Guided Task 4: Lists

Objective

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

There are three parts to this task.

Instructions

Part 1 – Practise using Lists

You'll work with the following List:
ages =

```
[12,18,33,84,45,67,12,82,95,16,10,23,43,29,40,34,30,16,44,69,70,74,38,65,36,83,50,11,7
9,64,78,37,3,8,68,22,4,60,33,82,45,23,5,18,28,99,17,81,14,88,50,19,59,7,44,93,35,72,25
,63,11,69,11,76,10,60,30,14,21,82,47,6,21,88,46,78,92,48,36,28,51]
```

1. Record the length of the ages List in a variable (you'll need it later).
   Display the length.
   Test your code.

2. Display these ages one on each line:
   Tip: Use a for loop to read each number.

   Test your code.

3. Add one year to every age.

   Tip: **ages[n]** is the nth element of ages.
   To increase, for example, element 2, you may do ages[2] += 1.
   len(ages) will return the length of the ages List.

   Redisplay ages to test your code.

4. Our code only takes into account those people in the age range of 16-65 (inclusively).
   Please delete all ages that are outside this range.

   Tip: There are other ways of achieving this task, but one way is to use a for loop that uses the len() function to examine each item and then use the del() function to remove an item at certain index.

   Redisplay ages to test your code.

5. Display the count of 16-25 year olds.

   Test your code.

6. Invoke the sort method on the ages List.

   Tip: Use this line of code: **ages.sort()**

   Display the ages List to make sure they are sorted.

7. What proportion of ages fall between 16-25?
   Test your code by displaying this value.

Part 2 – Count vowels

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

   Tip: You can scroll through every character of a string in the same way as you do with numbers in a range() function.

   Use a 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).

   Or, even better, you may consider using the 'in' keyword.
4. Save and run.

Part 3 - Time calculator

Your task is to write code for a range of calculation on times.
Times should be stored and inputted as strings in the format DD:HH:MM.
Days, Hours and Minutes should be stored as integers.

1. Add a new file called TimeCalculator.py and make it your start up file.
2. Input two day-time strings from the user.
   Your code will do calculations until the user selects Option 9 (see below).
3. Print a main menu:

```
Time Calculator
1-   Add 2 times
2-   Find the difference between 2 times
3-   Convert to Hours
4-   Convert to Minutes
5-   Convert Minutes to Time
6-   Convert Hours to Time
7-   Convert Days to Time
8-   Exit

     Enter an option:
```

Option 1 and 2
Input two time strings

Option 3, 4 and 5
Enter only one time string

Options 6 and 7
Enter a single integer

4. Save and run.

   Tip: there are several ways you can write code for this task.
   You can use the **split()** function to split the string into a List and process the day, hour and the minute components.

You may also want to investigate the mod operator (%) to find remainder of a division.
To get the integer part of a float, you may cast it to integer. For example:

```
print(25/24)          1.0416667
print(int(27/24))     1
print (27 % 24)       3
```

Well done, you've completed this guided task!