An Introduction to Python

What is Python

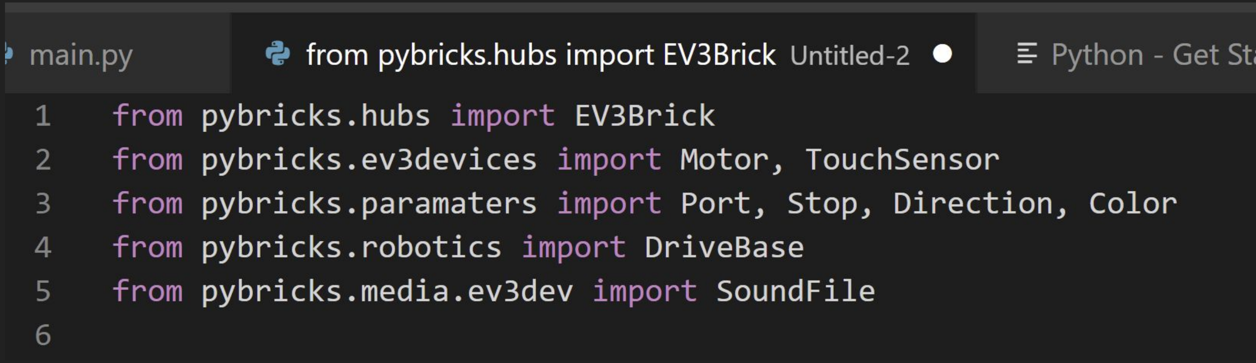
- Programing language
- Often used as a beginner-friendly code because commands similar to english
- Used in many fields- including physics
- We will use micropython
  - Usage of Python 3 that uses reduced amount of commands and libraries to run on constrained environment and microcontrollers

So What Will We Use?

- Import modules
- If statements
- Loops
- Variables
- Lists
- Micropython for EV3 commands (next week)

Modules

- Blocks of code that can serve a function written by someone else that we can use in our code
- Saves time and energy when coding
- However, Python doesn’t automatically know to use them (if it had every module ever it would be really slow)
- Therefore we need to import them into the program
  - From library mport module a

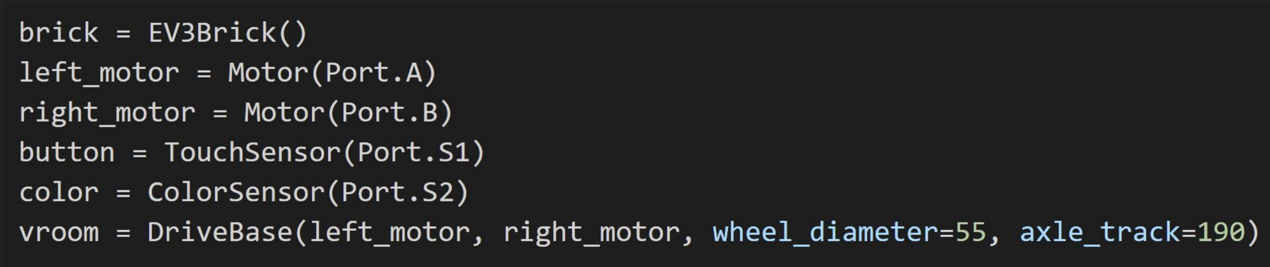


```
from pybricks.hubs import EV3Brick
from pybricks.ev3devices import Motor, TouchSensor
from pybricks.paramaters import Port, Stop, Direction, Color
from pybricks.robotics import DriveBase
from pybricks.media.ev3dev import SoundFile

```

Variables

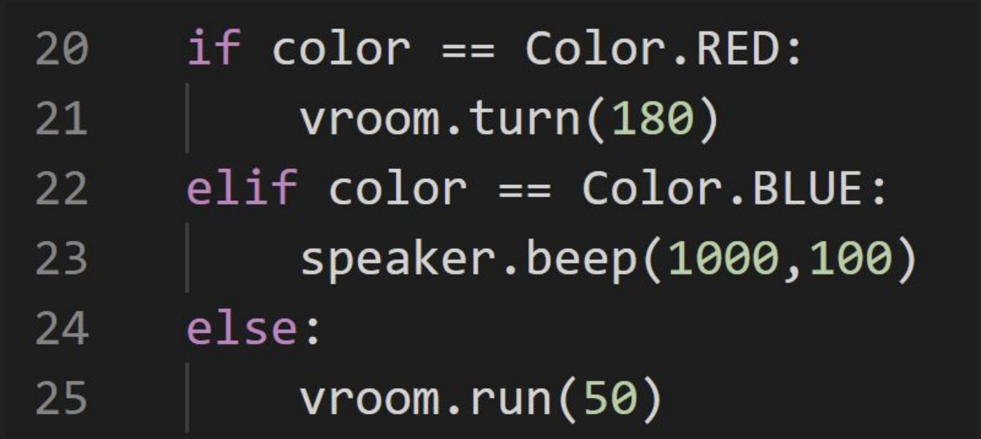
- “Nickname” for things in code
- Useful for things that change during the code
  - Such as an input that is sent (ex. the color the color sensor sees)
- Also useful for things that need establishing parameters that take a long time to write so they don’t need to be reestablished
- Allows you to use that information later in the program



```
brick = EV3Brick()
left_motor = Motor(Port.A)
right_motor = Motor(Port.B)
button = TouchSensor(Port.S1)
color = ColorSensor(Port.S2)
vroom = DriveBase(left_motor, right_motor, wheel_diameter=55, axle_track=190)
```

