THE UNIVERSITY OF
WAIKATO
*Te Whare Wānanga o Waikato*

DEPARTMENT OF
COMPUTER SCIENCE
*Te Tari Rorohiko*

# 2025

Contributors

J. Turner

V. Moxham-Bettridge

B. Jones

J. Kasmara

Au Reikura
*School of Computing and Mathematical Sciences*

# MEET NORBERT

Welcome to the newest addition to the CSNeT programme, Bricks, Barks & Binary. In today's session we introduce you to robotics with our robot dog Norbert.

*Figure 1: Our Robot Dog, Norbert*

## Robotics

As the name suggests, robotics, is the design, build, study, operation and maintenance of robots. These days interacting with a robot is common practice, from using a robot vacuum to smart home appliances to medical robots and self-driving cars. Robotics blends engineering, computer science, and creativity to create smart devices that make our lives easier, safer, and more efficient (provided they are designed correctly!).

## Norbert the Robot Dog

For the first 3 sessions of CSNeT Bricks, Barks & Binary you will get the opportunity to program Norbert, our robot dog. However, before we begin the exercises, let's take a closer look at the robot.

Norbert is a made by the robotics company, Petoi, which sell small animal robots with the purpose of making robotics and embedded systems engineering more accessible to people. They make a robotic dog called Bittle and a robotic cat called Nybble (see figure 2). Both types of

robots have open-source software which means the platform can be easily (and freely) developed by anyone.

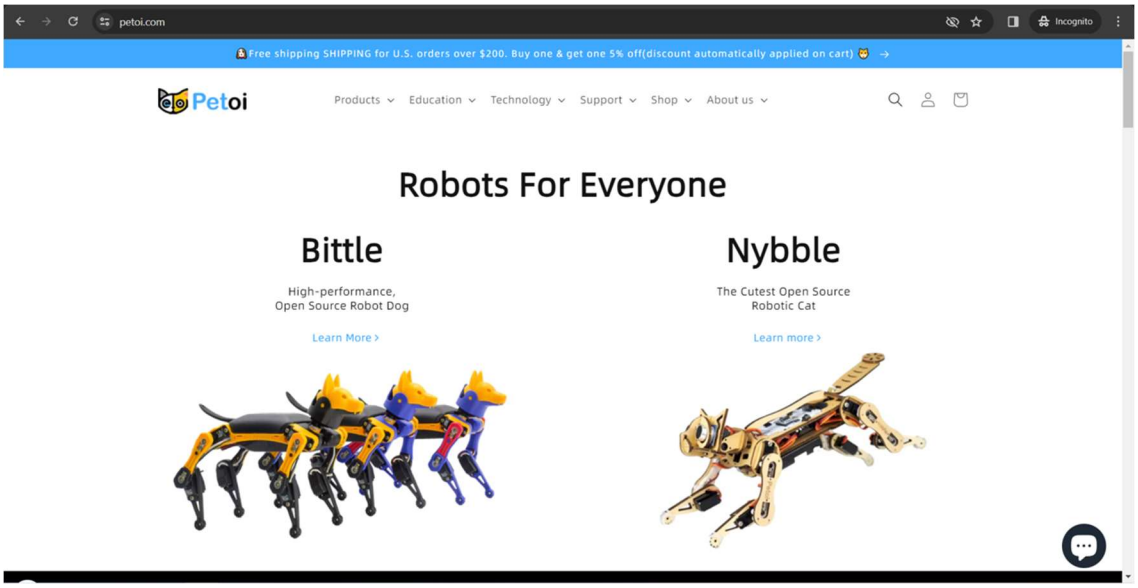


*Figure 2: Petoi.com Website*[1]

The Petoi Bittle has a frame constructed from plastic which the other components attach to (see figure 3). There is a rechargeable battery pack located on the underside of the frame's centre (i.e. the dog's belly) which delivers power to the circuit board and servos. A servomotor (or 'servo' for short) is a motor driven system which incorporates feedback into its operation (SparkFun, n.d.). These components are what makes the robot mobile, as without them the head and legs wouldn't move. Like when Micro:bits are used with Maqueen robots, the circuit board is the 'brain' of the robot (i.e. a micro-controller) which is responsible for delivering power to the components and instructing them what to do. It runs Petoi's OpenCat software which is an open-source Arduino and Raspberry Pi based framework specifically developed for use with quadruped robotic pets (PetoiCamp, n.d.).

