**Dictionary in Python**

**What is a Dictionary?**

Dictionary is a collection data type which contains key value pairs just like a map in other programming languages.

* Dictionary is mutable in nature, which means changes can be made even after declaring a dictionary in python.
* It is unordered and allows duplicate entries, only in the values since the keys has to be distinct.
* The values are accessed using the keys as indexes in a dictionary.
* A dictionary is declared in curly brackets.

**Example:**

mydictionary = { 'key1' : 'value 1' , 'key2' : 'value 2' , 'key3' : 'value 3'}

print(mydictionary)

**Output :**{ 'key1' : 'value 1' , 'key2' : 'value 2' , 'key3' : 'value 3'}

## ****Various Operations in a Dictionary****

Following are the operations we can perform on a dictionary in python.

1. clear
2. copy
3. fromkeys
4. get
5. items
6. keys
7. popitem
8. pop
9. setdefault
10. update
11. values

## ****Need of Sorting a Dictionary****

* A dictionary has a O(1) search time complexity whereas a list has a O(n) search time complexity, which makes dictionary is a viable option wherever necessary.
* A sorted dictionary yields better understanding and clarity in handling the operations.
* Sorting helps in making efficient analysis while working with any data structure.

## ****How to Sort a Dictionary?****

1. Sorting by keys
2. Sorting by values
3. Custom sorting algorithms – string, number
4. Reversing the sorted order

### ****Sorting By Keys****

* We can use the in-built sorted function which will take any iterable and return a sorted list.
* We can use the keys in order to get a sorted dictionary in the ascending order.

**Example:**

a = {1:2 ,2:1 ,4:3 ,3:4 ,6:5 ,5:6 }

#this will print a sorted list of the keys

print(sorted(a.keys()))

#this will print the sorted list with items.

print(sorted(a.items()))

**Output:**[1,2,3,4,5,6]

[(1,2),(2,1),(3,4),(4,3),(5,6),(6,5)]

### ****Sorting by Values****

* Just like keys, we can use the values as well.

**Example:**

a = {1:2 ,2:1 ,4:3 ,3:4 ,6:5 ,5:6 }

print(sorted(a.values()))

#this will print a sorted list of values.

**Output:**[ 1,2,3,4,5,6]

### ****Custom Sorting Algorithm – String, Number****

* To perform more complex sorts, we can use other arguments in the sorted method.

**Example:**

day = { 'one' : 'Monday' , 'two' : 'Tuesday' , 'three' : 'Wednesday' , 'four' : 'Thursday' , 'five': 'Friday' , 'six' : 'Saturday' , 'seven': 'Sunday'}

print(day)

number = { 'one' : 1 , 'two' : 2 , 'three' : 3 , 'four' : 4 , 'five' : 5 , 'six' : 6 , 'seven' : 7}

print(sorted(day , key=number.\_\_getitem\_\_))

print([day[i] for i in sorted(day , key=number.\_\_getitem\_\_)])

**Output:**

{'one': 'Monday', 'two': 'Tuesday', 'three': 'Wednesday', 'four': 'Thursday', 'five': 'Friday', 'six': 'Saturday', 'seven': 'Sunday'}

['one', 'two', 'three', 'four', 'five', 'six', 'seven']

['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']

* With the use of other arguments, we are able to sort the dictionary in a best way possible using both strings and numbers.
* We can reverse the order as well, below is an example to reverse the order of the sorted dictionary.

### ****Reversing the Sorted Order****

* We can reverse the order of the sorted dictionary. Following is an example to reverse the order of the sorted dictionary.

**Example:**

a = {1:2 ,2:1 ,4:3 ,3:4 ,6:5 ,5:6 }

print(sorted(a.values() ,  reverse= True))

**Output:**[6,5,4,3,2,1]

* Here, we have discussed how to sort a dictionary in python.
* Dictionary can be a optimized way of dealing with data which involves key value pairs.
* It becomes easier to work on dictionaries since they are mutable in nature and have a search time complexity less than that of a list.