**IS201 Fundamentals of Computing**

**H0P03 Dictionary, Set**

10/10/2020 Developed by Kim Nguyen

Center for Information Assurance (CIAE) @City University of Seattle (CityU)

**Before You Start**

* The directory path shown in screenshots may be different from yours.
* Some steps are not explained in the tutorial**.** If you are not sure what to do:
  1. Consult the resources listed below.
  2. If you cannot solve the problem after a few tries, ask a TA for help.

**Learning Outcomes**

Students will be able to:

* Understand the List Data Type and Tuples in Python
* Write a Python program with functions that can be performed in Lists and Tuples

**Resources**

Matthes, E. (2019). [Python Crash Course: A Hands-On, Project-Based Introduction to Programming, 2nd Edition](https://login.proxy.cityu.edu/sso/skillport?context=146803). No Starch Press. (ISBN 9781593279288)

**Preparation**

1. In Visual Studio Code, open the private repository generated when you accepted the HOP02 assignment (If you cannot find that repository in your machine, you might have not cloned the repo, if so, please do before proceeding).

A screenshot of a cell phone

Description automatically generated

Open the terminal from the VSCode by hitting the control + ~ key, navigate into Module 1 folder using the following command:

>>> cd Module 2

**Dictionary Data Type**

A dictionary is a collection of many values. Unlike List, indexes for dictionaries can use many different data types, it’s called keys, and a key with its associated value is called a key-value pair. Each key is separated from its value by a colon (:), the items are separated by commas and the whole thing is enclosed in curly braces. Keys are unique within a dictionary while values may not be. The values of a dictionary can be of any type, but the keys must be of an immutable data type such as strings, numbers, or tuples.

In the Dictionary functions like accessing values, updating the dictionary, adding new pair and deleting the elements can be performed.

1. Under Module3 create a file **dictionary.py** and type the below code. This program describes how to access, update, add and delete in dictionary.

A screen shot of a smart phone

Description automatically generated

1. In the terminal type the following to check the output for the above code

>>> python3 dictionary.py

A picture containing drawing

Description automatically generated

*Note: If the key you ask for doesn't exist, you'll get an error. More than one entry per key not allowed. Which means no duplicate key is allowed. When duplicate keys encountered during assignment, the last assignment wins.*

Dictionary is unordered. Thus, if you compare two dictionaries with the same content but not the same order, it will return true.

1. Let’s find out what will happen when we compare two lists and dictionary with similar values but different order. Create a file **compare.py** and type the below code.

A screenshot of a cell phone

Description automatically generated

1. In the terminal type **python3 compare.py** to see the output for the above code.

A picture containing drawing

Description automatically generated

The output on the terminal should be **False** and **True** respectively, since order matters in List but not in Dictionary as long as it contains the same values.

**Looping in Dictionary**

A single Python dictionary can contain just a few key-value pairs or millions of pairs. Python lets you loop through a dictionary. You can loop through all of a dictionary's key-value pairs, through its keys, or through its values.

1. Create a file **dict\_for.py** and type the below code. This program explains how multiple dictionaries are combined and used inside loop for accessing the key and value.

A screenshot of a cell phone

Description automatically generated

Type **python3 dict\_for.py** in the terminal for checking the output.

A close up of a logo

Description automatically generated

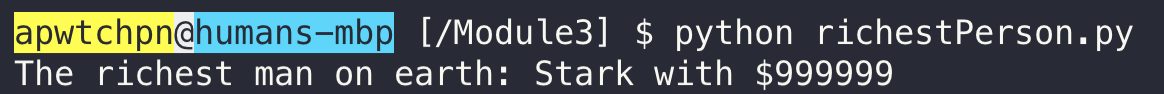
Note: For accessing only keys, use .keys() and for values use .values() in the loop.

2)Let’s create a program to find the richest man from the dictionary and save it as **richest.py**

Text

Description automatically generated

Type “python3 dict\_for.py” in the terminal. The result would show Stark is the richest man on earth.



**Challenge 3: Add to the code above to show the person with the lowest income.**

**Set**

A set is a collection which is unordered and unindexed. In Python sets are written with curly brackets.

Example:

thisset = {"apple", "banana", "cherry"}

**Accessing items:** You cannot access items in a set by referring to an index, since sets are unordered the items has no index. But you can loop through the set items using a for loop, or ask if a specified value is present in a set, by using the in keyword.

**Modify items:** Once a set is created, you cannot change its items, but you can add new items.

**Add items:** To add one item to a set use the add() method. To add more than one item to a set use the update() method.

**Remove items:** To remove an item in a set, use the remove(), or the discard() method.

**Joining two sets:** There are several ways to join two or more sets in Python. You can use the union() method that returns a new set containing all items from both sets, or the update() method that inserts all the items from one set into another.

**Constructor:** It is also possible to use the set() constructor to make a set.

1. Create a file **Set.py** and type the following program for understanding how joining of two sets work and constructor.

A screenshot of a cell phone

Description automatically generated

Type **python3 Set.py** in the terminal for the output. A black sign with white text

Description automatically generated

*Note: Both union() and update() will exclude any duplicate items.*

**Push your work to GitHub**

Open the terminal from the VSCode by hitting the “control” + “~” key and type the following command:

>>> git add .

>>> git commit -m “Submission for Module 3 – Your GitHub Username”

>>> git push origin master