Upper Key Stage 2 - Session 4
Understanding more Python commands (while, if.. elif..else)

Objectives

- Create a core program in Blockly and translate and understand how it is written in Python
- Understand how the syntax of selection statements works in Python
- Understand code written in Python using **while**, **if**, **elif**, **else** commands
- Analyse how **procedures** work in Python (extension)

Resources

- Interactive White Board (IWB)
- Levels 85 – 91 in Rapid Router
- Resource sheet UKS2-S4-1
- UKS2 Levels Guide
- UKS2 Program Solutions Table
- Blockly-Python phrasebook (copy the section for pupils on **while**, **if**… , pages 2-3)

Vocabulary

- while,
- if, elif, else
- Indent
- Colon
- Bracket

Let's get started

In this session, children will learn that the code in Python for the Blockly **repeat until**, uses the **while** command.

Who remembers what we discovered about repeat until at destination and repeat while not at destination?

What do you predict this will look like in Python?

Look at Level 85 on the IWB. *[fig S4.1]*

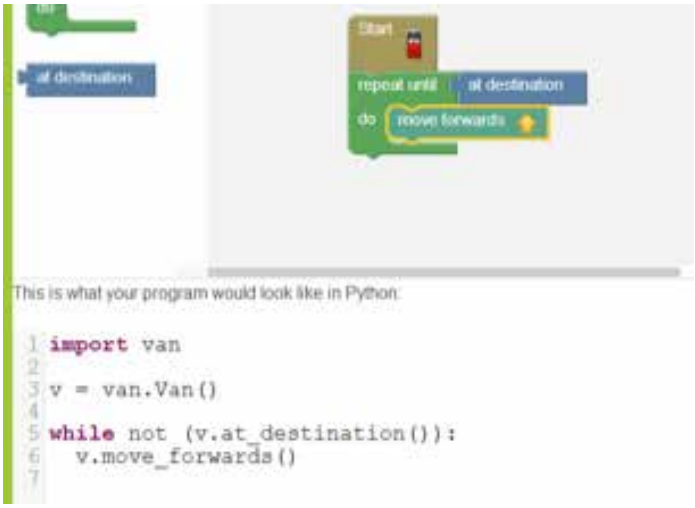
fig S4.1

Split the children up into pairs and take two minutes with their partner to write just two lines of code in Blockly using **repeat until**, which gets the van to the house.

Choose a pair to create their solution on their whiteboards, hopefully as follows:

What do you think this may look like when written in Python?

Now for the big translation. Look at the code in the Python pane as we create the Blockly code.



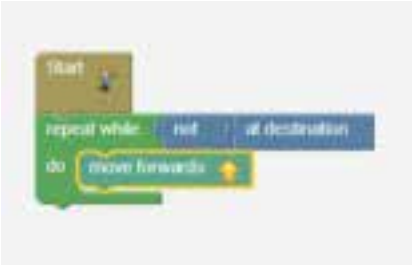
Is the code in Python what you expected? If not, why not?

What do you predict the code in Python would be if we used the Blockly while not at destination?

Let's test it.

Go to the Create mode and generate a simple straight road. Make sure to select all the Blockly blocks and select 'Both' in the code box. *[fig S4.2*

Create the program



And see what the code looks like in Pvthon:

```
while not (v.at_destination()):
  v.move_forwards()
```

What does this code tell us?

We now know that in Python we use **while not at destination**, Python does not have the until instruction in its language.

Let's look more closely at the Python language.

Experiment to see what happens if you remove the outer set of brackets in Python.

You will notice that they are not essential. In fact, if you are coding directly in Python you could use

```
while not v.at_destination():
```

