**Pi2Go Programming: Python Statements and Pi2Go Commands**



**AIM:** After completing this worksheet you should be able to control your Pi2Go robot using simple statements issued at the Python Command Line and be able to explain what a statement in a programming language is.

**You Need:** To complete this worksheet you need to have a Pi2Go that is connected to a keyboard, mouse and monitor (see WS1) and to understand how to start and stop IDLE from the Linux Command Line (see WS2).

**Set up your robot and start IDLE.**

**Remember: You can scroll back through commands in IDLE by typing *ctrl-p***

In order to control your robot, you need to *initialise it properly.* To initialise your robot, type the following at the Python Command Line

>> import pi2go

>> pi2go.init()

Now you can use simple commands to control your robot. Try typing the following:

>> pi2go.forward(20)

>> pi2go.stop()



What happens?

Each of these commands pi2go.init(), pi2go.forward(20), pi2go.stop() is a *python statement.* Statements are the basic commands that are used to build up programs. You have several commands available to you for operating the Pi2Go robot.

These commands are described in the box on the next page. Not that parts in *italics* are inputs to the commands which you have to select. So, for **pi2go.spinLeft(***speed***)** you have to replace *speed* with a number between 0 and 100.

**pi2go.stop()**

**pi2go.forward(***speed***)** where *speed* is a number and 0 <= *speed* <= 100

**pi2go.reverse(***speed***)** where *speed* is a number and 0 <= *speed* <= 100

**pi2go.spinLeft(***speed***)** where *speed* is a number and 0 <= *speed* <= 100

**pi2go.spinRight(***speed***)** where *speed* is a number and 0 <= *speed* <= 100

**pi2go.turnForward(***leftSpeed, rightSpeed***)** where *leftSpeed* and *rightSpeed* are numbers and 0 <= *leftSpeed, rightSpeed* <= 100

**pi2go.turnReverse(***leftSpeed, rightSpeed***)** where *leftSpeed* and *rightSpeed* are numbers and 0 <= *leftSpeed, rightSpeed* <= 100

**pi2go.go(***leftSpeed, rightSpeed***):** where *leftSpeed* and *rightSpeed* are numbers and -100 <= *leftSpeed, rightSpeed* <= 100

**pi2go.go(***speed***):** where -100 <= *speed* <= 100

**pi2go.setLED(***LED, Red, Green, Blue***):** where *LED, Red, Green* and *Blue* are numbers and 0 <= *LED* <=3 and 0 <= *Red, Green, Blue* <= 4095

**pi2go.setAllLEDs(***Red, Green, Blue***):** where *Red, Green* and *Blue* are numbers and 0 <= *Red, Green, Blue* <= 4095

Try typing the following:

>> pi2go.setLED(3, 0, 0, 1000)



What happens?

pi2go.setLED controls two *light emiting diodes (LED).* It takes four inputs: the number for the LEDs (0 is front, 1 is right, 2 is back and 3 is left), followed by numbers for the amount of Red, Green and Blue light to display. So, the command you just typed gets the right LEDs to shine quite a lot of blue light and no red or green light.

Try typing the following:

>> pi2go.setAllLEDs(1000, 0, 0)



What happens?

Try six commands from the table. What do they do?

**Command Result**

**Remember:** When you have finished working with the robot, type pi2go.cleanup() at the command line, quit IDLE, then select Shutdown from the Raspberry Pi menu item. Once the robot has shut down, switch it off.



University of Liverpool, 2019

This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-nc-sa/4.0/).