Task: Functions, Lists and Hashing

www.hyperiondev.com

Introduction

Welcome to the Functions, Lists, and Hashing Task!
Overview:

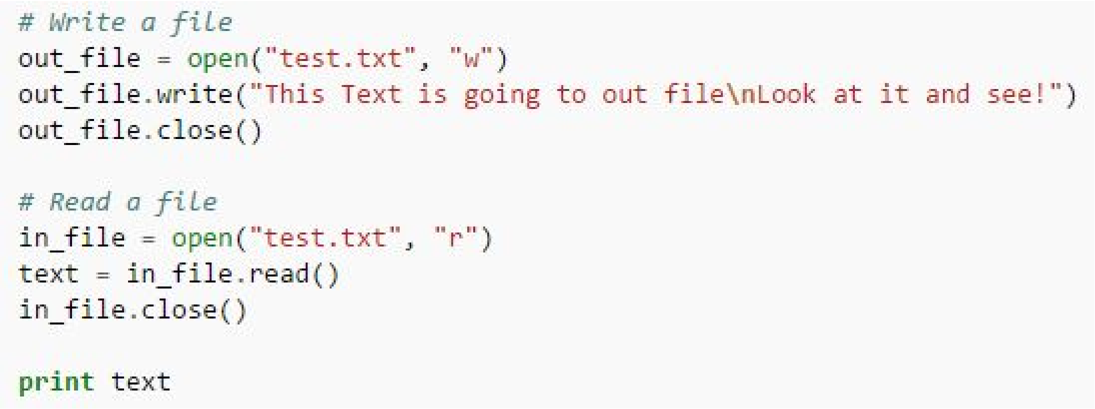
This Task is aimed to ensure that you have a concrete understanding of Strings, Lists and Lists manipulations, as these will be needed for upcoming more advanced tasks. In example.py, you will see examples that deal with lists and operations that can be applied to elements in lists. The Task also re-introduces Functions and how they can be used to compute certain values on list elements and/or text file contents.

- The Hyperion Team

A note from the Hyperion Team...

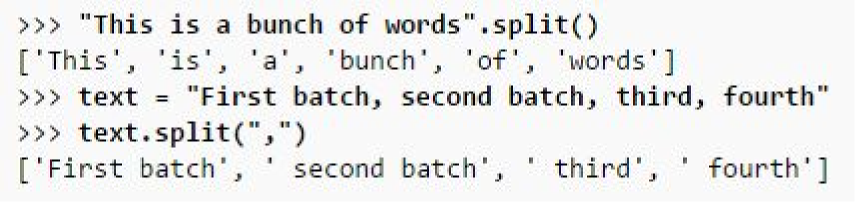
At this point you should have a comprehensive understanding of functions, lists, and dictionaries, as well as their applications. This task will focus on incorporating all of these features to build well catering applications.

You'll also need to recall how to deal with file input and output. A short template of this can be seen below:



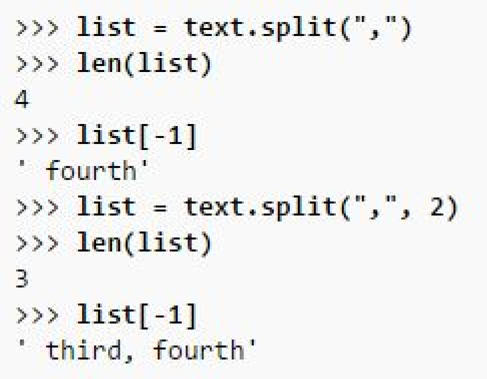
```
# Write a file
out_file = open("test.txt", "w")
out_file.write("This Text is going to out file\nLook at it and see!")
out_file.close()

# Read a file
in_file = open("test.txt", "r")
text = in_file.read()
in_file.close()

print text
```

Further list manipulation will also occur in this task. A helpful string function is the split() method. Here's an example:



```
>>> "This is a bunch of words".split()
['This', 'is', 'a', 'bunch', 'of', 'words']
>>> text = "First batch, second batch, third, fourth"
>>> text.split(",")
['First batch', ' second batch', ' third', ' fourth']
```

When running the above segment, you'll notice how split() converts a string into a list of strings. The string is split by whitespace by default or by the optional argument (in this case a comma). You can also add another argument that tells split()how many times the separator will be used to split the text. For example:



```
>>> list = text.split(",")
>>> len(list)
4
>>> list[-1]
' fourth'
>>> list = text.split(",", 2)
>>> len(list)
3
>>> list[-1]
' third, fourth'
```

You may now go through the example.py file for more information as well as tips for your next task. You should also go through the program in your example programs folder and ensure you can understand what each line of code does

.

Instructions

First read **example.py**, open it using Notepad++ (Right click the file and select ‘Edit with Notepad++’).

- example.py should help you understand some simple Python. Every task will have example code to help you get started. Make sure you read all of example.py and try your best to understand.
- You may run example.py to see the output. The instructions on how to do this are inside the file. Feel free to write and run your own example code before doing this task to become more comfortable with Python.
- You are not required to read the entirety of Additional Reading.pdf, it is purely for extra reference.

Compulsory Task 1

Follow these steps:

- Create a Python file called **amazon.py** in this folder.

- Write code to read the content of the text file **input.txt**. For each line in **input.txt**, write a new line in the new text file **output.txt** that computes the answer to some operation on a list of numbers.

- If the input.txt has the following:

  Min: 1,2,3,5,6
  Max: 1,2,3,5,6
  Avg: 1,2,3,5,6
  Your program should generate output.txt as follows:
  The min of [1, 2, 3, 5, 6] is 1.
  The max of [1, 2, 3, 5, 6] is 6.
  The avg of [1, 2, 3, 5, 6] is 3.4.

- Assume that the only operations given in the input file are min, max and avg, and that the operation is always followed by a list of comma separated integers. You should define the functions min, max and avg that take in a list of integers and return the min, max or avg of the list.

- Your program should handle any combination of operations and any length of input numbers. You can assume that the list of input numbers are always valid integers and that the list is never empty.

Hyperion Development

Optional Task 1

Follow these steps:

- Change your program to additionally handle the operation "px" where x is a number from 10 to 90 and defines the x percentile of the list of numbers. E.g.:

  Input.txt:

  Min: 1,2,3,5,6
  Max: 1,2,3,5,6
  Avg: 1,2,3,5,6
  P90: 1,2,3,4,5,6,7,8,9,10
  Sum: 1,2,3,5,6
  P70: 1,2,3

- Your output.txt should read:

  The min of [1,2,3,5,6] is 1.
  The max of [1,2,3,5,6] is 6.
  The avg of [1,2,3,5,6] is 3.4.
  The 90th percentile of [1,2,3,4,5,6,7,8,9,10] is 9.
  The sum of [1,2,3,5,6] is 17.
  The 70th percentile of [1,2,3] is 2.

Things to look out for:

1. Make sure that you have installed and setup all programs correctly. You have setup **Dropbox** correctly if you are reading this, but **Python or Notepad++** may not be installed correctly.
2. If you are not using Windows, please ask your tutor for alternative instructions.

Still need help?

Just write your queries in your comments.txt file and your tutor will respond.

Task Statistics

Last update to task: 23/12/2015.
Author: Riaz Moola and Jared Ping.
Main trainer: Umar Randeree.
Task Feedback link: Hyperion Development Feedback.