Exercise sheet 1

Points: / 20

Python Introduction 2022

by David Rieger

26. November 2022

Questions:

In this exercise some basic knowledge will be asked. Answer the questions with a few sentences in the *exercise_01.py* file!

(a) Which programs and resources do you need, to write and execute python code?

(b) Which type of language is python and what are the differences to C++ for example?

(c) How do you define variables in python? (Check answer (b))

(d) What's the difference between tuples, lists, sets and dictionaries?

(e) What are modules and how can you import and use them?

(f) What loop and statements are there in python?

0.1 Exercise 1:

Here you have to code in python. The files *Exercise_1.py* and *resources.py* are given to you. Complete the code with the following instructions!

(a) Import the *factorial* function from the *math* module

(b) Import the variables *ARR* and *NUMBERS* from the *resource.py* file. What makes these variables special? *Hint: it has to do with UPPERCASE*

(c) Code a function called *isValid* where an argument is given. The function should check if all contents of the argument are characters. If not, then a phrase x is not a character type! where x is the data, that isn't a character. Check out formatted strings from the lecture!

(d) Code the main function, so the code runs as a script! Check out this resource for further information!

(e) Code a loop function, that loops through *NUMBERS* and calculates the factorial for each number. Let the factorial and number print out in the console like this: The factorial of number x is: factorial(x)

0.2 Exercise 2:

You have to finfish *Exercise_2.py*. The program should do arithmetics according to the user input. You can select the arithemtic cas with an input beteween 1 and 4. Where 1 is addition, 2 is subtraction, 3 is multiplication and 4 is division. Also you can exit the program if the user inputs 'x'. The arithmetic functions as well as the evaluation for a valid input is already coded. You just have to finish the input cases and the corresponding print instructions.
Hint: Use If-Else-Statements and formatted strings!

0.3 Exercise 3:

Write a program called *Exercise_3.py* that calculates the row sums, the column sums and the total sum of a 2D matrix that was entered by the user. The program should work as follows:

- The 2D matrix should be implemented as a nested list of integer numbers.
- To extract the matrix from the console input, the following actions must be performed:
  - Until x is entered, the user can enter entire rows
  - Such an entire row input must be follow the format *int_1 int_2 ... int_n*, where int_i are integer numbers seperated by a single space character (you can assume correct user input w.r.t. the data types).
  - The individual integers must then be extracted and stored in a (row) list.
  - If the user enters rows with different sizesl, extend all shorter rows with 0, so that all rows are equally sized afterwards, i.e., apply 0-padding to the end of shorter rows.
  - All row lists must then be collected inside another list, which will then be the matrix/nested list.
- Using the matrix, calculate the row sum, i.e., for each row, compute the sum and store the result in a list (row sum list).
- Using the matrix, calculate the column sum, i.e., for each column, compute the sum and store the result in alist (column sum list).
- Using the matrix, calculate the total sums of all elemts.
- Print a nicely formatted matrix of the form
  $[[r1c1\, r1c2 \ldots r1cn]$

$[rmc1\, rmc2 \ldots rmcn]]$
where ricj indicates the element in the i-th row and j-th column.

- Print the row sums, the column sums and the total sum.

Example input:
Enter row: 1 2 3 4
Enter row: 5
Enter row: 0
Enter row: x

Example output:
```
[[1 2 3 4]
[5 0 0 0]
[0 0 0 0]]
row sums: [10, 5, 0]
column sums: [6, 2, 3, 4]
total sum: 15
```