

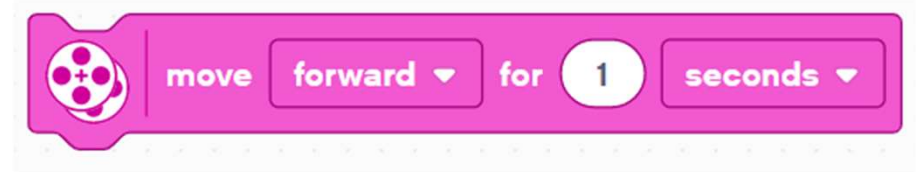
# Basics of EV3 Micropython

Time to play with something new!

# You will be typing! No more blocks

- But everything we did with the blocks can be reproduced by typing

```
robot.straight(500)
```



```
#this drives the robot forward for 500mm
```

# Intro Stuff

Unlike with the blocks, when programming there is often some setup code that helps us get up and running. Notice the “Python” tab in the simulator...

```
#!/usr/bin/env python3
# Import the necessary libraries
import math
import time
from pybricks.ev3devices import *
from pybricks.parameters import *
from pybricks.robotics import *
from pybricks.tools import wait
from pybricks.hubs import EV3Brick

ev3 = EV3Brick()
motorA = Motor(Port.A)
motorB = Motor(Port.B)
left_motor = motorA
right_motor = motorB
robot = DriveBase(left_motor, right_motor, wheel_diameter=56, axle_track=152)
robot.settings(straight_speed=200, straight_acceleration=100, turn_rate=100)

# Here is where your code starts
robot.straight(200)
```

All of this is setup! It helps the code know how the robot is put together, how it's wired, and what devices are available. We don't need to worry about changing it for now (but we need it!)

Where your actual code will start! The only thing we're actually telling our robot to do is “robot.straight(200)” – which means, go straight 200mm

# So what can you do?

- There are many commands. Some worth investigating are...

`robot.turn(<number>)`

*Makes the robot do a spin turn <number> degrees clockwise*

ex: `robot.turn(90)` OR `robot.turn(-60)`

`robot.straight(<number>)`

*Makes the robot drive forward <number> millimetres*

ex: `robot.turn(1000)` OR `robot.turn(-250)`

# Commands continued...

```
robot.drive(<drive_speed>,<amount_of_turn>)
```

*Makes the robot continually drive.*

*<drive\_speed> says how fast (-100 to 100)*

*<amount\_of\_turn> says if straight or turning (-300 to 300)*

*300 is a spin*

*150 is a pivot*

*0 is straight*

ex: `robot.drive(50,300)`

SPIN TO THE RIGHT AT  
SPEED 50mm/s

OR `robot.drive(100,0)`

DRIVE STRAIGHT AT  
SPEED 100mm/s

# Things to watch out for

- All the text needs to be lined up! Python is really particular about it.

```
robot.straight(200)
robot.turn(-90)
robot.straight(200)
robot.turn(-90)
```

← This is going to throw an error

```
robot.straight(200)
robot.turn(-90)
robot.straight(200)
robot.turn(-90)
```

← Python is happy now

# Same missions!

1. Straight forward 600mm
2. Straight forward 600mm, then backwards 600mm
3. Experiment with implementing the three kinds of turns (SPIN, PIVOT, CURVE)
4. Make your robot drive a square (60cm on one side, any type of turn of your choice)
5. Make your robot drive the same square, but this time do three in a row, using one kind of turn for each square
6. Repeat the simulated maze (see Friday's instructions)

- But Mr. Emmell! Where can I see more commands?

[pybricks.com/ev3-micropython/index.html](http://pybricks.com/ev3-micropython/index.html)

# But Mr. Emmell – give me more to do!

- Try this code, and change it to experiment to discover what it means

```
robot.drive(100,0)
while(True):
    if (obstacle_sensor.distance() < 50):
        robot.turn(180)
        robot.drive(200,0)
```

Careful! Watchout!

- Keep the spacing and tabbing as shown above
- Careful to match capitalizations (True vs true)
- Don't forget the colons! “ : ”