Content 5

**Dictionary and Sets**

Dictionary is a collection of key-value pairs.

**Syntax:**

''' a = {“key”: “value”,

“harry”: “code”,

“marks” : “100”,

“list”: [1,2,9]}

a[“key”] # Prints value

a[“list”] # Prints [1,2,9] '''

##### **Properties of Python Dictionaries**

1. It is unordered
2. It is mutable
3. It is indexed
4. Cannot contain duplicate keys

##### **Dictionary Methods**

Consider the following dictionary,

a = {“name”: “Harry”,

“from”: “India”,

“marks”: [92,98,96]}

1. **items()** : returns a list of (key,value) tuple.
2. **keys()**: returns a list containing dictionary’s keys.
3. **update({“friend”: “Sam”})**: updates the dictionary with supplied key-value pairs.
4. **get(“name”)**: returns the value of the specified keys (and value is returned e.g. “Harry” is returned here)

More methods are available on **docs.python.org**

**Code for Understanding the Dictionary:**

myDict = {  # we can also chnge he name myDict

    "Fast": "In a Quick manner",

    "Name": "Shaalam",

    "marks": (4, 5, 6),  # list

    "anotherDict": {'Nick Name': 'Aamaan'},

    1: 2

}

print(myDict['Fast'])

print(myDict['Name'])

print("Before changing the Values: ", myDict['marks'])

changed = myDict['marks'] = [1, 0]  # changing the Values

print("Values are Changed: ", changed)  # printing the changed values

myDict['marks'] = [9, 7, 8]  # another way of chaning

print("Values are Changed Again: ", myDict['marks'])

print(myDict['anotherDict']['Nick Name'])

print("\n\nUsing the Dictionary Methos")

# printing the key list in dictionary

print("This Dictionary containing keys are: ",myDict.keys())

# knowing the type

print("The Type is: ",type(myDict.keys()))

print("The Conversion into list: ",list(myDict.keys()))  #converting into the list

print("The values present inside the Dictioary are: ",myDict.values())

print("The Conversion into list: ",list(myDict.values()))  #converting into the list

# values and keys printing in an handy way in the form of tuples

print("Displaying both the keys as well as Values: ",myDict.items())

print("\n\nThe updation of the Dictionary....")

# updating Dictionary

print("The Dictionary before updaing: ",myDict)

myDict2 = {

    "Name2": "xyz"

}

myDict.update(myDict2)  #it will give this value to the myDict and update it

print("The Dictionary After updation: ",myDict) #This will be our updated dictionary

print("\n\nDifference between the two value callinng function")

#difference between calling a wethod in an dictionary

print("The value by .get function: ",myDict.get("Name"))

print("The value by Square Bracket function: ",myDict["Name"])

#here you think that both are same but they are not

#here I am giving an key which not present in the Dictionary then

print("The value by .get function: ",myDict.get("Name0"))  #It will give the none

#print(myDict["Name0"])  #while it will throw an error

##### **Output:**

In a Quick manner

Shaalam

Before changing the Values: (4, 5, 6)

Values are Changed: [1, 0]

Values are Changed Again: [9, 7, 8]

Aamaan

Using the Dictionary Methos

This Dictionary containing keys are: dict\_keys(['Fast', 'Name', 'marks', 'anotherDict', 1])

The Type is: <class 'dict\_keys'>

The Conversion into list: ['Fast', 'Name', 'marks', 'anotherDict', 1]

The values present inside the Dictioary are: dict\_values(['In a Quick manner', 'Shaalam', [9, 7, 8], {'Nick Name': 'Aamaan'}, 2])

The Conversion into list: ['In a Quick manner', 'Shaalam', [9, 7, 8], {'Nick Name': 'Aamaan'}, 2]

Displaying both the keys as well as Values: dict\_items([('Fast', 'In a Quick manner'), ('Name', 'Shaalam'), ('marks', [9, 7, 8]), ('anotherDict', {'Nick Name': 'Aamaan'}), (1, 2)])

The updation of the Dictionary....

