Lab 8: loops continued

Refresher quiz 📝

A(n) __________ loop has no way of ending and repeats until the program is interrupted.

○ infinite

○ timeless

○ interminable

○ indeterminate

Check

◎ ○ ○ ○

I am confused

1. EMAIL VALIDATION 📧

Write a program that asks for a user's email address and validates that it is in the correct format. If the address entered is not valid the program should continue asking for the email address until it is in the correct format. A valid email address should contain the @ symbol and it must end with ".ie".

FOR LOOP REVIEW 🔁

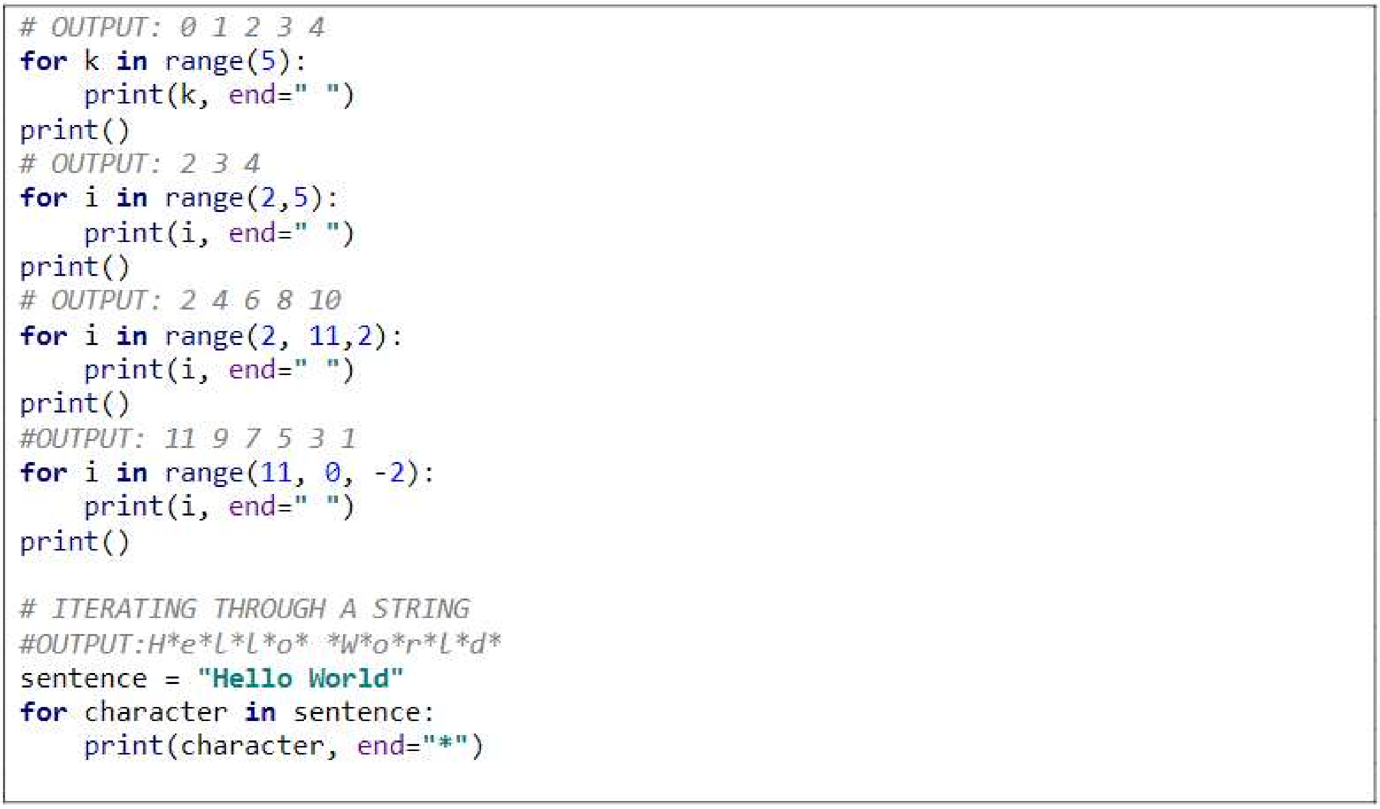
Recall last week we talked about for loops. These loops iterate through a set of values, and on each iteration of the loop a variable 'becomes' each one of the values. The following syntax shows how we typically use for loops:

```
for variable in tuple/string/range:
      statement(s)
```

We also discussed the `range()` function which generates a sequence of numbers, these are often used in for loops. We provide the start, stop and step value; the range function generates a sequence of number that begin at the start value, iterate in increments of step, and finish before the stop value.

```
range(start, stop, step)
```

