2020/2021 First Semester

|  |  |
| --- | --- |
| Course Code | DS540 |
| Course Name | Advanced Python for Data Science |
| CRN | 14045 |
| Assignment type | Critical Thinking |
| Module | 10 |
| Assignment Points | 100 |

|  |  |
| --- | --- |
| Student ID | G200007195 |
| Student Name | Mousa Ageeli |

**Python Dictionary**

### Dictionaries

Python dictionaries are mappings from keys to values, where the keys must be of hashable type; an example dictionary literal is {“ jam”: 1. 65, “ fish”: 2. 89}. We will restrict dictionaries to be homogeneous (keys all of one type, values all of one type) where both keys and values are basic types.

We represent dictionary blocks with two areas of color, the key type represented on the left, and the value type on the right, separated by a vertical saw-tooth bar “pointing” from left-to-right. Two dictionary types match if the colorings for keys and values both match.

Fig. 1 illustrates the construction of a dictionary literal of a single (string-float) pair; this is then dropped into the right-hand-side of an assignment block.

[A clock mounted to the side

Description automatically generated](https://ieeexplore-ieee-org.sdl.idm.oclc.org/mediastore_new/IEEE/content/media/8116722/8120394/8120400/8120400-fig-12-source-large.gif)

**Fig. 1.**Construction of a single-element dictionary literal which is then dropped into the right-hand side of an assignment block

**The project tasks:**

1. This will make a dictionary, which has an at first four key and value

mydict = {  
 "first name": 'Mousa',  
 "last name": 'Ageeli',  
 "age": 33,  
 "SEU NO": 'g200007195'  
}  
print(mydict)

Output

{'first name': 'Mousa', 'SEU NO': 'g200007195', 'last name': 'Ageeli', 'age': 33}

1. Accessing the item of a dictionary by using its key name

age = mydict['age']  
print(age)

Output

33

1. Loop through a dictionary, by using the items() method

for key, val in mydict.items():  
 print(key, "=>", val)

Output

first name => Mousa

SEU NO => g200007195

last name => Ageeli

age => 33

1. Change the value by using its key name:

mydict['age'] = 120  
print(mydict)

Output

{'first name': 'Mousa', 'SEU NO': 'g200007195', 'last name': 'Ageeli', 'car color': 'white', 'age': 120}

1. To check if a any key is exists use the in:

if "age" in mydict:  
 print('yes')

Output

yes

1. To know how a dictionary length, use the len() function.

print(len(mydict))

Output

4

1. Adding an items to the dictionary by using a new index key and add a value to it:

mydict['car color'] = 'white'  
print(mydict)

Output

{'first name': 'Mousa', 'SEU NO': 'g200007195', 'last name': 'Ageeli', 'car color': 'white', 'age': 120}

1. Remove an item with the del operator

del mydict['car color']  
print(mydict)

Output

{'first name': 'Mousa', 'SEU NO': 'g200007195', 'last name': 'Ageeli', 'age': 120}

References

Poole, M. (2017, 9-10 Oct. 2017). Extending the design of a blocks-based python environment to support complex types. Paper presented at the 2017 IEEE Blocks and Beyond Workshop (B&B).