The Dictionary before updaing: {'Fast': 'In a Quick manner', 'Name': 'Shaalam', 'marks': [9, 7, 8], 'anotherDict': {'Nick Name': 'Aamaan'}, 1: 2}

The Dictionary After updation: {'Fast': 'In a Quick manner', 'Name': 'Shaalam', 'marks': [9, 7, 8], 'anotherDict': {'Nick Name': 'Aamaan'}, 1: 2, 'Name2': 'xyz'}

Difference between the two value callinng function

The value by .get function: Shaalam

The value by Square Bracket function: Shaalam

The value by .get function: None

##### **Sets in Python**

Set is a collection of non-repetitive elements.

S= Set()          # No repetition allowed!

S.add(1)

S.add(2)

# or Set = {1,2}

If you are a programming beginner without much knowledge of mathematical operations on sets, you can simply look at sets in python as data types containing unique values.

##### **Properties of Sets**

1. Sets are unordered # Elements order doesn’t matter
2. Sets are unindexed # Cannot access elements by index
3. There is no way to change items in sets
4. Sets cannot contain duplicate values

##### **Operations on Sets**

Consider the following set:

S = {1,8,2,3}

1. **Len(s) :**Returns 4, the length of the set
2. **remove(8) :**Updates the set S and removes 8 from S
3. **pop() :**Removes an arbitrary element from the set and returns the element removed.
4. **clear() :**Empties the set S
5. **union({8, 11}) :**Returns a new set with all items from both sets. #{1,8,2,3,11}
6. **intersection({8, 11}) :**Returns a set which contains only items in both sets. #{8}

# ses are collection of nonrepeatative elements

set1 = {2, 1, 4, 6, 8, 1}

print("For proof printing the type: ", type(set1))

print("The set is ", set1)  # it will not print the repatative elements

# creating an Empty set

print("\n\nAbout the Empty sets....")

s1 = {}  # This will create an Empty Dictionary but not a set

print("The type is: ", type(s1))  # This will igve a proof of type

s2 = set()  # This will give me an Enpty set

print("The type is: ", type(s2))  # This will give the proof of type

# Methods for set

# For adding value in set

print("\n\nAdding the values at Empty Set...")

s2.add(5)  # it will not be accepted by the set as it is already signed at line 19

s2.add(4)

s2.add(4)  # it will not be accepted by the set as it is already signed above

s2.add(5)

print("THe set will get the provided values: ", s2)

# here if i want to add list and tuple

print("\n\nAdding List and Tuple....")

# s2.add([3,4,5])  #it will through an error because list can be change so it not accepted by set

# s2.add({2:3}) #it will ot accept dictionary also.

s2.add((3, 4, 2, 1))  # Adding of Tuple

print("I can also an Tuple but not the list(as it is changable) and dictionary: ", s2)

# method for obtaining the length of set

print("\n\nMethod for the length of set.....")

print("Our set is s2 = ", s2)

print("The length of the set is: ", len(s2))

# method for reoving an elements

print("\n\nMethod for removing the set Elements.....")

print("Our set s2 = ", s2)

s2.remove(4)    #this will only remove the present values in the set otherwise thee wouls be an eror

print("We are removing 4 then set would like: ", s2)

**Output:**

For proof printing the type: <class 'set'>

The set is {1, 2, 4, 6, 8}

About the Empty sets....

The type is: <class 'dict'>

The type is: <class 'set'>

Adding the values at Empty Set...

THe set will get the provided values: {4, 5}

Adding List and Tuple....

I can also an Tuple but not the list(as it is changable) and dictionary: {(3, 4, 2, 1), 4, 5}

Method for the length of set.....

Our set is s2 = {(3, 4, 2, 1), 4, 5}

The length of the set is: 3

Method for removing the set Elements.....

Our set s2 = {(3, 4, 2, 1), 4, 5}

We are removing 4 then set would like: {(3, 4, 2, 1), 5}