Edbot Python

Teacher's Guide

Table of Contents

Introduction

This unit is suitable for students in KS3 and is designed to be an introduction to Python programming. It is assumed that the teacher has already introduced their students to the Python programming environment, including opening, saving and running programs.

The lessons cover the following key concepts:

- Controlling Edbot with the Python API
- Variables and lists
- Using strings and numbers (integers)
- If statements
- Comparison and mathematical operators
- Joining strings, converting strings to integers and integers to strings
- While loops, for loops and the range function
- Using random numbers
- Getting and using user input
- Error handling with the try statement
- Using a Python function

The aim of each lesson is to allow the students to control Edbot with their Python program. Students will be asked to modify existing programs or write their own programs and correct them for errors.

For each lesson there is an accompanying presentation that the teacher can use to guide the class through the lesson. Depending on the level of the class the teacher might want to miss some slides out or verbally add to the content shown. The presentations guide the students through the Python concepts listed above, with plenty of examples and opportunities for the students to have a go at programming. The first lesson introduces some of the Edbot Python functions to enable students to control Edbot straight away. Once these functions have been introduced, it's back to Python basics from lesson 2, with the opportunity for the students to control Edbot.

All the presentations have slides with recap questions to test the students' understanding. The recap answers are not provided as they are based on the lesson content.

Some lessons have extra tasks at the end that might take too long to complete during the lesson. They are there to provide the higher ability students with extra coding tasks to keep them busy. They can also be completed by the whole class if the teacher feels the students would benefit from spending more time on a particular topic before moving on to the next lesson. If there isn't time to complete them at the end of a lesson they can be used at the start of the next lesson as a recap of programming learnt to date.

Number of lessons

Recommend 7 x 1 hour teaching lessons, although this will work with slightly shorter or longer lessons.

The timings shown are colour coded to help you split up the lesson timings to fit your length of lesson.

Pink	Short activities (under 5 minutes each)
Yellow	Medium length activities (between 6 and 12 minutes long)
Blue	Longer activities (over 12 minutes long)

The exact time taken to complete the exercises will partly depend on student ability and how many students need to test their code on Edbot each time. If time is short it might be useful to complete some of the exercises in pairs or small groups, as long as everybody has some individual coding time. Alternatively, if more time is available, some lessons could be split over 2 sessions.

Suitability

This unit is suitable for mixed ability classes. It is suitable for students with no prior experience of Python, but would also be suitable for students with some experience, to consolidate their Python learning before moving onto more advanced work.

Differentiation

Lessons are differentiated by outcome and this is reflected in the success criteria and part of each lesson involves the students working independently, freeing up the teacher to offer more one-to-one help to the students who need it. There are also instructions in the lesson plans on how to adapt the lesson for lower or higher ability students.

