Task: Consolidation - Files

www.hyperiondev.com

Introduction

Welcome to The Consolidation - Files Task!

At this point you should have a comprehensive understanding of string handling and working with external data sources. This task will focus on incorporating these subjects to build well catered applications and ensuring that you have a concrete understanding of strings, and files in Python, as these will be needed for upcoming more advanced tasks.

Connect with your mentor

CONNECT

Remember that with our courses - you're not alone! You can contact your mentor to get support on any aspect of your course.

The best way to get help is to login to www.hyperiondev.com/support to start a chat with your mentor. You can also schedule a call or get support via email.

Your mentor is happy to offer you support that is tailored to your individual career or education needs. Do not hesitate to ask a question or for additional support!

A note from the Hyperion Team...

Information Technology (IT) and Computer Science (CS) are often used interchangeably, but you might be surprised to know that they have two very different meanings. Even specialists with tertiary education in Computer Science, Engineering or related fields sometimes have predetermined (and quite possibly incorrect) ideas about what each of these terms mean. Find out more about the difference between these two terms *here.*

-The Hyperion Team

File Input and Output

A Recap on how to deal with file input and output, which will be needed for this task, can be found 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
```

The split() Method

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

```
print "This is a bunch of words".split()

# prints out ['This', 'is',  'a',   'bunch', 'of', 'words']


text = "First batch, second batch, third, fourth"

print text.split(",")

# prints out ['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:

```
text = "First batch, second batch, third, fourth"

list = text.split(",")

print len(list)

# prints out 4


print list[-1]

# prints out 'fourth'


list = text.split(",",2)

print len(list)

# prints out 3


print list[-1]

# prints out '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

Before you get started we strongly suggest you start using Notepad++ or IDLE to open all text files (.txt) and python files (.py). Do not use the normal Windows notepad as it will be much harder to read.

First read example.py, open it using Notepad++ (Right click the file and select 'Edit with Notepad++') or IDLE.

- 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. Feel free to write and run your own example code before doing the 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

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.

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 mentor for alternative instructions.

Give your thoughts..

★ RATE

Hyperion strives to provide internationally-excellent course content that helps you achieve your learning outcomes. Think the content of this task, or this course as a whole, can be improved or think we've done a good job?

Click here to share your thoughts anonymously.