Guided Task 3: Iteration

Objective

In this guided task, you'll work with iteration control flow, using while statements in Python.

There are six parts to this task. If you finish them all, move onto the extension at the end, in which you'll practise using for loops. If you don't have time to do the extension, don't worry – we'll cover this in the next guided task.

Instructions

Part 1 - Squares

1. Add a new file: Squares.py to your existing Solution and make it the startup file.
2. Write a while loop that starts at 1 and ends at 100.
3. Calculates and display each number and its square.
4. End the loop if that square is bigger than 2000.
5. Save and run.

Part 2 - Factorial

1. Add a new file: Factorial.py and make it the startup file.
2. Ask the user to input an integer.
3. Display the number's factorial using a while loop.

   Note: the factorial of a number is that number multiplied by all the preceding numbers.

   The factorial of 5 is = 5 x 4 x 3 x 2 x 1 = 120

   Or, if you like = 1 x 2 x 3 x 4 x 5
4. Save and run.

Part 3 - Investment

1. Add a new file: Investment.py and make it the startup file.
2. Calculates how many years it will take an initial investment of £100 to grow to a target value of £1000 if the interest rate is 10%.
   Tip: Don't start writing code until you've got a plan of action!
3. Save and run.

4. Make your calculation more usable by inputting the following values:
    - Initial investment
    - Target value
    - Interest rate
5. Save and run.

Part 4 – Input an integer between two limits

In this part, you'll ask the user to input an integer between a minimum and maximum values.

If the user fails to enter an acceptable value for three times, then you stop asking!

1. Add a new file: getInt.py and make it the startup file.
2. Create two variables for the *min* and *max* values.
   Set two values for these variables such as 1 and 100.
3. Write a while loop that attempts to get an integer from the user between the limits of min and max values.
4. If the user has tried three times and fails then print None.
   If a valid value is entered, just end the loop and print its value.

   Note: None is a valid keyword in Python, which stands for Null.

Part 5 - Count vowels

1. Add a new file: CountVowels.py and make it the startup file.
2. Input a word (a string).
3. Count how many vowels are in the word.

   Tip: You can scroll through every character of a string using its index.

   For example, if word = 'hello' then word[0] is the letter h and word[1] is the letter e.

   Use the len() function to find the length of a string. For example, in the above example, len(word) is 5.

   Use simple if statement / s to detect if the character is 'a', 'e', 'i', 'o' or 'u'.

   Every time you find a vowel, you must increase a counter (an integer variable).

   In the next chapter, (Lists) you'll discover a much easier way of performing this task.
4. Save and run.

Part 6 - Exam average

1. Add a new file: ExamAverage.py and make it the startup file.
2. Code a program that:
    a. Has code to calculate the average of three exam marks
    b. If the average mark is o >= 65, output a 'Pass'
    c. If it is < 65, output a 'Fail'
3. In the main body of the program input the marks for a student for Math, English and ICT exams.
4. Marks should be an integer between 0 and 100.
    a. Use a for loop until the user enters a valid mark.
    b. Calculate and display their average mark and overall result.
    c. Please also display the average mark and print out the average.
5. Save and run.

Extension – Using FOR loops

Objective

In this task, you'll gain some experience in the use of iteration control flow using FOR statements in Python. You'll also use these in the next guided task.

There are two parts to this task.

Part 1 - Squares

1. Add a new file: Squares.py and make it the startup file.
2. Write a loop that starts at 1 and ends at 100.
3. Calculates and displays each number and its square.
4. End the loop if that square is bigger than 2000.
5. Save and run.

Part 2 - Factorial

1. Add a new file: Factorial.py and make it the startup file.
2. Inputs an integer.
3. Display the number's factorial.

    Tip: the factorial of a number is that number multiplied by all the preceding numbers.

The factorial of 5 is = 5 x 4 x 3 x 2 x 1 = 120
Or, if you like = 1 x 2 x 3 x 4 x 5

4. Save and run.

Well done, you've completed this guided task!