the call to main.

You can add more imports to allow you to use the ports/sensors/motors/etc. in your programs



SPIKE 3 - ASYNC AND AWAIT

Spike 3 has introduced functionality to run coroutines using async/await. The Knowledge Base has good information on it.

These are very useful to run multiple functions concurrently without waiting for any to finish, if so desired

To wait for an async function to finish, simply call it with an await:

```
await myAsyncFunction()
doSomethingAfterWaiting()# will run when above function completes
```

To run a built-in async function without waiting, simply call it without waiting:

```
motor.run_for_degrees(port.A, 360, 200)
```

To run custom async function(s) without waiting, call it using the runloop:

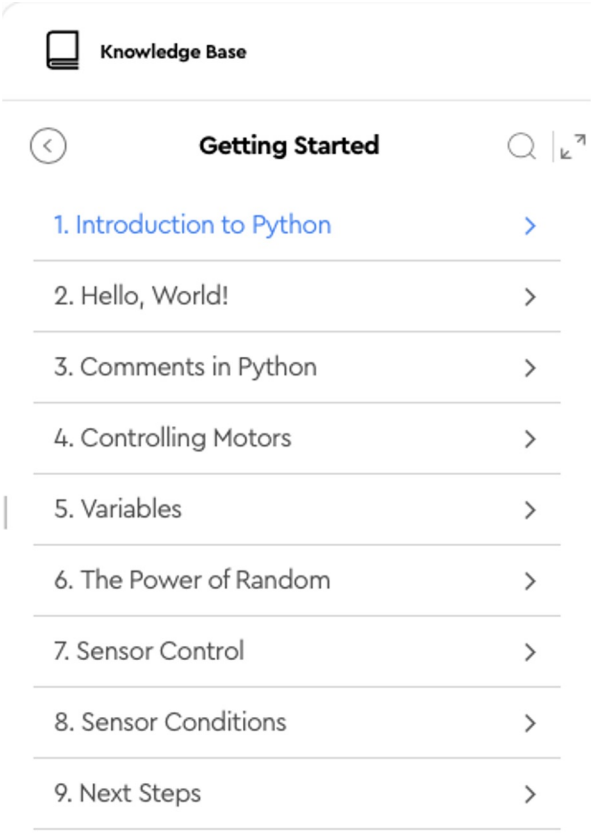
```
# create two async function calls to send to the runloop
a = myAsyncFunction(<parameters>)
b = mySecondAsyncFunction(<parameters>)
# run both the functions together
runloop.run(*[a,b])
```

More examples are available in the Functions Lesson

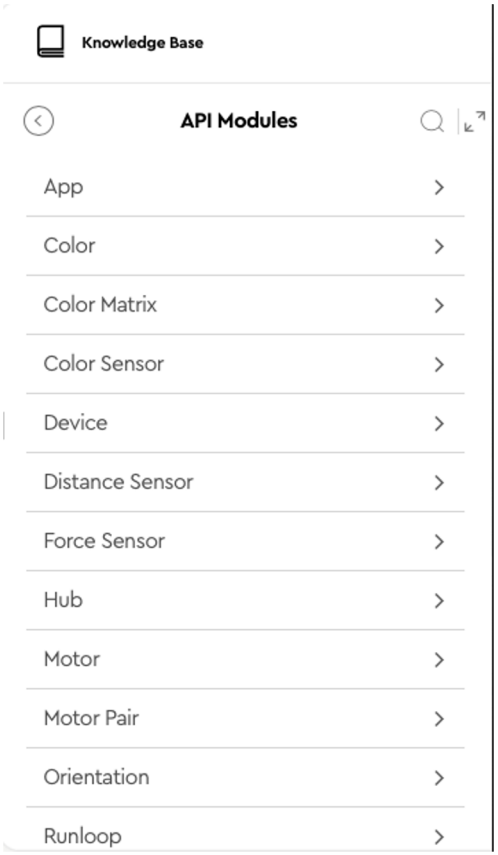
SPIKE 3 PYTHON RESOURCES

The Knowledge base has two sections you should explore:

Getting Started – great if you are new to Python API



API Modules – details on Spike 3



OTHER PYTHON RESOURCES

Note: No endorsement is implied.

W3Schools is a good, free resource for getting started with Python coding online. It has its own Python editor so you do not need to install Python on your machine. Here are some useful sections to get started if you are new to Python:

- Python Intro
- Python Syntax
- Python Comments
- Python Variables
- Python Numbers
- Python Booleans
- Python If...Else
- Python While Loops
- Python For Loops
- Python Functions
- Python Scope
- Python Math

CREDITS

This lesson was created by Sanjay and Arvind Seshan for Prime Lessons

Additional contributions by FLL Share & Learn community members.

More lessons are available at www.primelessons.org