

Edbot Basics

Answer Book



Table of Contents

Introduction	1
Lesson 1 – Motors and motions	2
Lesson 2 – Going Loopy	3
Lesson 3 – Operators and if blocks	4
Lesson 4 – Light show.....	6
Lesson 5 – Using the sensor	7
Lesson 6 – Let’s play football	8

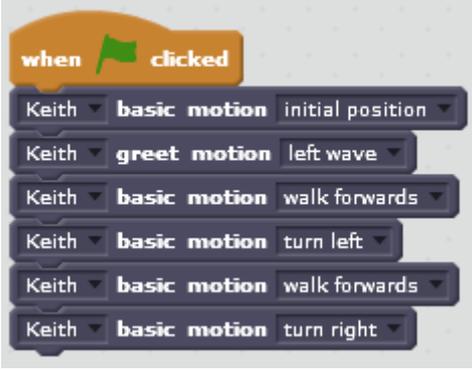
Introduction

This answer booklet has been produced to give the teacher some help on the questions and problems which are included in the Edbot Basics lesson plans. The answers within this booklet cover the questions on the PowerPoint presentations where there is a definite answer that is not included within the presentation itself.

Lesson 1 – Motors and motions

<p>Starter</p>	<ol style="list-style-type: none"> 1. A robot is a machine capable of automatically carrying out a complex series of actions. 2. There are several reasons why robots are used. These include: when it is too dangerous for a human to do it (bomb disposal or taking readings from inside a volcano), when it will take too long to travel there (Mars exploration), when the same tasks needs to be performed time and time again with exact precision (building an intricate circuit board), when it will be too boring for a human to do it (robot vacuum cleaners or moving items from one position to another in a warehouse – Amazon uses lots of robots in their warehouses now) etc. Robots can work 24 hours a day, 7 days a week and do not need to take a break or have sick days. 3. They are controlled by a computer program.
<p>Slide 14</p>	<ol style="list-style-type: none"> 1. When the green flag is clicked. 2. More Blocks. 3. Stand in the initial position. 4. A motor that controls how the arms and legs move. 5. [Name of Edbot] set servo [number] on 6. [Name of Edbot] set servo [number] position [number]
<p>Slide 20</p>	<ol style="list-style-type: none"> 1. Because it can be too dangerous or boring for a human to do it or the job requires fine precision and repetition that humans are not capable of. 2. A robot that looks like a human i.e. it has a head, arms, legs etc. 3. A motor that allows the parts of Edbot to move. 4. 16 and they control the arms and legs. 5. It makes programming Edbot easier rather than having to program each servo individually. 6. In More Blocks, drag the correct section into the area and select the motion from the drop-down box. 7. To make sure they are working correctly and allows you to change it if necessary.

Lesson 2 – Going Loopy

Starter	 A Scratch script starting with a 'when green flag clicked' event block. It contains six 'basic motion' blocks for the character 'Keith': 'initial position', 'left wave', 'walk forwards', 'turn left', 'walk forwards', and 'turn right'.
Slide 6	<p>1. When the green flag is clicked, set Edbot to the start position (initial position) and wait for a second. Make Keith walk forwards and then wave its left arm. Keep repeating this action until the space bar is pressed and then stop.</p>

Lesson 3 – Operators and if blocks

Starter	<ol style="list-style-type: none"> When the green flag is clicked set Edbot to the correct starting position (initial position). Make Edbot walk forward until the space bar is pressed and then it will wave its left arm. Once it has waved its arm, Edbot should start walking forward again until the next time the space bar is pressed.
Slide 4	<ol style="list-style-type: none"> When the green flag is clicked, set Edbot to the correct starting position (initial position). It will keep checking if the space bar has been pressed. If it has, Edbot will wave its arm, otherwise it will continue waiting. The forever block is used to make sure it keeps checking if the space bar has been pressed otherwise it will check once and then stop.
Slide 7	<ol style="list-style-type: none"> Initial Position. In the Sensing category use the “ask [question] and wait”. Control category. Operators category. If (answer) = [The correct answer]. Two.
Slide 9	<ol style="list-style-type: none"> When the green flag is clicked set Edbot to the correct starting position (initial position). Ask for the user’s name and then Edbot will wave its left arm and say “Hello [name]” for 2 seconds. The join block allows two pieces of text (known as strings) to join together to form a single output.
Slide 11	<ol style="list-style-type: none"> 10. To give a more complex output for the response. It picks a random number between 2 and 12. Say is in “Looks” and play sound is in “Sounds”. Create another variable called score and set that to 0 when the green flag is clicked before it starts the loop. Every time they get the question correct add 1 to the score. At the end of all the questions add another line telling the user their final score.
Slide 13	<ol style="list-style-type: none"> “num” will select a random number between 1 and 10, “end” will save true or false. In Data select “make a variable”, give it a name and click on OK. Repeat until end = true. Sensing. Control. Looks.
Slide 14	<p><u>Extension Activities:</u> If the student completes both extension activities their program may look the same as the program on the next page. However, please note there are several ways they can complete the program and this is only one possible solution.</p>

