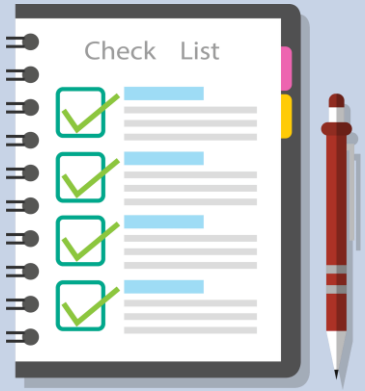
Python – Basics

Digital Lync

INNOVATION - EDUCATION - INCUBATION

Shah Ayub Quadri
aquadri@digital-lync.com

Index



Python Basics – Dictionaries

- Introduction
- Accessing values in Dictionaries
- Working with Dictionaries
- Properties
- Functions and Methods

Dictionary

- Dictionary is a Mapper data type, which consists of 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.
- An empty dictionary without any items is written with just two curly braces, like this: {}.
- 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.

Examples of dictionary :

```
dic={1:10,2:20,3:30,4:40,5:50}
print(dic)
```

```
C:\Users\ravikiran\Desktop\python>a.py
{1: 10, 2: 20, 3: 30, 4: 40, 5: 50}
```

Dictionary

Accessing Values in Dictionary:

- To access dictionary elements, you can use the familiar square brackets along with the key to obtain its value.

For example:

```
dict = {'Name': 'ravi', 'Age': 23}

print ("dict['Name']: ", dict['Name'])
print ("dict['Age']: ", dict['Age'])
```

```
C:\Users\ravikiran\Desktop\python>a.py
dict['Name']:  ravi
dict['Age']:  23
```

- If we attempt to access a data item with a key, which is not part of the dictionary, we get an error.

```
dict = {'Name': 'ravi', 'Age': 23}
print("dict['Address']: ", dict['Address'])
```

```
C:\Users\ravikiran\Desktop\python>a.py
Traceback (most recent call last):
  File "C:\Users\ravikiran\Desktop\python\a.py", line 19, in <module>
    print("dict['Address']: ", dict['Address'])
KeyError: 'Address'
```

Updating Dictionary :

- You can update a dictionary by adding a new entry or a key-value pair, modifying an existing entry, or deleting an existing entry as shown below.

For example:

```
dict = {'Name': 'ravi', 'Age': 23}
dict['Age'] = 35; # update existing entry
dict['Address'] = "HYD"; # Add new entry
print ("dict['Age']: ", dict['Age'])
print ("dict['Address']: ", dict['Address'])
print(dict)
```

```
C:\Users\ravikiran\Desktop\python>a.py
dict['Age']:  35
dict['Address']:  HYD
{'Name': 'ravi', 'Age': 35, 'Address': 'HYD'}
```

Delete Dictionary Elements :

- You can either remove individual dictionary elements or clear the entire contents of a dictionary. You can also delete entire dictionary in a single operation.

- To explicitly remove an entire dictionary, just use the **del** statement.

For example:

```
dict = {'Name': 'ravi', 'Age': 23}
del dict['Name'] # remove entry with key 'Name'
print(dict)
dict.clear()     # remove all entries in dict
print(dict)
```

```
C:\Users\ravikiran\Desktop\python>a.py
{'Age': 23}
{}
```

Properties of Dictionary Keys:

- Dictionary values have no restrictions. They can be any arbitrary Python object, either standard objects or user-defined objects. However, same is not true for the keys.

There are two important points to remember about dictionary keys :

- 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.

```
dict = {'Name': 'ravi', 'Age': 23,'Name': 'khan'}
print(dict['Name'])
```

```
C:\Users\ravikiran\Desktop\python>a.py
khan
```

Dictionary

- Keys must be immutable. Which means you can use strings, numbers or tuples as dictionary keys but something like ['key'] is not allowed.

.

For example:

```
dict = {['Name']: 'ravi', 'Age': 23}
print(dict['Name'])
```

```
C:\Users\ravikiran\Desktop\python>a.py
Traceback (most recent call last):
  File "C:\Users\ravikiran\Desktop\python\a.py", line 18, in <module>
    dict = {['Name']: 'ravi', 'Age': 23}
TypeError: unhashable type: 'list'
```

Built-in Dictionary Functions & Methods:

- **cmp(dict1, dict2):**Compares elements of both dict.

- **len(dict):**Gives the total length of the dictionary. This would be equal to the number of items in the dictionary.

Dictionary

Python includes following dictionary methods **:**

Method	Description
a.clear()	Removes all elements of dictionary *a*
a.copy()	Returns a shallow copy of dictionary a
a.fromkey()	Create a new dictionary with keys from seq and values set to value.
a.has_key(key)	Returns true if key in dictionary dict, false otherwise
a.Items()	Returns a list of *a* (key, value) tuple pairs
a.keys()	Returns list of dictionary a's keys
a.setdefault(key,default =None)	Similar to get(), but will set dict[key]=default if *key* is not already in dict
a.update(b)	Adds dictionary *b*'s key-values pairs to *a*

Assignment - 8

1. Write a Python script to add a key to a dictionary.

Sample Dictionary : {0: 10, 1: 20}
Expected Result : {0: 10, 1: 20, 2: 30}

2. Write a Python script to concatenate following dictionaries to create a new one.

Sample Dictionary :

dic1={1:10, 2:20}
dic2={3:30, 4:40}
dic3={5:50,6:60}
Expected Result : {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}

3. Write a Python program to sum all the items in a dictionary.
4. Write a Python program to remove a key from a dictionary.