Lecture 10: Dictionary

Swakkhar Shatabda

B.Sc. in Data Science
Department of Computer Science and Engineering
United International University

January 17, 2024

Dictionary

Suppose, we need to store country and their capitals in pairs. <country,capital> are stored as <key,value> pairs in a dictionary.

```
infoDict = { "Bangladesh":"Dhaka",
             "India":"Delhi",
             "Pakistan": "Islamabad",
             "Srilanka":"Colombo",
             "Nepal":"Kathmandu"}
```

- There are five <key,value> pairs in the dictionary.
- Each key corresponds to a country as string and value corresponds to the capital as string.

Dictionary

- A dictionary is an unordered collection which stores key–value pairs that map immutable keys to values.
- A dictionary's keys must be immutable (such as strings, numbers or tuples) and unique (that is, no duplicates).
- Each key is connected to a value, and you can use a key to access the value associated with that key.
- A key's value can be a number, a string, a list, or even another dictionary.

```
print(infoDict)
print(len(infoDict))
```

```
{'Bangladesh': 'Dhaka', 'India': 'Delhi', 'Pakistan': 'Islamab
'Srilanka': 'Colombo', 'Nepal': 'Kathmandu'}
5
```

Accessing Dictionary

```
roman_numerals = {'I': 1, 'II': 2, 'III': 3,
                  'V': 5, 'X': 100}
print(roman_numerals)
print(roman_numerals['I'])
print(roman_numerals['X'])
```

```
{'I': 1, 'II': 2, 'III': 3, 'V': 5, 'X': 100}
1
5
```

```
print(infoDict["Bangladesh"])
print(infoDict["Srilanka"])
```

Creating a dictionary from scratch

- We can start with an empty dictionary. Then add key-value pairs.

```
dict ={}
```

- Assigning a value to a nonexistent key inserts the key–value pair in the dictionary.

```
dict["Dhaka"]=0
dict["Chittagong"]=255
print(dict)
print("length of the dictionary:",len(dict))
```

```
{'Dhaka': 0, 'Chittagong': 255}
length of the dictionary: 2
```

Updating Values for existing keys

```
roman_numerals['X'] = 10
print(roman_numerals)
```

```
{'I': 1, 'II': 2, 'III': 3, 'V': 5, 'X': 10}
```

- You can delete a key–value pair from a dictionary with the del statement.

```
del roman_numerals['III']
print(roman_numerals)
```

```
{'I': 1, 'II': 2, 'V': 5, 'X': 10}
```

Accessing Dictionary - for non existing keys

```
print(infoDict["Afganistan"])
```

```
KeyError: 'Afganistan'
```

- Before accessing we should check if the key is existent or not.

```
if "Afganistan" in infoDict:
    print(infoDict["Afganistan"])
else:
    print("Not in the dictionary")
```

get() - to access values

- Returns its argument's corresponding value.

```
print(infoDict.get("India"))
```

- Returns None if the key is non-existent.

```
print(infoDict.get("USA"))
```

- If you specify a second argument to get, it returns that value if the key is not found.

```
print(infoDict.get("USA","Unknown"))
```

Accessing all items

```
for i in infoDict.items():
    print(i)
```

```
('Bangladesh', 'Dhaka')
('India', 'Delhi')
('Pakistan', 'Islamabad')
('Srilanka', 'Colombo')
('Nepal', 'Kathmandu')
```

Accessing all items

```
for key, value in infoDict.items():
            print(f"Key: {key}",end=" ")
            print(f"Value: {value}")
```

```
Key: Bangladesh Value: Dhaka
Key: India Value: Delhi
Key: Pakistan Value: Islamabad
Key: Srilanka Value: Colombo
Key: Nepal Value: Kathmandu
```

Accessing all keys and values

```
for i in infoDict.keys():
    print(i)
```

```
Bangladesh
India
Pakistan
Srilanka
Nepal
```

```
for i in infoDict.values():
    print(i)
```

update method

```
myDict = {}
myDict.update({"Potato":65, "Onions":80})
print(myDict)
```

```
{'Potato': 65, 'Onions': 10}
```

```
myDict.update({"Onions":130})
print(myDict)
```

```
{'Potato': 65, 'Onions': 130}
```

Processing keys in sorted order

```
for country in sorted(infoDict.keys()):
      print(country, end='   ')
```

```
Bangladesh   India   Nepal   Pakistan   Srilanka
```

A list in a dictionary

Suppose you have a a social network with four people: Shakib, Tamim, Litton and Mashrafe. Shakib is friend to Tamim and Mashrafe. Litton is friend to Tamim and Mashrafe. Tamim is friend to Shakib, Litton and Mashrafe. Mashrafe is friend to all. Now create a dictionary with these people names as keys and as value it will have a list of people that are friends.

A list in a dictionary

Suppose you have a a social network with four people: Shakib, Tamim, Litton and Mashrafe. Shakib is friend to Tamim and Mashrafe. Litton is friend to Tamim and Mashrafe. Tamim is friend to Shakib, Litton and Mashrafe. Mashrafe is friend to all. Now create a dictionary with these people names as keys and as value it will have a list of people that are friends.

```
networkDict={}
networkDict["Shakib"] = ["Tamim", "Mashrafe"]
networkDict["Litton"] = ["Tamim", "Mashrafe"]
networkDict["Tamim"] = ["Shakib","Litton", "Mashrafe"]
networkDict["Shakib"] = ["Shakib","Litton","Tamim"]
print(networkDict)
```

Practice - I

Write a python code that will get a long sentence as input. It will create a dictionary with all the unique words in it as keys and store the frequency of the words in it as value.