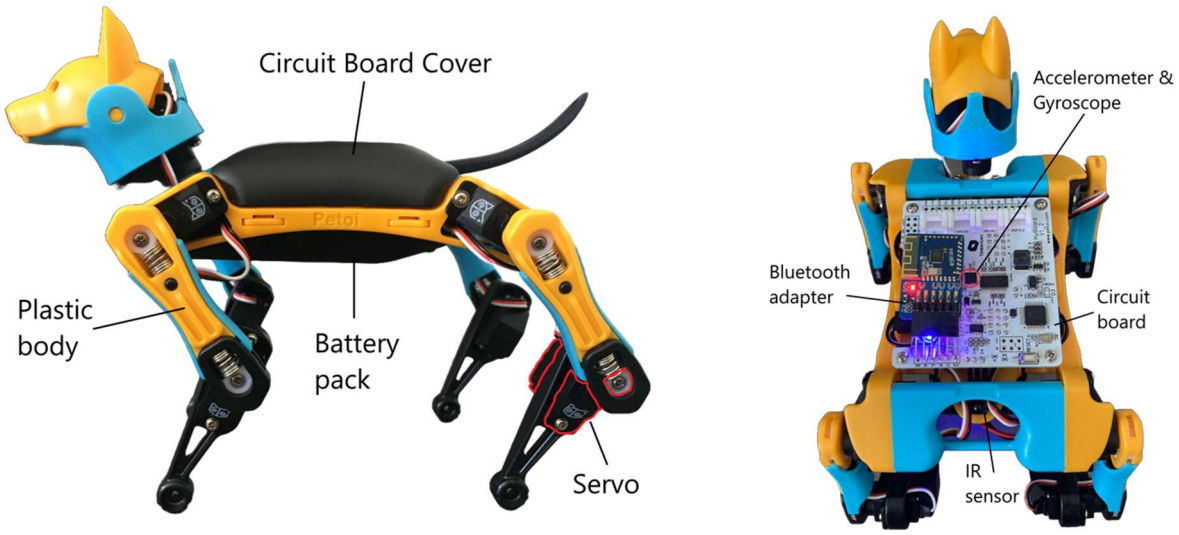


*Figure 3: Labelled Diagrams of Norbert*

Petoi includes extra sensors and adapters with Bittle that can be attached to the circuit board to expand the robot's functionality. The Bluetooth adapter can be seen in figure 3 above. This adapter allows communication with Norbert via Bluetooth connection from computers (regular Bluetooth) and mobile devices (Bluetooth LE). Also included is a Wifi adapter (allow communication via the internet) and a USB adapter (allow communication via a MicroUSB to USB-A cable).

Other sensors (referred to as 'Extensible Modules' by Petoi) include light, touch, gesture, PIR motion, IR distance, MU vision and Voice sensors/modules which are connected to digital pins on the circuit board and attached to Norbert's mouth. In later sessions, you will be using some of these to make Norbert perform tricks but in this session we cover the basic movements.

## Set Up and Configuration

Different modes are included with the robot, and which one you use depends on what you want to do with Norbert. To configure the mode, we will use the Firmware Uploader. Open the 'UI.exe' file located in the Downloads directory (see figure 4).