If Statements

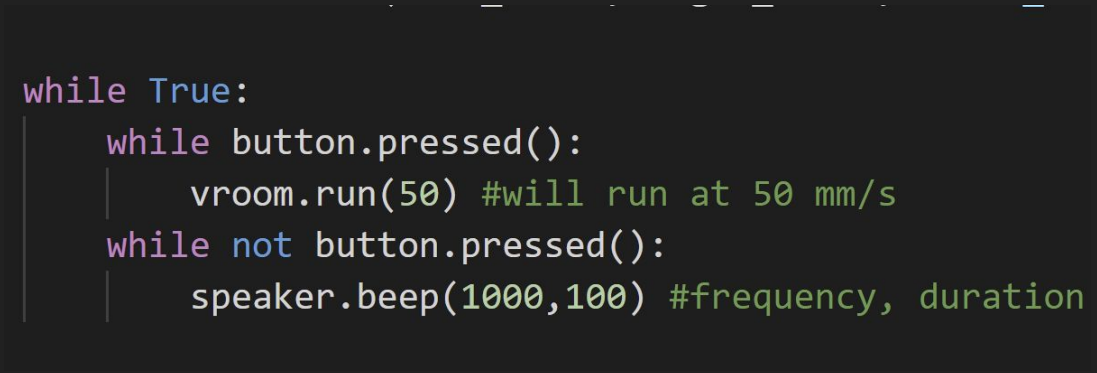
- Create a parameter that **if** it is met, a certain code will run
  - Indent code you want to run in that scenario under the if statement
- If you want to set another parameter, use **elif** (else if)
- Use **else** to run code that does not meet any of the parameters set in the above if and elif statements



