

Edbot Play&Code

Python

Teacher's Guide Sample



Table of Contents

Introduction	2
Number of lessons	2
Suitability	3
Differentiation.....	3
Preparation needed by the teacher	3
Assessing without Levels.....	5
Assessing Progress	5
Lesson 1 - Windmill 1: Make it & make it go!.....	6
Lesson 2 - Windmill 2: Variables	7
Lesson 3 - Windmill 3: Lists	8
Lesson 4 - Puppy 1: Time for a walk	9
Lesson 5 - Puppy 2: If only.....	10
Lesson 6 - Puppy 3: Going Loopy	11
Lesson 7 - Car 1: Learning to drive	12
Lesson 8 - Car 2: Reverse parking	13
Lesson 9 - Car 3: That didn't work! Time to race?.....	14
Lesson 10 - Scorpion 1: Walking speed	15
Lesson 11 - Scorpion 2: No escape!.....	16
Lesson 12 - Scorpion 3: Follow that line.....	17

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 Play&Code robots 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
- Use of different sensors to control a robot

The aim of each lesson is to allow the students to control their Edbot Play&Code robot with a 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 PowerPoint presentation that the teacher can use to guide the class through the lesson. The presentations guide the students through the programming concepts listed above, with plenty of examples and opportunities for the students to have a go at programming.

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

Most lessons have extension coding tasks for higher ability students, so that they will always have something to keep them busy.

Example Python files are included for some of the coding tasks. In other cases, example code is shown in the presentation. This code can be copied into a Python file for demonstration purposes.

Number of lessons

There are 12 lessons which can be split into 2 blocks of 6. Most of the programming techniques listed above are introduced in the first 6 lessons and are based on the windmill and puppy robots. Lessons 7 - 12 provide an opportunity to further practice Python programming techniques using the car and scorpion robots. A minimum of 12 x 1 hour teaching lessons are recommended. Some of the lessons have extension ideas and further activities that can fill 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)

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 knowledge before moving onto more advanced work.

Differentiation

Lessons are differentiated by outcome and this is reflected in the success criteria. 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

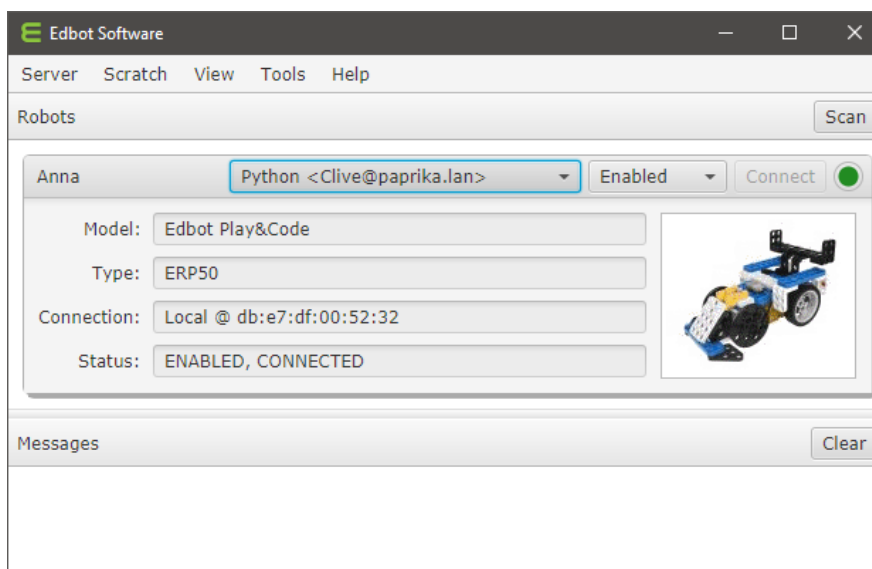
The teacher needs to make sure that they are familiar with the contents of each lesson and the activities and questions involved which will help them with assisting the students in the lesson.

The teacher will need to be familiar with the Edbot Play&Code Python API. The documentation is available via the Edbot Software by selecting “Documentation...” in the “Help” menu. The teacher can use the edbot_play_python_api_intro.pptx presentation as a reference guide to the key features of 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 network name of their 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:



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 Play&Code robot.

NOTE: For single user development on the same machine, you can select the “Open access” (previously “Bypass active user”) option in the Edbot Software -> Server Setup -> Configuration page. This will disable the active connection menu and means any connection can control the Edbot Play&Code robot. 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 Play&Code robots have been fitted with batteries that are installed correctly.

Four of the lessons include a time slot to build the Edbot Play&Code robots. The teacher will need to decide whether the construction of the Edbot Play&Code robots will be part of the lesson, or whether this will be an activity given to the students at another time. This may depend of the number of Edbot Play&Code kits available to the class. Building the robots provides the students with an opportunity to better understand how the robots work. It also provides an opportunity for developing fine motor skills, teamwork, communication and organisational skills.

Approximate robot assembly times (for an adult):

Edbot Play&Code robot	Approximate time to make (minutes)	Approximate time to take apart (minutes)
Windmill	15	5
Puppy	17	6
Car	20	7
Scorpion	20	7

Each student should be issued with the mark sheet at the beginning of the unit.

The lessons require the students to use a python file (e.g. edbot_play_python_blank.py) that has some code already added which sets the Edbot Play&Code robot up. The teacher will need to explain to the students roughly what this code does and that the students do not need to change it (other than the Edbot Play&Code robot name and teacher’s pc 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 Play&Code functions to operate their robot. 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.

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.

