DICTIONARIES

Introduction: : Dictionary is a collection of key-value pairs. Each key is separated from its value by a colon (:), the items are separated by commas, and the whole thing is enclosed in curly braces. An empty dictionary without any items is written with just two curly braces, like this: {}.Keys are unique within a dictionary while values may not be. The values of a dictionary can be of any type, but the keys must be of an immutable data type such as strings, numbers, or tuples.

E.g.: dict = {'Name': 'Zara', 'Age': 7, 'Class': 'First'}

Accessing Dictionaries: To access values from the dictionaries we can use square brackets along with the key to obtain its value .

E.g.: dict1 = {'Name': 'shyam', 'Age': 22, 'gender': 'male', 'work': 'teacher'}
dict1['Age'] #22
dict1['Name'] #shyam

Working with dictionaries: In dictionaries we use items to get all the key value pairs, keys to get all the keys values and values to get all the values from the dictionary.

Items: To access all the key and value pairs items keyword is used. Every key and value pair will be returned by using items keyword

E.g.: dict = {'Name': 'Zara', 'Age': 7, 'Class': 'First'}
dict.items() #dict_items([('Name', 'Zara'), ('Age', 7), ('Class', 'First')])

Keys: Using keys we can access all the keys present in the dictionary.

E.g.: dict = {'Name': 'Zara', 'Age': 7, 'Class': 'First'}
dict.keys() # dict_keys(['Name', 'Age', 'Class'])

Values: Using values we can access all the values present in the dictionary.

E.g.: dict = {'Name': 'Zara', 'Age': 7, 'Class': 'First'}
dict.values() # dict_values(['shyam', 22, 'male', 'teacher'])

Functions and methods: : Python contains functions and methods to perform operations on dictionaries.

Len: Len finds the length of the dictionary and returns the length of the dictionary that is equal to total number of items.

Syntax: len(dict)
E.g: dict1 = {'Name': 'shyam', 'Age': 22, 'gender': 'male', 'work': 'teacher'}
len(dict1) #4

Type: The method **type()** returns the type of the variable. If passed variable is dictionary then it would return a dictionary type.

Syntax: type(dict)

E.g: dict1 = {'Name': 'shyam', 'Age': 22, 'gender': 'male', 'work': 'teacher'}
type(dict1) # <type' dict'>

clear: This method removes all the elements from the dictionary and returns an empty dictionary.

Syntax: dict.clear()
E.g: dict1 = {'Name': 'shyam', 'Age': 22, 'gender': 'male', 'work': 'teacher'}
dict1.clear()

fromkeys: It creates a new dictionary with set of keys and a value.

Syntax: dict.fromkeys(seq,[values])
E.g: seq=(1,2,3,4,"hello")
dict.fromkeys(seq[,100]) # {1: [100], 2: [100], 3: [100], 4: [100], 'hello': [100]}

Update: Update methods updates key and value of first dictionary to another dictionary

Syntax: dict_1.update(dict_2)

E.g: dict1 = {'Name': 'shyam', 'Age': 22, 'gender': 'male', 'work': 'teacher'}
dict = {'Name': 'Zara', 'Age': 7, 'Class': 'First'}
dict1_update(dict)