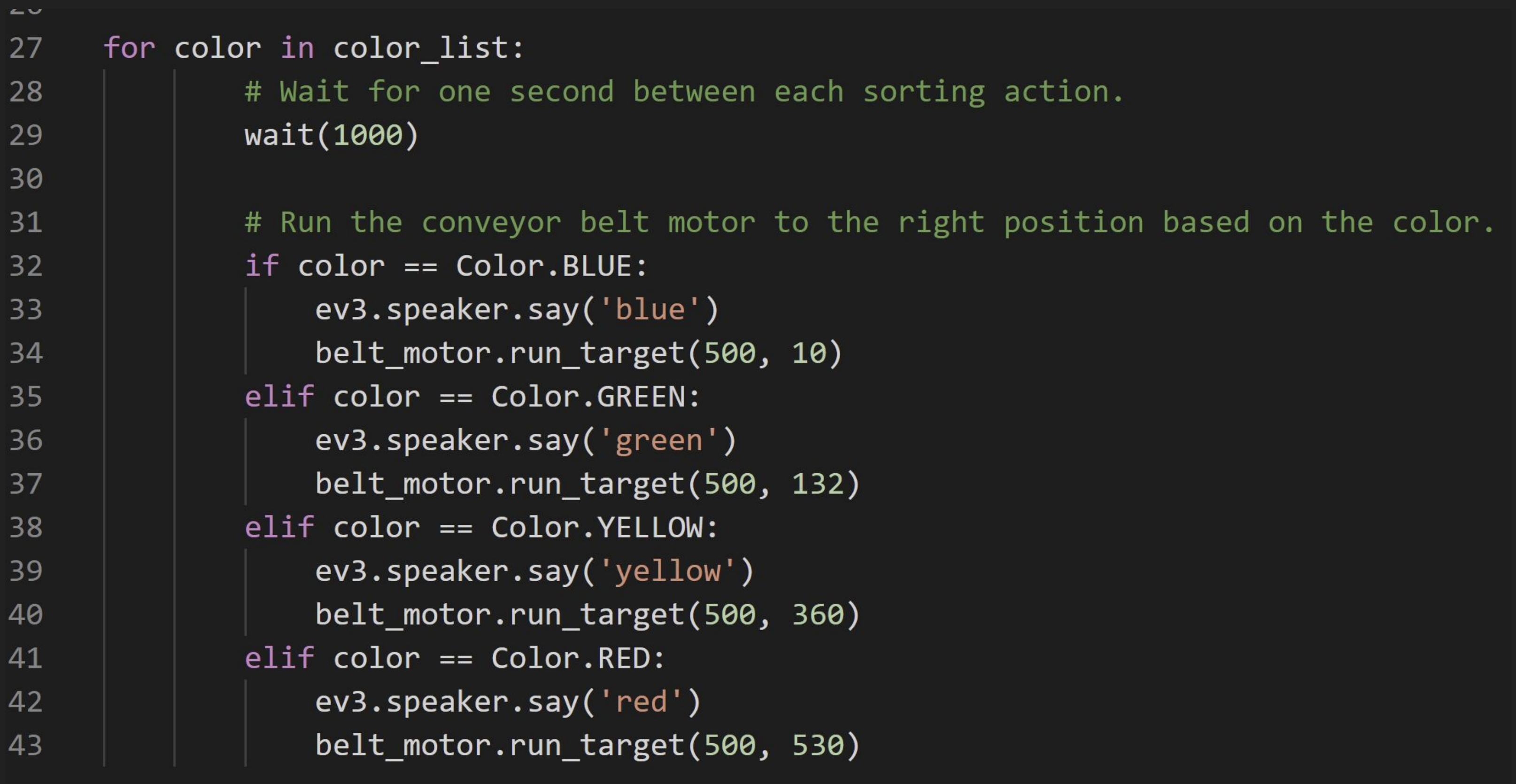
```
if color == Color.RED:
    vroom.turn(180)
elif color == Color.BLUE:
    speaker.beep(1000,100)
else:
    vroom.run(50)
```

Loops

- Set a scenario where when that scenario is met the program indented under it will be repeated
- Will repeat a specific amount of times or until a situation is met that tells it to stop
- Two main types of loops we will be using, for loops and while loops
- While loops have a parameter where the code will loop as long as that parameter is met
- For loops set a repeat amount using a parameter



```
while True:
    while button.pressed():
        vroom.run(50) #will run at 50 mm/s
    while not button.pressed():
        speaker.beep(1000,100) #frequency, duration
```



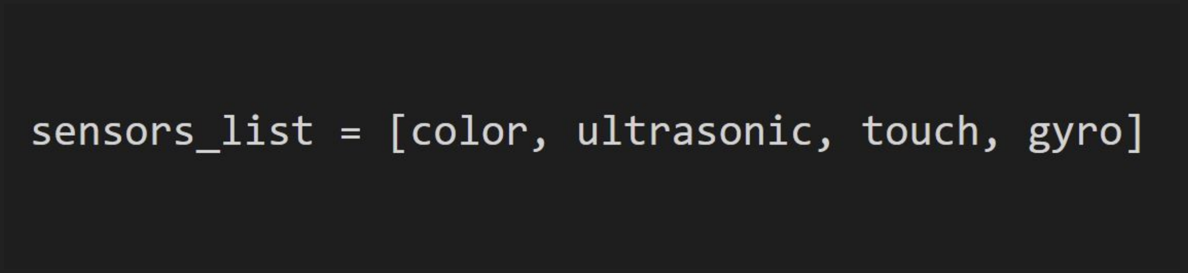
```
for color in color_list:
    # Wait for one second between each sorting action.
    wait(1000)

    # Run the conveyor belt motor to the right position based on the color.
    if color == Color.BLUE:
      ev3.speaker.say('blue')
      belt_motor.run_target(500, 10)
    elif color == Color.GREEN:
      ev3.speaker.say('green')
      belt_motor.run_target(500, 132)
    elif color == Color.YELLOW:
      ev3.speaker.say('yellow')
      belt_motor.run_target(500, 360)
    elif color == Color.RED:
      ev3.speaker.say('red')
      belt_motor.run_target(500, 530)
```

For loop with if statements

Lists

- Create a group of items
- Good for combining similar data to use together
- Can be any data type
- Add an item to your list with append
  - listname.append(‘item”)



```
sensors_list = [color, ultrasonic, touch, gyro]
```