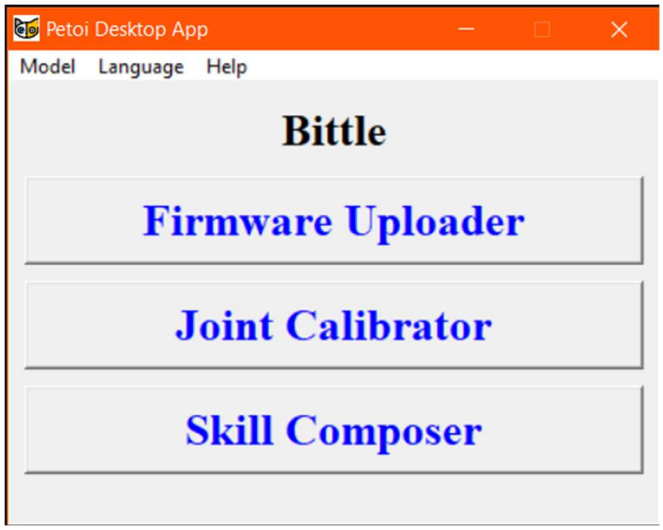


*Figure 4: Petoi Desktop App*

Select 'Firmware Uploader' which will bring up the screen shown in figure 5.

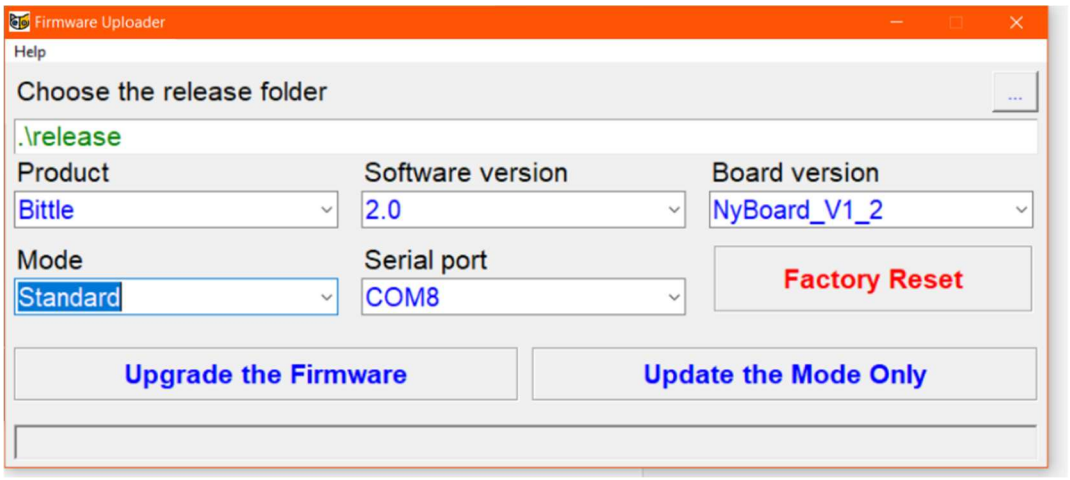


*Figure 5: Petoi Firmware Uploader*

Ensure the fields are set to what is shown above, **except the Serial Port. Please note the Serial Port number (e.g. 'COMx') must be the one that Norbert is connected to.** Each time a Bluetooth connection is established a different serial port may have been used so it must be checked using the Device Manager in Computer Management settings. If you're unsure as to how to do this, feel free to ask staff for assistance or read this document: Configuring Norbert.

Once ready, click on the 'Update the Mode Only' button and wait for the process to finish before exiting.

## Making Norbert Move

There are many preprogrammed actions provided by Petoi that are usable/doable straight out of the box. If using the IR remote (see figure 6) the actions are represented by icons and are limited to what is displayed on the remote. If using the mobile app (available on iOS and Android, see figure 7), there are more actions available and more can be added by the 'Customised Commands' feature. If using a computer, Norbert can be programmed using C++, Python or Petoi Coding Blocks (see figure 8) to use the built-in actions or to create custom actions and sequences.


