

Pi-Top Workshop #1

Introduction to Programming
Pi-Tops in block code

<https://further.pi-top.com/>

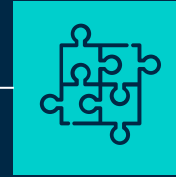
Workshop #1 Agenda



01

BLOCK CODE

Learn to code
Pi-Tops with block
code.



02

MAZE CHALLENGE

Program the
Pi-Tops to
maneuver through
a maze.

Connecting to the Pi-Top

Connect to Further

Create a link between the rover and Further.

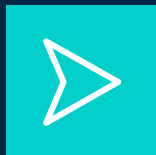
- **Find your pi-top's IP address.** Press the down arrow on top of the pi-top until you see a number on the mini screen.
- Look at the bottom right area of this Further lesson, **find the pi-top symbol**. **Click on it.**
- **Enter the IP address** in the box. The periods must be entered too.
- **Click connect.** The words should change from connect to disconnect.

☐ Mark section as complete



BASIC MANEUVERS

Forward &
Reverse



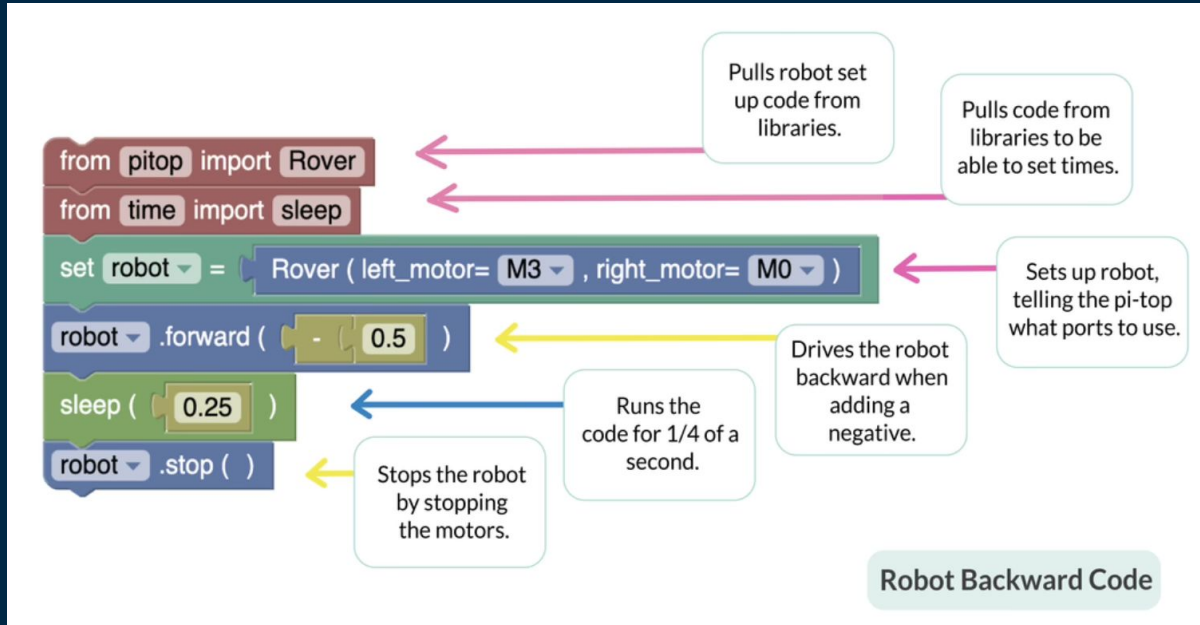
Turning Left &
Right



Changing Speed

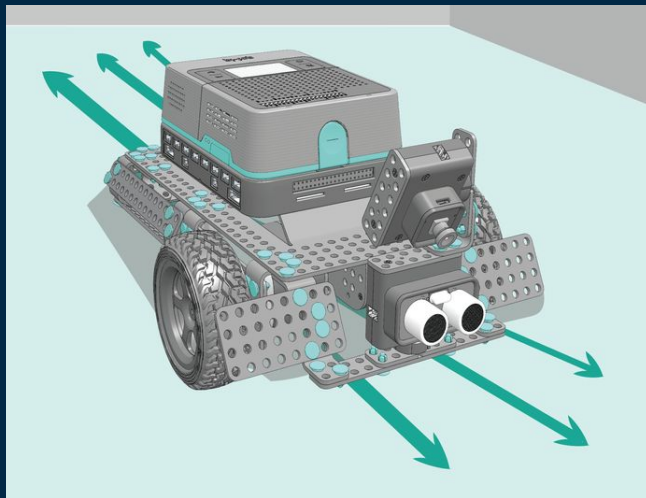


REVERSE

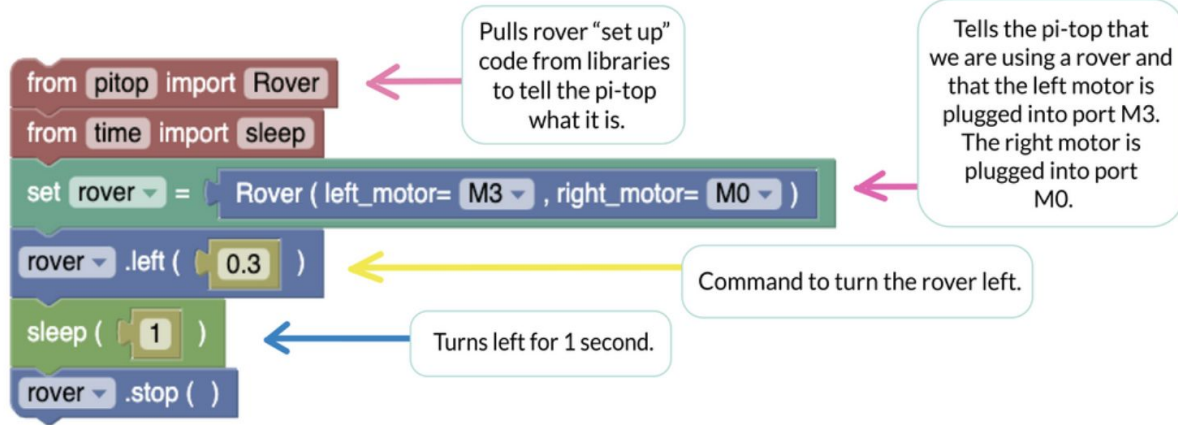


DISTANCE TEST

1. Set the variable for sleep to (1)
2. Run the code and measure the distance that the robot moves
3. Repeat the process, measuring the distances when setting the sleep variable to (0.5) and (2)
4. Record the distances for each variable



TURNING RIGHT



Left Turn Code

TURNING LEFT

```
from pitop import Rover
```

```
