

Python 6: Iteration Part 2

Teaching Resource

Resources

- Slides **Python 6: Iteration 2**
- You will need to either have Python IDLE installed or have access to an online Python IDE. We have used <https://editor.raspberrypi.org/en>
- Activity worksheets are included in this lesson. You will need to distribute these to your pupils
- We have added a walk-through video below

Prior Knowledge

Printing, mathematical operators, data types and input using Python

- Selection in Python using `if...elif...else`
- counted `while` loops in Python



2MB

Python 6 - Iteration 2.pptx

Lesson walk-through

You need to log in to access videos



Learning Objectives

To teach pupils to:

- use what they have learned about `while` loops so far in Python exercises outside Rapid Router
- to write more complex programs by using prior solutions as scaffolds
- to repeat input in a loop for totalling
- to begin to use selection inside a loop

Slide Notes

Starter

Slide 2: Show the six questions on the board and ask pupils to write down their answers. They should work on their own and do their best to answer all questions, even if they are not sure.

Slide 3: Show the answers and ask pupils to correct their own answers. If they are going to keep the sheet, ask them to mark and correct in a different colour so it is obvious what they did know and what they did not.

Activity 1: Investigation

Slide 5: This slide shows the solution to level 122. Discuss this with pupils and ensure they understand why it is fine to re-use the `count` variable - you might find that some used `count1`, `count2` and `count3`. There is no need to introduce unnecessary variables into the solution.

For the rest of this lesson, we will practice Python while loops outside Rapid Router.

Show **slide 6** and ask the pupils how we can use what we did before to answer this new question. The expected output of this new program is:

```
0
1
2
3
4
5
6
7
8
9
10
Finished
```

Slide 7 identifies the code that is repeated. We need to replace this with a print statement but keep the counting there.

Slide 8: On this slide, we change the loop so that it repeats ten times instead of four times. Then we add the `print` statement. Ask pupils where the `print("Finished")` should go. This should only be output once.

Slide 9 shows the correct placement for this line of code. Ensure that pupils understand this. You might like to ask pupils what would happen if it was indented instead. If pupils find this confusing, demonstrate it yourself. E.g.

main.py	Text output
1 count = 0	0
2 while count <= 10:	1
3 print(count)	2
4 count = count + 1	3
5 print("Finished!")	4
	5
	6
	7
	8
	9
	10
	Finished!

Example 1: print("Finished!") is not in the loop so it is not repeated.

main.py	Text output
<pre>1 count = 0 2 while count <= 10: 3 print(count) 4 count = count + 1 5 print("Finished!")</pre>	<pre>1 Finished! 2 Finished! 3 Finished! 4 Finished! 5 Finished! 6 Finished! 7 Finished! 8 Finished! 9 Finished! 10 Finished!</pre>

Run ►

Example 2: print ("Finished!") is in the loop so it is repeated each time

Activity 2: Exploring the while loop

Slide 11: Once you have explored this code, ask pupils to do the **Activity 1: Using Python** exercises on the worksheet. The solutions follow on **slides 12–14**. Avoid showing these solutions until you have checked pupils' work.

Activity 3: Input and Totalling with the `while` loop

Slides 15–18 start to explore how we can use what we have done to ask the user to enter five numbers and output their total. Work through these slides, asking pupils for answers to the questions posed. Pupils should try the code on slide 18.

Slides 19–24 explore how they can further extend this code to total up these numbers. Once you have worked through the slides, ask the pupils to update their programs and test them. If they get stuck, ask them to carefully compare their code to the example given.

Slides 25–26 explore some common errors with loops. See if pupils can spot the problems. The output is there to help them.

Slide 27: Once pupils are feeling confident with this work, ask them to do the exercises. The answers follow on slides **28–30**. Again, check pupils' work before showing these answers.

Plenary

Slide 31: The exit ticket shows some Python code with five errors. Ask pupils to work in pairs to find the errors.

The errors are:

- `count` has not been initialised before the loop
- `total = 0` must come before the loop
- You must ask for the number to be input before you can add it to the total
- You must store `count + 1` back into the `count` variable so it changes
- The final `print` statement should output `total` and not `number`