MIDDLE EAST TECHNICAL UNIVERSITY, NORTHERN CYPRUS CAMPUS
CNG445 Software Development with Scripting Languages – Worksheet 3

Regular Expressions, Exception Handling, Command Line Arguments, Files Processing, System Programming and Operating System Interface in Python

1. Assuming regular expression package is imported as `from re import *`, find single line Python expressions to convert given strings to requested values. Your expressions should be general, meaning for different inputs with similar nature, they should find expected results. You can use also use basic list operations.

    a. Input, a string denoting a path, as s: '/home/eraslan/445/2018/prj/project.py' Return: The filename, without the directory information: 'project.py'

    b. Input, a string denoting a path, as s: '/home/eraslan/445/2018/prj/project.py'
    Return: A list of directories denoting the directory of the file, without the filename: ['home','eraslan','445','2018','prj']

    c. Input, a string as s: "I doon't waaaant to haaave aannny eeexxaaam"
    Return: String with all repeating a characters are replaced with single one. "I doon't want to have annny eeexxam"

2. Write a program that asks the user to enter their social security number (SSN). The program will check if SSN is entered correctly. If so, it prints the SSN, otherwise, it keeps asking to enter an SSN in a correct form. SSN can be entered as XXX-XX-XXXX or XXXXXXXXX where X represents a digit.

3. Write a program that generates 100 random numbers between 1 and 100 and keep them in a list. The program should then take a number from the user and print the index of the number in the list. If the number is not found in the list, the program should print "Not found". Use the index function to take the index of the number in the list.

    list.**index**(*x*[, *start*[, *end*]])

    Return zero-based index in the list of the first item whose value is *x*. Raises a ValueError if there is no such item.

4. Write a program takes a file name as input and reads the file and creates a dictionary with words as keys and counters as the corresponding values. The first time you see a word, you would add an item to the dictionary. After that you would increment the value of an existing item. Your program should ignore the case of the words, so that "apple" and "Apple" are considered the same. However, words that are actually spelled differently, such as "apple" and "apples" are considered to be different words. Assume that there are no punctuations in the file.

5. Assume that you have a folder with different files with different extensions, such as .docx, .xlsx, .txt. Write a program that takes the path of the folder and creates sub-folders for each file extension and put these files in the appropriate files, for example example.docx should be under the folder called docx.

References:

Downey, Allen (2009) Python for Software Design: How to Think Like a Computer Scientist, New York, NY: Cambridge University Press. ISBN: 978-0521725965