

## Handling Dictionary Data

Abdulaziz M. Alqumayzi

(DS-540) – Intro to python for computer science and data science: Learning to program with

AI, Big Data and The Cloud

Colorado State University – Global Campus

Dr. Ernest Bonat

September 27, 2020

## Handling Dictionary Data

### Introduction

In this critical thinking assignment, we are handling dictionary data in Python language. As a conventional dictionary maps words to definitions, a dictionary is an unordered collection that stores key-value pairs that map immutable keys to values. To illustrate, a dictionary associates values with the keys. Every key maps to a particular value. The purpose of this critical thinking is to write a program to provide a Dictionary object with the following tasks that will be applied in the discussion.

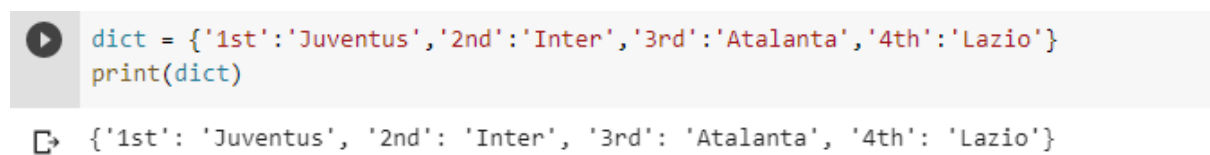
### Discussion

#### Create and print a dictionary.

Creating a dictionary in Python must follow this syntax `dictionaryName = {'key': 'value'}`. 1st is the key and Juventus is the value in the dictionary.

#### Figure 1

*create a dictionary code and output*



```
dict = {'1st': 'Juventus', '2nd': 'Inter', '3rd': 'Atalanta', '4th': 'Lazio'}
print(dict)
```

```
{'1st': 'Juventus', '2nd': 'Inter', '3rd': 'Atalanta', '4th': 'Lazio'}
```

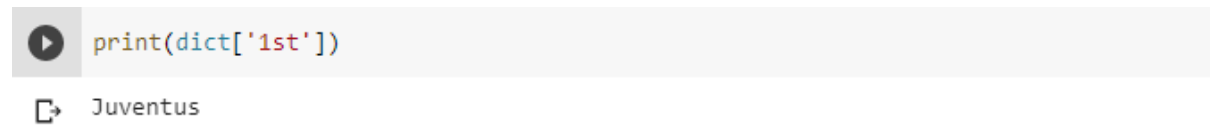
*Note.* In figure 1, code is next to the circle that has a triangle inside it. The output is next to the rectangle with the arrow that coming from it.

#### Accessing items in a dictionary.

Items or values in a dictionary can be accessed via its key, not via index. In the figure 1, example of accessing the first value.

**Figure 2**

*accessing items code and output*



```
print(dict['1st'])
```

Juventus

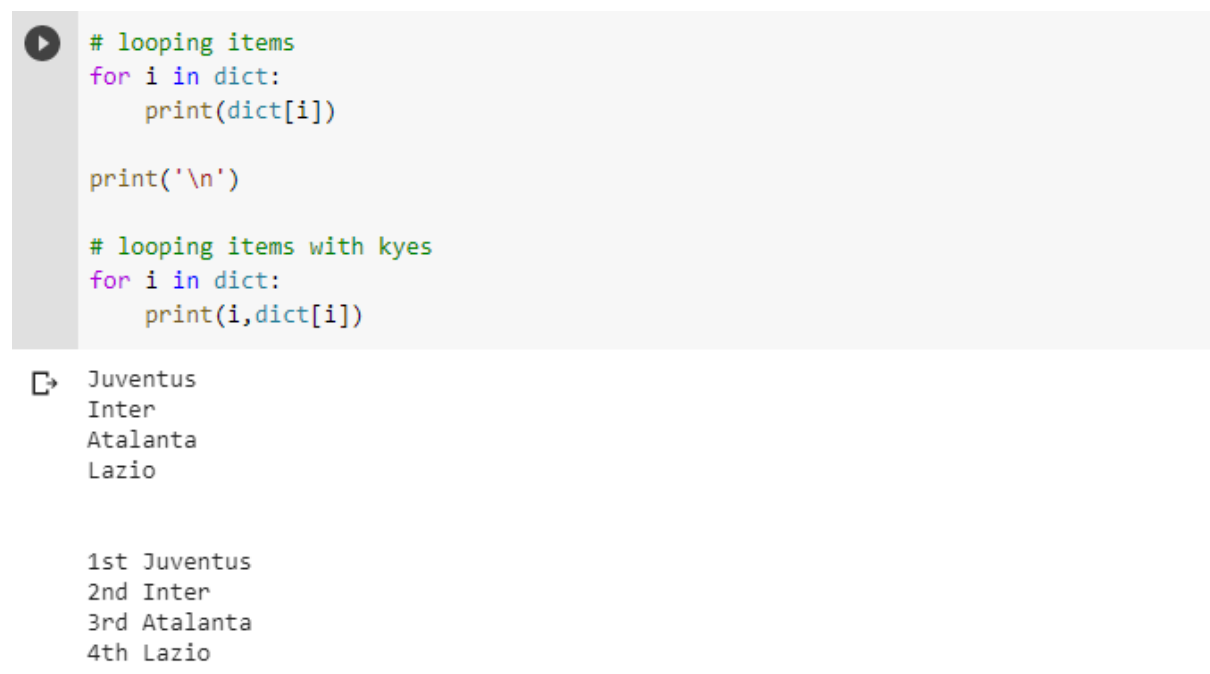
*Note.* In figure 2, the code shows how to access the values by keys.