*Figure 6: The Included IR Remote*

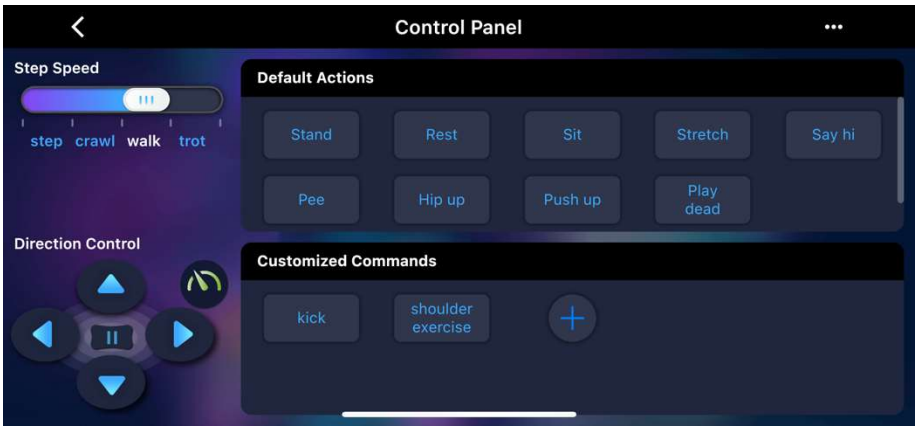


*Figure 7: Petoi Mobile App*

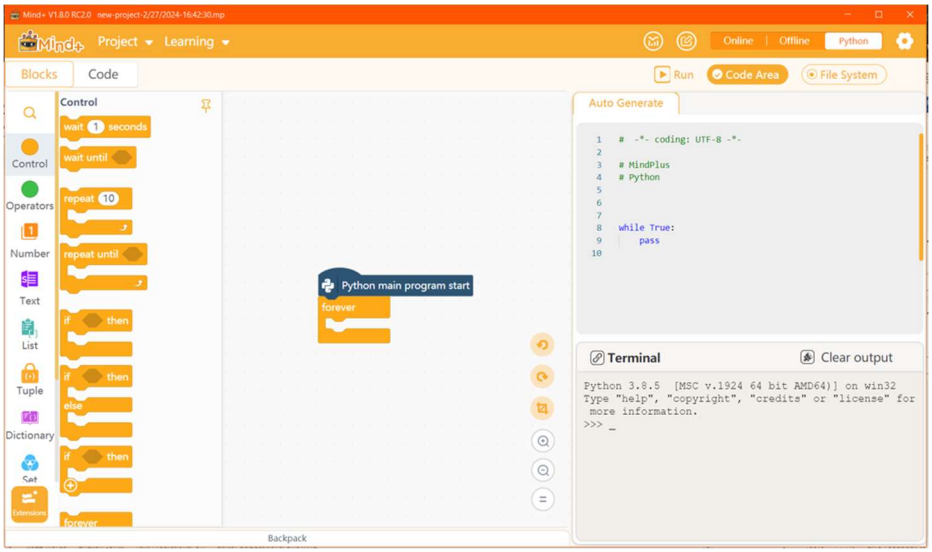


*Figure 8: Petoi Coding Blocks Interface*

To program Norbert, we are going to use Python via the Mind+ interface. Once Mind+ is opened, you should be presented with the screen shown in figure 8. However before we can program Norbert, the Petoi extension needs to be enabled. Select ‘Extensions’, ‘User-Ext’ and then the Petoi tile (see figure 9).


*Figure 9: Enabling the Petoi Coding Blocks Extension*

To check the extension has been enabled successfully, scroll down the bar on the left-hand side and click on ‘User Expansion’ (see figure 10).

