Hyperion Development

Task: Introduction to Lists and Hashing

www.hyperiondev.com

Introduction

Welcome to the Introduction to Lists and Hashing Task.
Overview:

This task is aimed to ensure that you have a concrete understanding of Strings, Functions and basic List manipulations, as these will be needed for upcoming more advanced tasks. In example.py, you will see examples that deal with 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 as well as dealing with hash maps (otherwise known as dictionaries).

-The Hyperion Team

An interesting and useful data structure is the dictionary. The dictionary is built into Python and other programming languages may have a similar data type equivalent.

A dictionary is a set of key-value pairs. Keys are unique and can be used to access a value associated with it. Strings and numbers can be keys.

A dictionary is defined as follows with braces:

```
>>> d = {}
>>> d
{}
>>> type(d)
<type 'dict'>
>>>
```

Above shows an empty dictionary called ‘d’. However a dictionary can be populated like follows as well:

```
>>> positions ={'Riaz' : 1, 'Tom' : 2}
```

The keys are 'Riaz' and 'Tom' and the values are 1 and 2 respectively for the 'positions' dictionary. To access a value use the key, for example:

```
>>> positions['Riaz']
1
```

positions['Riaz'] gives you 1.

```
>>> positions.keys()
['Riaz', 'Tom']
```

You can also use the keys() method, as seen above, to get a list of all the keys of a dictionary. For example:

positions.keys() gives you ['Riaz', 'Tom'].

Look at the Python documentation for more functions as well as other data structures.

- **The Hyperion Team**

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 new Python file in this folder called **comprehend.py**
- Imagine there's a really rude friend of yours, that always sends emails with all words in capital letters. Your friend also doesn't know how to use a spacebar, so he separates words with the ':' (colon) character. Imagine your friend sends you the message "HI:HOW:R:U:TODAY:".
- Take this message, as a string. Split it into a list of string words and make each word lowercase using a list comprehension on each element. Read here for information on how to make words lowercase.
- Now, edit the first and last word (i.e. the first and last element of your list) to capitalise the first letter of the sentence, and add a full stop to the end of the sentence.
- Print out this new intelligent sentence!

Compulsory Task 2

Follow these steps:

- Create a new Python file in this folder called **hash.py**
- Think about three celebrities/famous people that you know.
- Create a hash called hotMap, where the the KEYS are the name and surname of the person (i.e. a String), and the VALUE for each key is either the string 'hot' or 'not' based on whether you think that person is hot or not!
- Here's my hash:

```
hotMap = {
        'EmmaWatson': 'Hot',
        'JustinBieber': 'Not',
       'LeoDiCaprio': 'Hot',
       }
```

- What does print hotMap['EmmaWatson'] return?

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.
Authors: Riaz Moola, Jared Ping and Sinead Urisohn.
Main trainer: Umar Randeree.
Task Feedback link: Hyperion Development Feedback.