Dictionary : Container type which is mutable. Can store any number of python objects including other container types. Dictionary consists of pairs (called as items) of keys and their corresponding values. Also known as associative arrays or hash tables.

Create a dictionary

```
sicsr = {"MBA" : 41, "MSc" : 42}                    # {key : value}
```

Each key is separated from its value using a colon and the items are separated using a comma. Keys should be unique whereas values can repeat. The values can be of any type but keys are of immutable types like strings, numbers, or tuples.

If key is repeated, then only last key:value pair is considered as valid.

Accessing elements of a Dictionary

```
InstDet = {“Name” : “SICSR” , “Code” : “0301”, “NoOfProg” : 4, “TotStud” : 1000}

print(InstDet)
print(InstDet['Name'])
```

Updating Dictionary : Dictionary can be updated by adding new item, modify an existing item, or delete an existing item

```
InstDet['TotStud'] = 900            # update existing item
InstDet['City'] = 'PUNE'            # add new item
InstDet['Pin Code'] = 411016        # add new item
```

Deleting an item

```
Temp = InstDet              # create new dictionary
del Temp['City']            # delete item with key 'City'
del Temp                    # delete entire dictionary
```

```
dict = {['Name']: 'Zara', 'Age': 7};        # wrong as key cannot be list. It can be only string, number or tuple (immutable types only)
```

Built-In dictionary functions

```
len(InstDet)            # no of items in the dictionary
type(InstDet)           # type of the object InstDet
str(InstDet)            # converts dictionary to a printable string representation of the dictionary
```

Built-In dictionary methods

```
InstDet.clear()         # Removes all elements of dictionary. Dictionary object remains (empty)
InstDet.copy()          # Creates a shallow copy of the dictionary
```

Example

```
temp = InstDet          # create temp
temp.clear()            # clear all elements of temp as well as InstDet
temp = InstDet.copy()       # create shallow copy of InstDet
temp.clear()            # removes all elements of temp, InstDet remains intact
```

temp.fromkeys() **#** The method fromkeys() creates a new dictionary with keys from seq and values set to value.

Example :

```
seq = ('name', 'code')          # create a sequence
temp = {}                       # create empty dictionary
t=temp.fromkeys(seq)                # create dictionary t using sequence seq
t=temp.fromkeys(seq, 10)            # create dictionary t using sequence seq
```

InstDet.get() # The method get() return a value for the given key. If key is not available then returns default value None.

Example

```
InstDet.get('Name')
InstDet.get('Code')
```

update() # The method update() adds dictionary dict2's key-values pairs in to dict.

Example

```
temp = {'Country' : 'India',}
temp.update(InstDet)
```

Values() #The method values() returns a list of all the values available in a given dictionary.

Example list(InstDet.values())

Keys() #The method keys() returns a list of all the keys available in a given dictionary.

Example list(InstDet.keys())

Items() #The method items() returns a list of all the key:value pairs available in a given dictionary.

Example list(InstDet.items())

Zip() #The method zip() merges to dictionaries

Example

```
keys = ('name', 'age', 'food')
values = ('Monty', 42, 'spam')
d1=dict(zip(keys,values))
```

InstDet.pop('Name') # remove item with key 'Name'

InstDet.popitem() # pops items one by one

```
InstDet.values(0
InstDet.items()
InstDet.keys()
```

Access elements of a dictionary using for loop

```
for i in temp :
        print(i)
        print(temp[i])
```

InstDet.has_key() # The method has_key() return true if a given key is available in the dictionary otherwise it returns a false. May not be available. Version dependability

Example

```
        InstDet.has_key('Name')
```

InstDet.setdefault() # This method return the value associated with the key else it adds new key with None value.

Example

```
        InstDet.setdefault('Name')          #returns value associated with key “Name”
        InstDet.setdefault(“MobileNo”)      #adds item MobileNo to the dictionary
```

```
for k,v in d.items():
        print(k,v)
```

```
line = "I am in a big I am IN a big "
w=line.split()
c={}
for i in w:
        c[i]=c.get(i,0)+1
```