### Looping items through a dictionary.

Looping items in dictionary can be for keys, values, or both. In the following codes show looping in keys and keys with their values. The code `print('\n')` is to separate the output of the two codes.

**Figure 3**

*looping items codes and outputs*



```
# looping items
for i in dict:
    print(dict[i])

print('\n')

# looping items with kyes
for i in dict:
    print(i,dict[i])
```

Juventus  
Inter  
Atalanta  
Lazio

1st Juventus  
2nd Inter  
3rd Atalanta  
4th Lazio

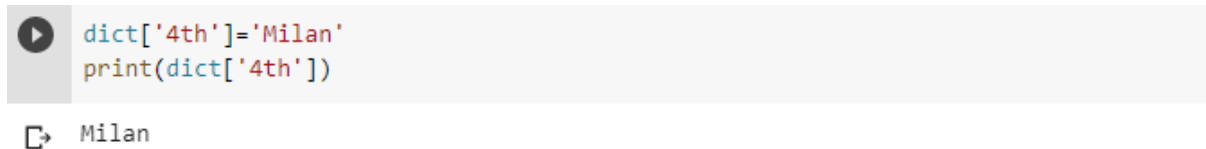
*Note.* In figure 3, the first code shows looping for values. the second code shows looping keys and values together.

**Change a value in a dictionary.**

Change values in a dictionary can be by using keys.

**Figure 4**

*change a value code and output*



```
dict['4th']='Milan'
print(dict['4th'])
```

➤ Milan

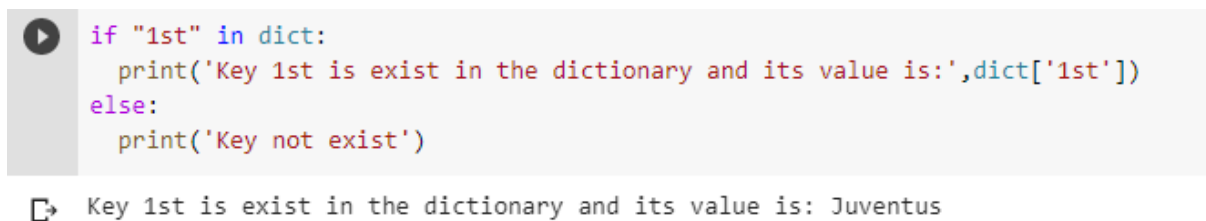
*Note.* In figure 4, the code shows how to change a value in dictionary.

**Check if key exists in a dictionary.**

The following in figure 5 , the code shows if statement used to check if the key exists or not.

**Figure 5**

*check a key in a dictionary code and output*



```
if "1st" in dict:
    print('Key 1st is exist in the dictionary and its value is:',dict['1st'])
else:
    print('Key not exist')
```

➤ Key 1st is exist in the dictionary and its value is: Juventus

*Note.* In figure 5, the code shows how to check a key in a dictionary.

**Print a dictionary length.**

The function len() was used to return the length of the dictionary. The faster way to find the length of a list, set, tuple, or dictionary.

**Figure 6**

*find length code and output*

```
▶ print(len(dict))
```

↳ 4

*Note.* In figure 6, the code shows how to find the length using len() function.

### **Adding Items to a dictionary.**

Items can be added in a dictionary by adding a key and its value together.

### **Figure 7**

*adding an item code and output*

```
▶ dict['5th'] = "Roma"  
print(dict)
```

↳ {'1st': 'Juventus', '2nd': 'Inter', '3rd': 'Atalanta', '4th': 'Lazio', '5th': 'Roma'}

*Note.* In figure 7, the code shows how to add new item in a dictionary.

### **Removing Items in a dictionary.**

The function (del) was used to remove the item. Items in a dictionary removed by the key.

### **Figure 8**

*removing an item code and output*

```
▶ del dict['2nd']  
print(dict)
```

↳ {'1st': 'Juventus', '3rd': 'Atalanta', '4th': 'Lazio'}

*Note.* In figure 8, the code shows how to delete an item in a dictionary.

## References

- Deitel, P. J., & Deitel, H. M. (2020). *Intro to Python for computer Science and data science learning to program with AI, big data and the cloud*. Hudson Street, NY: Pearson Education.
- Kalb, I. (2016). *Learn to program with Python*. Berkeley, CA: Apress.
- Mueller, J., & Emid, A. (2018). *Beginning programming with Python® for dummies®*. Hoboken, NJ: For Dummies, a Wiley brand.