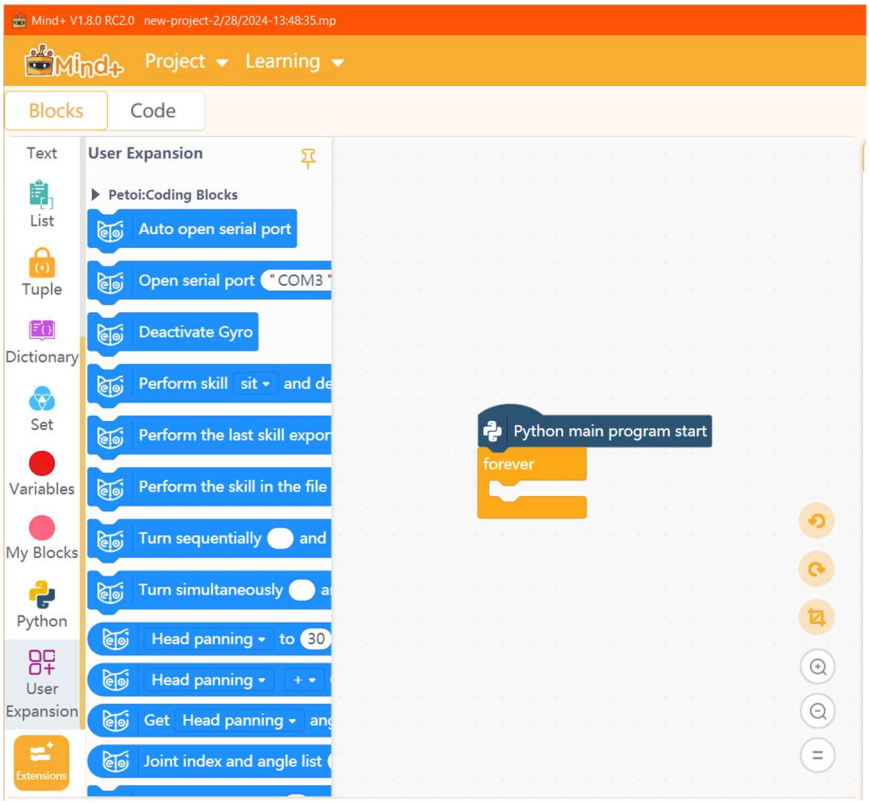


*Figure 10: Enabling the Petoi Coding Blocks Extension*

As we are going to program in Python, select the 'Code' option (located next to the 'Blocks' option) to display the Python interface. Your screen should now look like this:

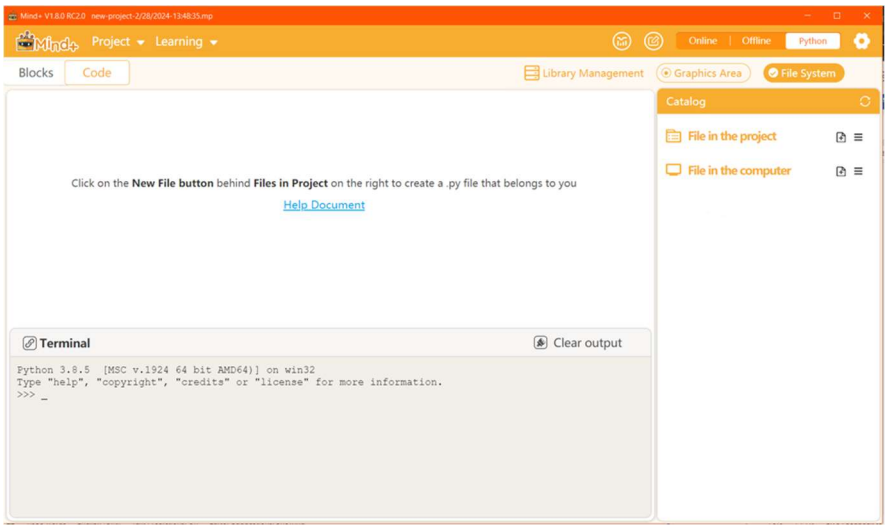


*Figure 11: Python Code Interface*

Create a new file called `woof.py` and save it in the project (see figure 12), this is where we will write our code for today's session.