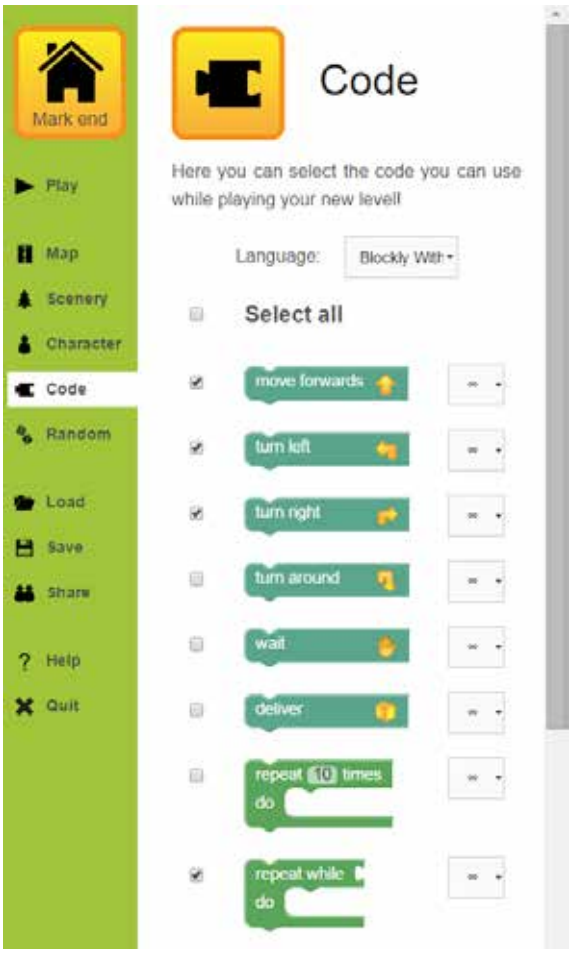


fig S4.2

Main activity

Tell the children that they are now detectives. Ask them to solve Level 86 in Blockly and discuss how it works in Python. *[fig S4.3]*

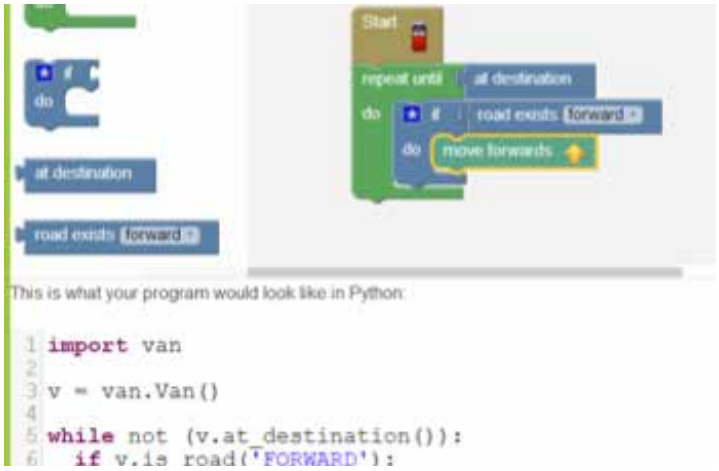
Their challenge is to be able to explain how **if**… **elif**… **else** works in Python.

fig S4.3

Mini review

What have you discovered about the way the if statement is coded in Python?

Choose a pair to demonstrate the **if** statement, and discuss the syntax, pointing out that each line after the **while** statement is indented. Also point out the importance of the brackets, inverted commas and colons.



Give out the extract from the Blockly-Python phrasebook for the children to refer to.

Paired activity

Try Levels 87 to 89 which include the use of traffic light **variables** within the **if** statements.

Share and review

Summarise what they have learnt about **if** statements in Python. Discuss the importance of indents, brackets and colons.

When do we find colons: at the end of lines of code?

What do you notice about the lines that are indented?

When are single inverted comas used?

Give out Resource Sheet UKS2-S4-1 and ask the children to debug the program code. This could also be a homework activity. *[fig S4.4]*

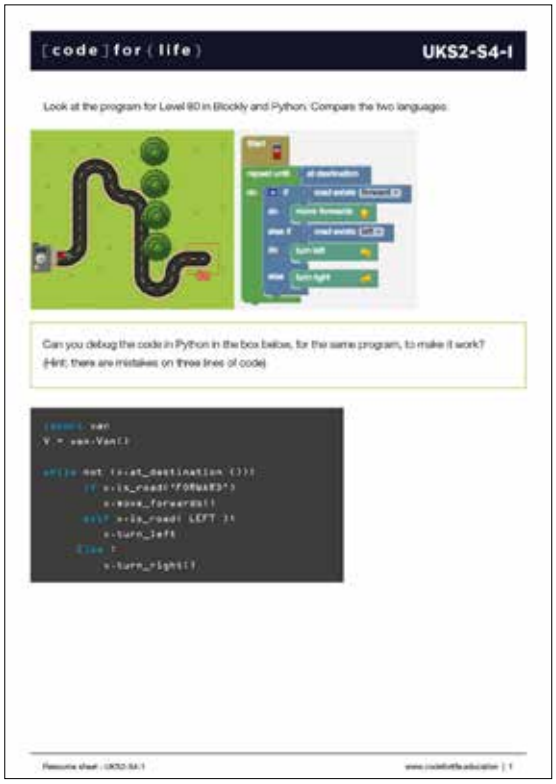


fig S4.4

Extension activity

Remind the children of how they defined **procedures** using Blockly in Session 2.

Show Level 90 on the IWB and point out the Blockly **Define** block. *[fig S4.5]*

Give each pair a copy of the Level 90 sheet from the UKS2 Levels Guide and ask them to identify a **procedure** which re-occurs at different points of the route.

Ask the children to create the **procedure** in pairs and then discuss how it is written in Python.

```
4
5 def bend():
6   v.turn_right()
7   v.turn_left()
8
```

What can you spot about the Python syntax?

Point out the importance of the brackets and the colon.

fig S4.5