Preparation needed by the teacher

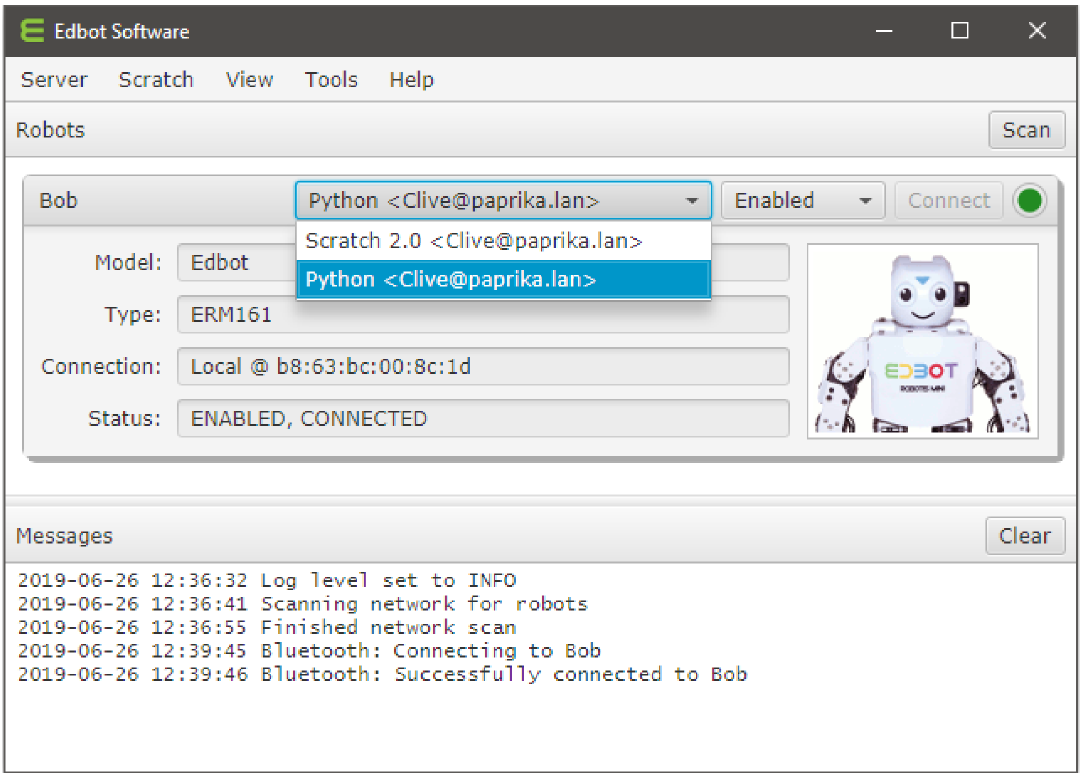
The teacher needs to make sure that they are familiar with the contents of each lesson and the activities and questions involved which will help them with assisting the students in the lesson.

The teacher will need to be familiar with the Edbot Python API. The documentation is available via the Edbot Software by selecting "Documentation…" in the "Help" menu. The teacher can use the edbot_python_api_intro.pptx presentation as a reference guide to the Python API and share it with the students to remind them of the syntax of the commands they have learnt.

The Edbot Software needs to be installed and configured on the teacher's PC with the Edbot connected via Bluetooth. The teacher needs to know the name of the PC. Students should update their Python programs to connect to the teacher's PC over the network by replacing "localhost" with the name of the teacher's PC in their code:

ec = edbot.EdbotClient("<teacher's pc name>", 8080)

The teacher can then use the active connection dropdown menu in the Edbot Software to assign control to a specific student:



Bob
Python <Clive@paprika.lan>
Scratch 2.0 <Clive@paprika.lan>
Python <Clive@paprika.lan>

Model: Edbot
Type: ERM161
Connection: Local @ b8:63:bc:00:8c:1d
Status: ENABLED, CONNECTED

```
2019-06-26 12:36:32 Log level set to INFO
2019-06-26 12:36:41 Scanning network for robots
2019-06-26 12:36:55 Finished network scan
2019-06-26 12:39:45 Bluetooth: Connecting to Bob
2019-06-26 12:39:46 Bluetooth: Successfully connected to Bob
```

It is a good idea to have the students log in with their individual user names if they have them. This makes it easier to find their connection in the Edbot Software to give them control of the Edbot.

NOTE: For single user development on the same machine, you can select the “Bypass active user” option in the Edbot Software -> Server Setup -> Configuration page. This will disable the active user menu and means any connection can control the Edbot. In practice this means you don’t need to keep selecting your own connection from the dropdown menu every time you run your Python program.

Make sure the Edbot’s batteries are fully charged and installed correctly.

Each student should be issued with a mark sheet at the beginning of the unit. Some lessons use videos to teach the students the skills they need; this will require the teacher to make sure that they have speakers attached and switched on.

The lessons require the students to use a python file (edbot_python_blank.py) that has some code already added which sets the Edbot up. The teacher will need to explain to students roughly what this code does and that the students do not need to change it (other than the Edbot name). The students do not need to understand all this code to start with. At a later stage the teacher can go back and talk about it in more detail. The students will be adding code to this program or modifying code in other programs, as specified in the lessons. The teacher also needs to explain that the students will be using special Python Edbot functions to operate Edbot. Initially they may not understand the structure of these functions but they will need to be able to use them by copying the syntax and changing values.