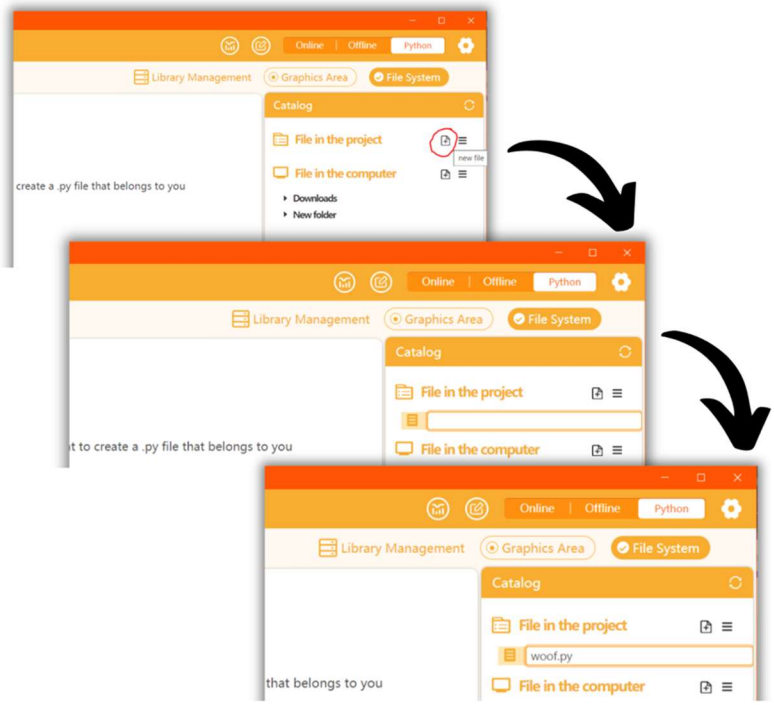


*Figure 12: Creating a Python File in Mind+*

Open the file in the editor by double clicking on the file name under the 'File in the project' heading. Now we can start writing code to control Norbert.

Firstly, we must include this code:

```
01    from PetoiRobot import *
02
03    openPort(<serial port>)
04
05    closePort()
```

Line 1 tells the program to use the PetoiRobot library (this is the user extension we enabled earlier), without it none of the commands specific to the robot would be recognised (i.e. Norbert cannot be programmed without importing this library). Line 3 instructs Norbert to use the specified serial port as the communication channel. **Note: the serial port must match the one used to configure Norbert earlier in the session, otherwise errors will occur.** Line 5 closes the communication channel and should be included after the main body of the program at the end of the file (i.e. after the skills and tricks you instruct Norbert to do).

Let's begin with instructing Norbert to sit, stand up and then walk forward for 2 seconds (see the code in figure 13).