The following examples should help you to start writing for loops:



```
# OUTPUT: 0 1 2 3 4
for k in range(5):
    print(k, end=" ")
print()
# OUTPUT: 2 3 4
for i in range(2,5):
    print(i, end=" ")
print()
# OUTPUT: 2 4 6 8 10
for i in range(2, 11,2):
    print(i, end=" ")
print()
#OUTPUT: 11 9 7 5 3 1
for i in range(11, 0, -2):
    print(i, end=" ")
print()

# ITERATING THROUGH A STRING
#OUTPUT:H*e*l*l*o* *W*o*r*l*d*
sentence = "Hello World"
for character in sentence:
    print(character, end="*")
```

2. COUNTING 🖐

1. Write a **for** loop to iterate from 1 to 10, printing the number on each iteration with spaces between each as follows: 1 2 3 4 5 6 7 8 9 10
2. Write a **for** loop to iterate from 1 to 100 in steps of 2, printing the number on each iteration with spaces between each as follows: 1 3 5 7 9 11 13 15…….
3. Write a **for** loop to print the even numbers in descending order from 20 down to 1.

3. FOR LOOP STRING ITERATION 💬

1. Write a program to read a sentence from the user. The program should count and display the number of vowels in that sentence.
2. Add code to count and display the number of letters in the sentence. Approach: iterate through the string character by character. Use **char.isalpha()** on each character in turn to determine if it is a letter. If so, add 1 to the counter. If not, do nothing. After the loop, print the number of letters.

3. Add code to count and display the number of punctuation marks in the sentence This necessitates a second **if** inside the loop but you can't use **.isalpha()** this time. So what do we do? Well Python gives us **punctuation** which is a string containing all the punctuation characters !"#$%&'()*+,-./:;<=>?@[\]^_`{|}~ so you can use something like: **if char in string.punctuation** to determine if a character is a punctuation mark. Note: import string library

4. REVERSE A STRING 🔙

Write a program that accepts a word from the user and prints the reverse using a for loop, e.g. if user inputs “hello”, then output would be “olleh”.

5. NESTED LOOP 🐣

Write a program that asks the user to enter a single integer. It should then print a triangle of that size specified by the integer so that each row in the triangle is made up of the integer displayed.

```
1
2 2
3 3 3
4 4 4 4
5 5 5 5 5
```

6. FOR LOOP WITH CONTINUE 🔝

1. Write a **for** loop to iterate through the numbers between 1 and 234, printing only those numbers divisible by 8. (Iterate 1 to 234 inclusive use an if statement to print the number if divisible by 8).
2. Modify the previous code so instead of going from 1 to 234, you ask the user where they would like to begin and end. This requires 2 inputs and you should modify the **range**() function to use these inputs.
3. Use a **continue** inside an **if** statement inside a **for** loop to write a program that prints all the numbers from 0 to 100 except for those divisible by 3.

7. GROUP EXERCISE 🤸 (please wait for me to initiate this exercise)

First, please complete this survey based on your experience of the first in lab assessment **https://vevox.app/#/m/114729789** (https://vevox.app/#/m/114729789)

In exercise 3, you wrote code to analyse the content of a string and count the number of alphabetic characters and the number of punctuation characters. If you were a lecturer and you were in charge of evaluating student's code how would you assign marks for this exercise?

What types of skills are you expecting the student to demonstrate in this exercise? The table below shows the expected Learning Outcomes from this module, which Learning Outcomes are being explored in this exercise?

SOFT6018 Learning Outcomes	
LO1	Determine the outcome of programs, written in a high-level programming language, that utilise basic programming concepts and constructs.
LO2	Implement solutions to programming tasks by identifying necessary variables and constants and choosing appropriate data types for these variables and constants.
LO3	Choose appropriate sequential, conditional and iterative constructs for a given task.
LO4	Implement solutions to programming tasks that require files.
LO5	Test and debug programs developed.
LO6	Document code using best practices and conventions.

You have 10 marks to give for successful completion of this exercise, create a rubric for these 10 marks. Write your completed rubric down and clearly define each gradable component and the marks assigned to it.

Once you have completed your rubric discuss it with your neighbour. After your discussion use the rubric to evaluate your neighbour's code. Ensure you point out to good things that they did that gained them marks. Make a note of the improvements they could make in future iterations of their code. You should also let the student know the Learning Outcomes that they have demonstrated.

EXTRA CHALLENGES ❓

Challenge 1

Write a program that asks the user for a number, n, and displays the factors of that number, excluding n itself.
Sample run:

```
What is the number? 12
The factors of 12 are:
1
2
3
```

```
4
6
```

Recall that modulus (%) operator allows you to test for factors. If number % 3 is 0, then 3 is a factor of number.

Challenge 2

Ask the user for several numbers - the choice of loop is your. You can ask the user to specify the number of times they want to enter a number in advance of the loop or you can ask them at the end of each loop iteration if they want to continue.

- Read that many integer numbers from the user.
- Determine if all numbers are zero.
- Determine if *any* numbers are zero.
- Determine if all numbers are negative.
- Determine if *any* of the numbers is negative.
- Determine if *any* of the numbers is an odd number.
- Determine if any of the numbers is a square of another number (to Challenge 2 first).

It might help to count the number of zeros, count the number of negative numbers, count the number of off numbers etc.