Assessing without Levels

"As part of our reforms to the national curriculum, the current system of 'levels' used to report children's attainment and progress will be removed. It will not be replaced.

We believe this system is complicated and difficult to understand, especially for parents. It also encourages teachers to focus on a pupil's current level, rather than consider more broadly what the pupil can actually do. Prescribing a single detailed approach to assessment does not fit with the curriculum freedoms we are giving schools." [1]

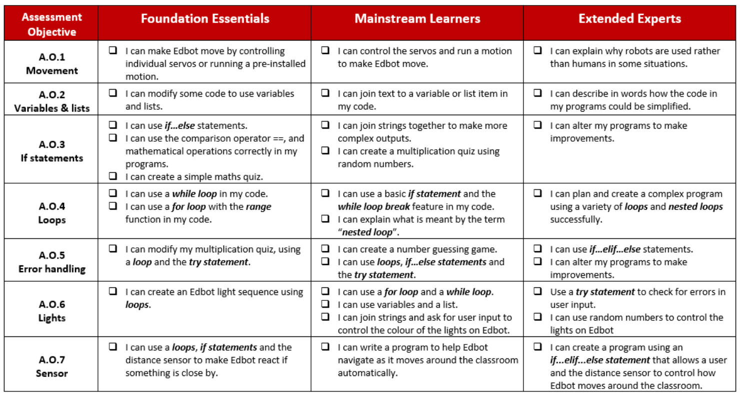
With this in mind, we have developed a three-tier system which can easily be adapted to any system your school has implemented. We have referred to these bands as

- Foundation Essentials
- Mainstream Learners
- Extended Experts

Assessing Progress

Each student should be given a mark sheet on which they will need to write their name, so that they can get the same sheet back each lesson and could be kept in a work folder which they can refer to every lesson.

The assessment sheet is based on "A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives"[2] and avoids use of the old national curriculum levels altogether. This means that the same unit can be used with different year groups and the students can still show they have made progress.



Assessment Objective	Foundation Essentials	Mainstream Learners	Extended Experts
A.O.1 Movement	❑ I can make Edbot move by controlling individual servos or running a pre-installed motion.	❑ I can control the servos and run a motion to make Edbot move.	❑ I can explain why robots are used rather than humans in some situations.
A.O.2 Variables & lists	❑ I can modify some code to use variables and lists.	❑ I can join text to a variable or list item in my code.	❑ I can describe in words how the code in my programs could be simplified.
A.O.3 If statements	❑ I can use ***if…else*** statements. ❑ I can use the comparison operator ==, and mathematical operations correctly in my programs. ❑ I can create a simple maths quiz.	❑ I can join strings together to make more complex outputs. ❑ I can create a multiplication quiz using random numbers.	❑ I can alter my programs to make improvements.
A.O.4 Loops	❑ I can use a ***while loop*** in my code. ❑ I can use a ***for loop*** with the ***range*** function in my code.	❑ I can use a basic ***if statement*** and the ***while loop break*** feature in my code. ❑ I can explain what is meant by the term "***nested loop***".	❑ I can plan and create a complex program using a variety of ***loops*** and ***nested loops*** successfully.
A.O.5 Error handling	❑ I can modify my multiplication quiz, using a ***loop*** and the ***try statement***.	❑ I can create a number guessing game. ❑ I can use ***loops, if…else statements*** and the ***try statement***.	❑ I can use ***if…elif…else*** statements. ❑ I can alter my programs to make improvements.
A.O.6 Lights	❑ I can create an Edbot light sequence using ***loops***.	❑ I can use a ***for loop*** and a ***while loop***. ❑ I can use variables and a list. ❑ I can join strings and ask for user input to control the colour of the lights on Edbot.	❑ Use a ***try statement*** to check for errors in user input. ❑ I can use random numbers to control the lights on Edbot
A.O.7 Sensor	❑ I can use a ***loops***, ***if statements*** and the distance sensor to make Edbot react if something is close by.	❑ I can write a program to help Edbot navigate as it moves around the classroom automatically.	❑ I can create a program using an ***if…elif…else statement*** that allows a user and the distance sensor to control how Edbot moves around the classroom.

