* **Que1.What is dictionary?**

**Ans🡪**Dictionaries are used to store data values in key:value pairs.In dict key is a immutable meance we can not change key but we change value.

Each key-value pair maps the key to its associated value. You can define a dictionary by enclosing a comma-separated list of key-value pairs in curly braces ( {} ). A colon ( : ) separates each key from its associated value

A dict is unique ,unoreder and mutable. But Dictionary items are ordered,

As of Python version 3.7, dictionaries are ordered. In Python 3.6 and earlier, dictionaries are unordered.

Dictionaries are written with curly brackets, and have keys and values:.

**Eg.** D1={1:”a”,2:”b”,3:”c”,4:”d”}

## Creating a dictionary

* **Using curly brackets**: The dictionaries are created by enclosing the comma-separated Key: Value pairs inside the {} curly brackets. The colon ‘:‘ is used to separate the key and value in a pair.
* **Using dict() constructor**:  Create a dictionary by passing the comma-separated key: value pairs inside the dict().

## **Que2.** **How we can create dictionaries by multiple ways in python ?**

Ans🡪 we can create dictionary in 2ways.

**1.by using {}**

D1={“Name”,:”Sachin”,”Age”:50}

**2.dict()**

d3=dict(Name=’Sachin’,age=45)

## **Accessing elements of a dictionary-**

## There are two different ways to access the elements of a dictionary.

1. Retrieve value using the key name inside the [] square brackets
2. Retrieve value by passing key name as a parameter to the get() method of a dictionary.

### Get all keys and values

Use the following dictionary methods to retrieve all key and values at once

| **Method** | **Description** |
| --- | --- |
| keys() | Returns the list of all keys present in the dictionary. |
| values() | Returns the list of all values present in the dictionary |
| items() | Returns all the items present in the dictionary. Each item will be inside a tuple as a key-value pair. |

## **Adding and modify items to the dictionary**

We can add new items to the dictionary using the following two ways.

**Using key-value and Using update() method:**

**#** add by using key\_value

d5= {"name": "Madhuri", 'country': "India", "Mobile": 1178464}

print(d5)

d5['Age']=26

print(d5)

# modify or change

d5.update({'country':'USA'})

print(d5)

|  |  |
| --- | --- |
| **Method** | **Description** |
| [clear()](https://www.w3schools.com/python/ref_dictionary_clear.asp) | Removes all the elements from the dictionary |
| [copy()](https://www.w3schools.com/python/ref_dictionary_copy.asp) | Returns a copy of the dictionary |
| [fromkeys()](https://www.w3schools.com/python/ref_dictionary_fromkeys.asp) | Returns a dictionary with the specified keys and value |
| [get()](https://www.w3schools.com/python/ref_dictionary_get.asp) | Returns the value of the specified key |
| [items()](https://www.w3schools.com/python/ref_dictionary_items.asp) | Returns a list containing a tuple for each key value pair |
| [keys()](https://www.w3schools.com/python/ref_dictionary_keys.asp) | Returns a list containing the dictionary's keys |
| [pop()](https://www.w3schools.com/python/ref_dictionary_pop.asp) | Removes the element with the specified key |
| [popitem()](https://www.w3schools.com/python/ref_dictionary_popitem.asp) | Removes the last inserted key-value pair |
| [setdefault()](https://www.w3schools.com/python/ref_dictionary_setdefault.asp) | Returns the value of the specified key. If the key does not exist: insert the key, with the specified value |
| [update()](https://www.w3schools.com/python/ref_dictionary_update.asp) | Updates the dictionary with the specified key-value pairs |
| [values()](https://www.w3schools.com/python/ref_dictionary_values.asp) | Returns a list of all the values in the dictionary |

## **Nested Dictionaries**

A dictionary can contain dictionaries, this is called nested dictionaries.

d6={'Name':

{'F\_name':'Madhuri','L\_name':'Burande'},

'class':{'Python':100,'jva':110}

}

## **Copy a Dictionary**

You cannot copy a dictionary simply by typing dict2 = dict1, because: dict2 will only be a reference to dict1, and changes made in dict1 will automatically also be made in dict2.

There are ways to make a copy, one way is to use the built-in Dictionary method copy().

d7= {"brand": "Ford","model": "Mustang","year": 1964}

d8=d7.copy()

print("d8=",d8)

Another way to make a copy is to use the built-in function dict()

my\_dict=dict(d7)

print("my\_dict=",my\_dict)

## **Loop Through a Dictionary:-**

You can loop through a dictionary by using a for loop.

When looping through a dictionary, the return value are the keys of the dictionary, but there are methods to return the values as well.

d9= {"brand": "Ford","model": "Mustang","year": 1964}

for key,value in d9.items():

print(key,value)

* **Remove Dictionary Items**

**1.pop():-**The pop() method removes the item with the specified key name:

d5= {"name": "Madhuri", 'country': "India", "Mobile": 1178464}

print(d5)

d5.pop('Mobile')

print(d5)

**2.popitem():-**The popitem() method removes the last inserted item (in versions before 3.7, a random item is removed instead):

d1={"Name":"Madhiuri","Age":26,"Class":"Python"}

print(d1)

d1.popitem()

print("popitem=",d1

**3.del:-** The del keyword removes the item with the specified key name:

d2={"Name":"Madhiuri","Age":26,"Class":"Python"}

print(d2)

del d2['Age']

print("del=",d2)

## **Dictionary comprehension:-**

Dictionary comprehension is one way of creating the dictionary where the values of the key values are generated in a for-loop and we can filter the items to be added to the dictionary with an optional if condition.

d2={x:x\*2 for x in range(1,10)}

print(d2)