

Week 5: Python Programming - Loops and consolidation

In this week's session we are continuing to look at loops and put all the skills we have learned so far together.

Task 1: Predict In pairs, look at the program below and write out what you think might happen when it runs.

```
def starter():
    number = 0
    while number <= 5:
        print("Hello")
        number = number + 1
    print("Goodbye")

starter()
```

What would you expect the computer to do? Write the output exactly as you think it will appear.

Task 2: Run Download and run the program and see if it does what you think it might do. You will find it at <http://teachinglondoncomputing.org/ks3-week-5>

Did the program run as you predicted? _____

What were the differences? _____

Task 3: Trace the code

Carry out the tracing exercise given on the accompanying sheets.

This method is called TRACS and involves:

- Identifying the expressions
- Showing the flow of control in the program
- Hand-tracing the program

Does this help you to understand how the while loop works?

Would this approach be useful for your students?

Task 4: Exercises

As you do these exercises, make a note of any errors that you come across in the notebook and how you fix them.

Part 1

Write some small programs using the while loop and with conditions using different operators. Remember the operators are as follows:

- == is equal to
- != is not equal to
- > is greater than
- < is less than
- >= is greater than or equal to
- <= is less than or equal to

<p>1. Write a program that writes "Today is Monday" one hundred times using a while loop. Compare this to the way that we might have done this last week with a for loop.</p>	<pre>Today is Monday Today is Monday Today is Monday Today is Monday and so on</pre>
<p>2. Write a program that asks the user for the name of the recent storm and keeps asking until they put in the name "Doris". To the right is some code for the input statement to get you started</p>	<pre>print("What is the name of the recent storm? ") answer = input() OR answer = input("What is the name of the recent storm? ")</pre>
<p>3. Write a program that asks the user for a number until they put in 99. When they put in 99 the program stops.</p>	<pre>>>> exercise3() Enter a number ...55 Enter a number ...24 Enter a number ...13 Enter a number ...99 Thank you - 99 was the number I was looking for!</pre>
<p>4. Write a function bored() using a while loop that has the output shown to the right. Start with</p> <pre>def bored():</pre>	<pre>>>> bored() Are you bored yet? n Are you bored yet? n Are you bored yet? n Are you bored yet? n Are you bored yet? n Are you bored yet? n Are you bored yet? n Are you bored yet? y Got to you in the end!!</pre>

<p>5. Write a program that prints 1 to 10. You can do this with a while loop or a for loop. Try to do it both ways.</p>	<pre>>>> exercise5() 1 2 3 4 5 6 7 8 9 10</pre>
<p>6. Now change your program from (3) so that it prints the numbers from 10-1.</p>	<pre>>>> exercise6() 10 9 8 7 6 5 4 3 2 1</pre>
<p>7. The program to the right asks the user for a password. When does the program stop? Run it and see.</p> <p>Next add some code to count how many tries the user has had and print that out at the end of the program.</p> <p>How else might you improve this program?</p>	<pre>def logging_in(): print("Enter your password on the next line ") password = input() while password != "secret": print("That password is not the one stored ") print("Try again!") password = input() print("Yes that's the correct password!")</pre>
<p>8. Improve the quiz you wrote in week 3 by adding a while loop. Keep asking the question until the user gets it right.</p>	<pre>>>> question() How many cities are there in England? 45 No,that's not correct Try again 50 No,that's not correct Try again 51 That's correct! >>> </pre>

Check your answers at <http://teachinglondoncomputing.org/ks3-week-5>

Part 2: ASCII and Python

Try the following statements out in the Python shell (not in a new file).

```
>>> ord("a")
97
>>> ord("A")
65
>>> chr(97)
'a'
>>> chr(65)
'A'
>>> chr(66)
'B'
>>> chr(67)
'C'
```

ord() and chr() are functions that you can use to work out the ASCII code in decimal for characters that are entered in the keyboard.

Extension tasks

1. Write a program that will write out all the letters from "a" to "z" using the chr function.
2. Write a program that will ask the user for a character and output its ASCII value.

Key concepts to remember

Term	What it means	How to do it in Python
Iteration	A loop. There are two loops – for loops (a set number of repetitions) and conditional loops or while loops.	<pre>for line in range(4): print("hello") number = 0 while number <= 5: print("Hello") number = number + 1 print("Goodbye")</pre>
Variable	Something you can give a value to and then change it at other times in the program	<pre>name = "Rhiannon" # name is a variable number = 56 # number is a variable</pre>
Selection	Where there is a choice point in the program design and an if statement is used to create more than one possible pathway.	<pre>if answer == "Paris": print("Correct") else: print("Not correct")</pre>
Input/Output	Getting input from the keyboard or outputting something to the screen.	<pre>name = input("What is your name?") print(name)</pre>
Assignment	Where a variable is given a value	<pre>number = 56 # number is assigned the value 56 name = "Rhiannon" # name is assigned the value Rhiannon</pre>