The assessment should be completed at the end of every lesson by the students as a form of self-assessment and the last slide in each presentation tells the students the skills that they have covered. The students tick the box next to the objective if they feel they have fully met that criteria. The teacher can then use this as a basis to help them assess the students' ability along with class observations, questioning students and viewing the students' work.

[1] Taken from www.education.gov.uk/schools/teachingandlearning/curriculum/nationalcurriculum2014/a00225864/assessing-without-levels downloaded on 5th March 2014

[2] Anderson, L.W. (Ed.), Krathwohl, D.R. (Ed.), Airasian, P.W., Cruikshank, K.A., Mayer, R.E., Pintrich, P.R., Raths, J., & Wittrock, M.C. (2001). A taxonomy for learning, teaching, and assessing: A revision of Bloom's Taxonomy of Educational Objectives (Complete edition). New York: Longman.

Edbot Python — Lesson 2 – Variables and lists

Lesson objective:	Use variables and lists in your programs.	
All will be able to: Modify some code to use variables and lists.	**Most will be able to:** Join some text to a variable or list item in your code.	**Some will be able to:** Describe in words how the code in the programs could be simplified.

Differentiation

Low Ability:	**High Ability:**
When they are modifying the code, you may need to remind them how to define a numeric and a string variable. In the second exercise you may need to check that they are putting the list items in the correct order and remind them that the first item in a list has an index number of 0.	If some students finish the list task early, they could modify their lists to contain the variables from the first exercise. Alternatively get them to define blank lists and then use the insert or append functions to build their lists up.

Starter

Time	Description	Resources
Short Activity	Show slide 2 as a reminder of how to run an Edbot motion. Let the class know that they will be using the run_motion command during the lesson. Explain the objectives to the class.	edbot_python_ lesson2.pptx Slide 2 - 3

Main Activities

Time	Description	Resources
Medium Activity	Go through the slides about variables and getting Edbot to speak. Once you have discussed the code on slide 7 and explained the task ask the students to open the Python file ***edbot_python_lesson2_variables.py*** and modify it as indicated. Let the students run their code on Edbot when they are ready.	Slides 4 - 8 edbot_python_ lesson2_ variables.py
Short Activity	Go through the example answer. Find out whether the students used generic variable names e.g. move1 = 1, as shown, or specific names e.g. initial_position = 1. Mention that using generic names is helpful if they want to change the associated values of those variables in the future.	Slide 9
Long Activity	Go through the slides about lists. Slides 14-16 cover list functions that are not used in these lessons (unless the higher ability students are asked to use them as an extension exercise), so you might want to omit them or go through them briefly, depending on the level of your students. Slide 17 shows the same code that we saw on slide 8. This time the students need to re-write it using lists as explained on the slide. Ask the students to open the Python file ***edbot_python_lesson2_lists.py*** and modify it as indicated.	Slides 10 – 17 edbot_python_ lesson2_lists.py
Short Activity	Go through the example answer. Point out the repetitive code, and that this can be simplified further once the students have learnt about loops. They will be modifying this code in lesson 4.	Slide 18

Review

Time	Description	Resources
Medium Activity	Go through the questions on the slide with the whole class. This will help them recap what they have learnt in the lesson.	Slide 19

Self-Assessment

Time	Description	Resources
Short Activity	Give out the mark sheet from the previous lesson. The students read through the highlighted objectives and if they feel they have met the criteria fully they need to tick the box. If they do not feel they have met the objective they should not tick the box.	Slide 20 edbot_python_ mark_sheet.pdf