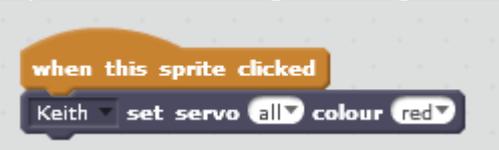
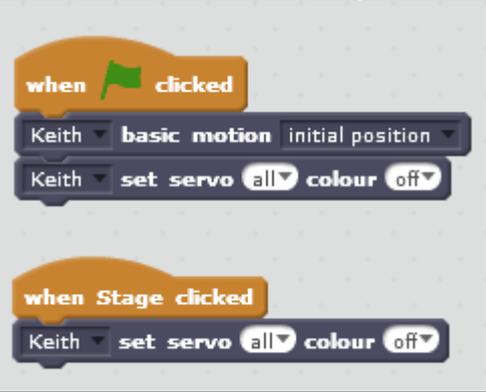
**Slide 14
continued**

```
when clicked
  Keith basic motion initial position
  set num to pick random 1 to 10
  set end to false
  set guesses to 0
  repeat until end = true
    ask Guess the number I am thinking of and wait
    if answer = num then
      say Correct for 2 secs
      Keith dance motion break dance
      set end to true
    else
      if answer < num then
        say Too low, try again for 2 secs
      else
        say Too high, try again for 2 secs
      change guesses by 1
    say join You took join guesses guesses for 2 secs
```

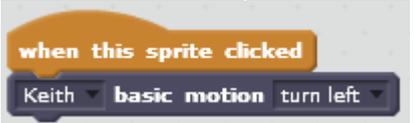
Slide 15

1. An if block allows the program to test a condition and decide which route to take depending on whether that condition has been met or not.
2. In the Sensing category you can ask a question and wait for the user to type in their answer.
3. A value that can be changed.
4. In the Operators category you can pick a random number.
5. Operators.
6. Join two or more pieces of text together.

Lesson 4 – Light show

Starter	The servos are listed individually as we only want the bottom half of Edbot to turn yellow and not all the servo lights to turn yellow.
Slide 6	<ol style="list-style-type: none"> 1. Click on the sprite and then add the code. 2. When that sprite is clicked. 3. It should switch off all the lights. 4. It should set Edbot to the initial position and switch off all the lights. 5. Right-click on it and select duplicate and then drag the block onto the sprite you want it to be attached to.
Slide 7	<p>On each sprite there should be the following script (make sure the colour is set correct for each sprite to make the lights change to the correct colour:</p>  <p>On the stage there should be two blocks of script added as follows:</p> 
Slide 8	<ol style="list-style-type: none"> 1. A variable is a value that can change 2. In Data click on “Make a variable”, give it a name and click on OK. 3. When the green flag is clicked Edbot will stand in the initial position. A variable called servonum is set to 1. It will change that servo to red, yellow, blue and then off with a quarter of a second pause between each. It will then go to servo 2 and repeat that sequence, then 3 and so on until the servo number is 17 when it will stop.
Slide 9	It will change servo number 3 to red (this is explained more on the following slides).

Lesson 5 – Using the sensor

<p>Starter</p>	<p>When the green flag is clicked set Edbot to the initial position. It will check the distance sensor and if it is below 25 (i.e. something is within 25cm of the sensor) Edbot will bow and then wave its right arm. It will keep checking this until the program is stopped.</p>
<p>Slide 10</p>	<p>On the stage there should be the following script:</p>  <p>On the right arrow sprite there should be the following script:</p>  <p>On the left arrow sprite there should be the following script:</p>  <p>On the up arrow sprite (forwards) there should be the following script:</p>  <p>On the down arrow sprite (backwards) there should be the following script:</p> 

Lesson 6 – Let’s play football

Starter	When the green flag is clicked set Edbot to the initial position. If the up arrow key is pressed Edbot will walk forwards, if the down-arrow key is pressed it will walk backwards. It will keep checking if either key is pressed until the program is stopped.
----------------	--

When they are planning their action try to get them to think about lots of possibilities. Below are some examples with possible keyboard keys suggested but your pupils are welcome to develop a new set of mappings if they wish.

As they play the football match encourage them to make changes to their program if necessary (use a “Time Out” option to give them time to do this).

Motion	Suggested Key
Walk forwards	Up arrow
Walk backwards	Down arrow
Sidestep left	Left arrow
Sidestep right	Right arrow
Turn left	Q
Turn right	W
Left kick	1
Right kick	2
Goalie Block	9
Goalie Spread	0
Gangnam Style (goal celebration)	C
Get up	Space bar