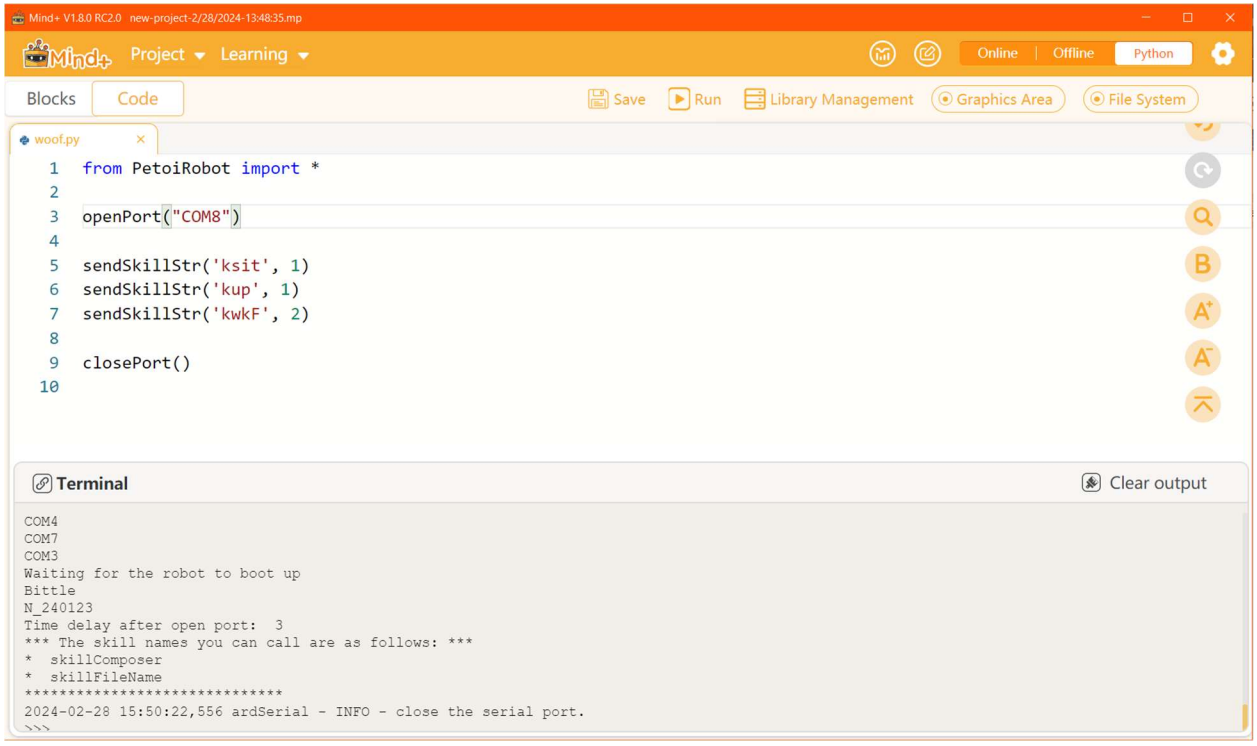


*Figure 13: Python Code for Sitting, Standing and then Walking Forward For 2 Seconds*

Click on the 'Run' button to run the program. If the program quits without Norbert moving, the reason might be due to an incorrect skill string (e.g. 'kwk**F**' – walk forward) or incorrect serial port. Feel free to ask staff for assistance if you get stuck.

What is the number passed to the `sendSkillStr()` function for? What does it do?

The number is referred to as a delay by Petoi however it's more like a duration (i.e. how long the skill will be executed for) measured in seconds (so a 4 second delay should mean the skill executes for 4 seconds before stopping).

Now that you know the command for walking forward, how do you get Norbert to walk to the left, right or backwards?

Hopefully walking left and right was easy (i.e. `kwkL, kwkR`) but walking backwards may not have been as easy because it requires a different command of `kbk`.

To make Norbert walk at different speeds, try the `kcrF` and `ktrF` commands. What does each one do? (Write it in a comment above each command in your code).

Now that you know some of the basic movements, try out each of these actions and see what they do:

- `kbuttUp`
- `kstr`
- `krest`
- `kzero`
- `kchr`
- `kck`
- `kdg`
- `khg`
- `khi`
- `khu`
- `kkc`
- `knd`
- `kpd`
- `kpee`
- `kpu`
- `kscrh`
- `ksnf`
- `kvtF`
- `kpd`
- `kphF`

## Today's Exercise

We think Norbert is an awesome dog but we believe he would benefit from some training – this is where you come in! For the first 3 sessions you will be Bertie's trainer, where you will prepare him in sessions 1 and 2 for the The Dog Show which is happening in session 3. The dogs

competing in the show must demonstrate great obedience, agility and wit to have a chance at winning. Today, you have been tasked with teaching Norbert basic agility skills so that he can perform for a staff member. What is required for his routine is listed below (remember to save your code in a safe place as you may find it helpful for The Dog Show!).

**Routine:**
Norbert should start sitting, then walk towards the bone and stop. He then needs to perform 3 different tricks (of your choosing) before trotting towards the box. Once at the box, Norbert should push it until they both cross the finishing line.

Here is the routine broken down into exercises:
**Exercises:**

1. Begin with getting Norbert to sit, walk and then stop on the bone.
2. Select the 3 tricks Norbert will perform (they must be unique).
3. Make Norbert trot forward towards the box.
4. Once at the box Norbert should change to the pushing movement (this can be done by either going directly from trot to push, or from trot to standing to push).

Once you think Norbert is ready, ask a staff member to watch his performance!

## Summary

In this session we introduced you to our robot dog, Norbert and instructed him to perform basic movements such as sitting, standing and walking. We used the Bluetooth adapter to communicate with Norbert and the Python programming language to instruct what actions he was to perform. Next week we will move onto more complex movements and incorporating various sensors to see how Norbert responds obstacles and light.

## Useful Resources

- What is robotics? (Grades 5 – 8): https://www.nasa.gov/learning-resources/for-kids-and-students/what-is-robotics-grades-5-8/
- Robotics: A Brief History: https://cs.stanford.edu/people/eroberts/courses/soco/projects/1998-99/robotics/history.html
- The Complete History and Future of Robots: https://www.wired.com/story/wired-guide-to-robots/
- Boston Dynamics Website: https://bostondynamics.com/
- Servos Explained: https://www.sparkfun.com/servos
- Petoi's OpenCat Repository on GitHub: https://github.com/PetoiCamp/OpenCat