from time import sleep
```

```
set robot = Rover ( left_motor= M3 , right_motor= M0 )
```

```
robot .forward ( - 0.5 )
```

```
sleep ( 0.25 )
```

```
robot .stop ( )
```

Pulls robot set up code from libraries.

Pulls code from libraries to be able to set times.

Sets up robot, telling the pi-top what ports to use.

Drives the robot backward when adding a negative.

Runs the code for 1/4 of a second.

Stops the robot by stopping the motors.

Robot Backward Code

CODING THE BLINKERS

The blinkers will blink in the direction that the robot is turning.

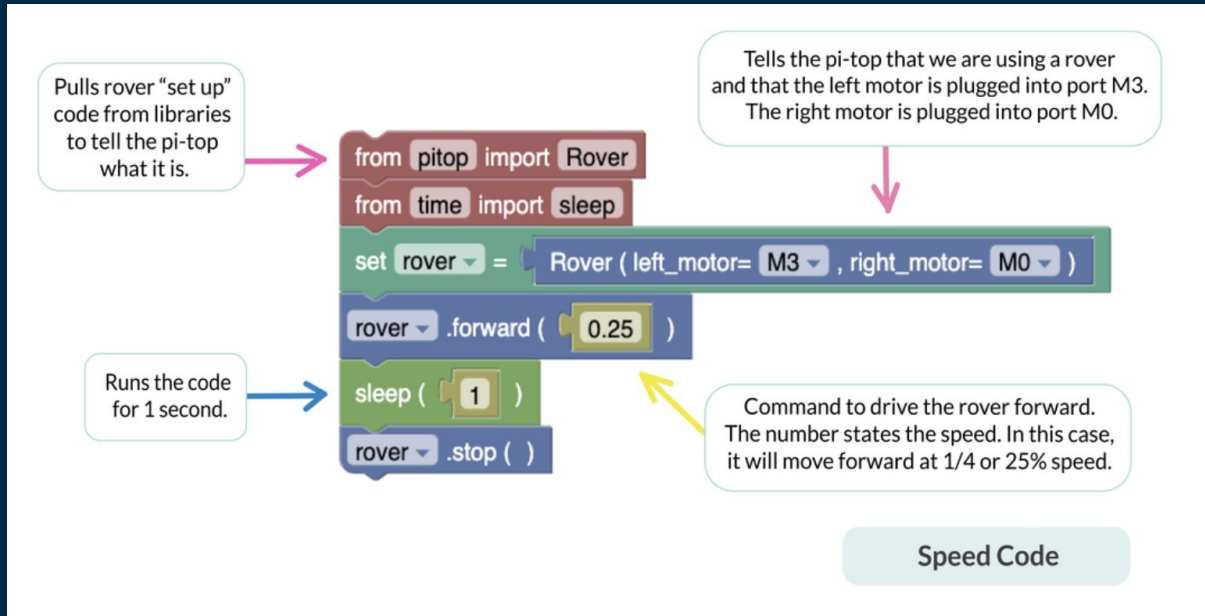
```
from pitop import Rover
from time import sleep
from pitop.pma import LED

set rover = Rover ( left_motor= M3 , right_motor= M0 )

set green1 = LED D6

set green2 = LED D2
```

CHANGING SPEED >>>



- Speed is changed by changing the value for “rover forward (...)”

The background is a dark blue field filled with various geometric elements. There are numerous small squares in solid colors (teal, orange, pink) and some as thin white outlines. Long, thin white vertical lines of varying lengths are scattered across the composition, some intersecting with the colored squares. The overall aesthetic is modern and minimalist.

MAZE CHALLENGE

MAZE CHALLENGE

OBJECTIVE: Program the robot to maneuver through a maze. Apply knowledge of basic maneuvers and modifying variables.