Lesson	Foundation Essentials	Mainstream Learners	Extended Experts
1. Make it & make it go!	<input type="checkbox"/> I can make a robot move by controlling a motor’s speed and direction.	<input type="checkbox"/> I can build a windmill robot.	<input type="checkbox"/> I can explain why robots are used rather than humans in some situations.
2. Variables	<input type="checkbox"/> I can write a Python program to control the speed and direction of a robot’s motor.	<input type="checkbox"/> I can create and use variables. <input type="checkbox"/> I can combine a string & a number variable.	<input type="checkbox"/> I can debug and correct a program.
3. Lists	<input type="checkbox"/> I can modify some code to use a list.	<input type="checkbox"/> I can join some text to a list item in my code.	<input type="checkbox"/> I can describe in words how the code in my programs could be simplified.
4. Time for a walk	<input type="checkbox"/> I can use 2 motors to control the speed and direction of a robot’s movement.	<input type="checkbox"/> I can build the puppy robot.	<input type="checkbox"/> I can plan and write my own program to make the puppy walk about.
5. If only...	<input type="checkbox"/> I can use an if statement to control a robot. <input type="checkbox"/> I can ask for and use user input.	<input type="checkbox"/> I can use an if...elif statement to control a robot with user input.	<input type="checkbox"/> I can give examples of the conditions that can be used with an if statement.
6. Going loopy!	<input type="checkbox"/> I can use 2 motors to control the direction of the puppy’s movement. <input type="checkbox"/> I can use a while loop in my code.	<input type="checkbox"/> I can use a while loop with a break statement to control a robot. <input type="checkbox"/> I can modify some code to use a for loop and the range function.	<input type="checkbox"/> I can make improvements to my programs.

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 Play&Code Python

Lesson 1 - Windmill 1: Make it & make it go!

Lesson objective:	Understand how to write simple commands to control a robot.	
All will be able to: Make a robot move by controlling a motor's speed and direction.	Most will be able to: Build a windmill robot.	Some will be able to: Explain why robots are used rather than humans in some situations.

Differentiation

Low Ability: Partner them with another student who might be able to help them.	High Ability: Encourage them to think about how robots may be used in real-world situations.
--	--

Starter

Time	Description	Resources
Medium Activity	Display the starter slide and give the students a short time to talk about the questions together as a pair and then ask random pairs for their answers. Use slide 3 to explain what the unit is about.	edbot_play_python_lesson1.pptx Slides 2 – 3

Main Activities

Time	Description	Resources
Short Activity	Explain the objectives to the class and go through slides 5 & 6 to explain what a robot is.	Slides 4-6
Long Activity	The students build the windmill robot.	Slide 7 Robotis Play 700 kit
Long Activity	Go through the slides to explain how the robot can be controlled using Python commands. Encourage the students to hold and examine the robot whilst you go through slides 9 & 10, explaining how to control the robot's motors.	Slides 8 - 15
Medium Activity	The students now have a go at using the Edbot Play&Code Python functions to control the windmill's motor. They should open the file edbot_play_python_blank.py and save it as a new name. When they are ready to turn the windmill robot on, they must hold the power button down until it beeps twice . When they raise their hand, select their computer from the dropdown list in the Edbot Software which gives them control of the Edbot Play&Code robot. An example answer is shown on slide 17.	Slide 16 – 17 edbot_play_python_blank.py

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 18

Self-Assessment

Time	Description	Resources
Short Activity	Give out the mark sheet for lessons 1 to 6 from edbot_play_python_mark_sheet.pdf and make sure your students write their name on it. They 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.	edbot_play_python_mark_sheet.pdf Slide 19

Edbot Play&Code Python

Lesson 2 - Windmill 2: Variables

Lesson objective:	To write a program to control a robot and learn how to use variables.	
All will be able to: Write a Python program to control the speed and direction of a robot's motor.	Most will be able to: Create and use variables and combine a string with a number variable.	Some will be able to: Debug and correct a program.

Differentiation

Low Ability: Provide a Python file which already has the relevant functions added and ask them to edit the function parameters.	High Ability: Get them to complete the extension on slide 12 and the debugging exercise on slide 14.
---	--

Starter

Time	Description	Resources
Short Activity	Remind the students what they learnt in the last lesson and explain the objectives of this lesson.	edbot_play_python_lesson2.pptx Slides 2 - 3

Main Activities

Time	Description	Resources
Short Activity	Explain the Edbot Play&Code Python function that allows the robot to speak via their computer's speakers.	Slide 4
Long Activity	The students now have a go at creating a program, using edbot_play_python_blank.py , to control the windmill (slide 5). Remind them to save their program with a new name. When they raise their hand, select their computer from the dropdown list in the Edbot Software which gives them control of the Edbot Play&Code robot. Slide 6 asks them to modify their program so that the windmill will make some announcements.	Slides 5 – 6 Windmill robot edbot_play_python_blank.py
Short Activity	Go through the example answer.	Slide 7
Medium Activity	Go through these slides with the class, explaining what variables are and how to create them. Go through the examples for combining variables and explain how to convert a string to an integer and an integer to a string.	Slides 8 - 11
Medium Activity	The students modify their program from the start of the lesson by using variables for the motor speeds and time delays. Make sure everyone has time to test their programs. An example answer is shown on slide 13.	Slides 12 - 13
Medium Activity	Extension - slide 14: A program has been incorrectly coded. Can the students use the storyboard shown to debug and change it so that it follows the storyboard correctly	Slides 14 – 15 edbot_play_python_lesson2_debug.py

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 16

Self-Assessment

Time	Description	Resources
Short Activity	Give out the mark sheet from the previous lesson. They 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.	edbot_play_python_mark_sheet.pdf Slide 17