PCL1 functional Python Exercise(s) 9

Contents

Functional Programming using Python

In this exercise, we are going to practice recursion, lambda and higher order functions.

9.1 – Recursion and lambda

Given numLst = [1, 2, 3, 4, 5]:

a. define a functional fun_list_sum to sum the list using recursion (not with the reduce function as presented on the power point slides).
b. define the same function using lambda

9.2 – Recursion

Recall from F# sessions, where the factorial of an integer number is defined by the following formula:

$$n! = \begin{cases} 1 & \textbf{if } n = 0 \\ n \times (n-1)! & \textbf{if } n > 0 \end{cases}$$

a. Write a Python **recursive_factorial()** function using recursion.

9.3 – Higher-Order, lambda

Write a function to find even numbers (print_even_numbers) given a list of natural numbers:

natural_numbers = [0,1,2,3,4,5,6,7,8,9]

Use lambda and filter() function.

9.4 – Higher-Order, lambda

Given the following:

```
gtg_sales  = [('Coffee', 2018, 525.05),
        ('Juice', 2021, 526.03),
        ('Apple', 2020, 525.12),
        ('Green Tea', 2019, 525.02),
        ('Banana', 2022, 524.08)]
```

Sort the list by year. Use lambda function.

9.5 – Recursion

(Sum series) Write a recursive function to compute the following series:

$$m(i) = \frac{1}{3} + \frac{2}{5} + \frac{3}{7} + \frac{4}{9} + \frac{5}{11} + \frac{6}{13} + \ \ldots \ + \frac{i}{2i+1}$$

Write a test program that displays m(i) for i 1, 2, ..., 10.