(3.6) Dictionary

1. Introduction

We can use List,Tuple and Set to represent a group of individual objects as a single entity.
If we want to represent a group of objects as key-value pairs then we should go for Dictionary.

Hash Map

Collection of key-value pairs

Duplicate keys are not allowed but values can be duplicated

If we are trying to insert an entry with duplicate key then old value will be replaced with new value.

Hetrogeneous objects are allowed for both key and values

keys are hashable / immutable

values are valid python type

Unordered

insertion order is not preserved

Dictionaries are mutable

Dictionaries are dynamic

indexing and slicing concepts are not applicable

Note:

In C++ and Java Dictionaries are known as "Map" where as in Perl and Ruby it is known as "Hash"

2. How to create Dictionary?

Empty

In [98]:

```
d = {}
print(type(d))

d = dict()
print(type(d))
```

```
<class 'dict'>
<class 'dict'>
```

Adding element

In [99]:

```
d = dict()

d['f_name'] = 'Pankaj'
d['l_name'] = 'Yadav'
d[100] = 'hundred'
d[200] = 'Two hundred'

print(d)
```

```
{'f_name': 'Pankaj', 'l_name': 'Yadav', 100: 'hundred', 200: 'Two hundred'}
```

If we know data in advance then we can create dictionary as follows

d={key:value, key:value}

In [100…

```
d = {'subject':'Python','Topic':'Dictionary','Level':'Easy'}
print(d)
```

```
{'subject': 'Python', 'Topic': 'Dictionary', 'Level': 'Easy'}
```

3. How to access data from Dictionary?

```
We can access data by using keys
```

In [101…

```
d = {'subject':'Python','Topic':'Dictionary','Level':'Easy'}

print(d['subject'])
print(d['Level'])
```

```
Python
Easy
```

If the specified key is not available then we will get KeyError

In [102…

```
print(d['name'])
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
C:\Users\PANKAJ~1\AppData\Local\Temp/ipykernel_7156/673128875.py in <module>
----> 1 print(d['name'])

KeyError: 'name'
```

We can prevent this by checking whether key is already available or not by using

has_key() function or by using in operator.

But has_key() function is available only in Python 2 but not in Python 3. Hence compulsory we have to use in operator.

In [103…

```
if 'name' in d:
    print(d['name'])
```

4. How to update Dictionary?

d[key]=value

```
If the key is not available then a new entry will be added to the dictionary with
the specified key-value pair

If the key is already available then old value will be replaced with new value.
```

In [104…

```
d = {'name':'Pankaj','branch':'cs','degree':'btech'}
print(d)

d['college'] = "ACEIT"
print(d)

d['branch'] = 'CSE'
print(d)
```

```
{'name': 'Pankaj', 'branch': 'cs', 'degree': 'btech'}
{'name': 'Pankaj', 'branch': 'cs', 'degree': 'btech', 'college': 'ACEIT'}
{'name': 'Pankaj', 'branch': 'CSE', 'degree': 'btech', 'college': 'ACEIT'}
```

5. How to delete element from Dictionary?

del d[key]

```
It deletes entry associated with the specified key.
If the key is not available then we will get KeyError
```

In [105…

```
d={100:"Pankaj",200:"Kumar",300:"Yadav"}
print(d)

del d[100]
print(d)

del d[400]
```

```
{100: 'Pankaj', 200: 'Kumar', 300: 'Yadav'}
{200: 'Kumar', 300: 'Yadav'}

---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
C:\Users\PANKAJ~1\AppData\Local\Temp/ipykernel_7156/3556796623.py in <module>
      5 print(d)
      6 
----> 7 del d[400]

KeyError: 400
```

d.clear()

```
To remove all entries from the dictionary
```

In [106…

```
d={100:"Pankaj",200:"Kumar",300:"Yadav"}
print(d)

d.clear()
print(d)
```

```
{100: 'Pankaj', 200: 'Kumar', 300: 'Yadav'}
{}
```

del d

To delete total dictionary.Now we cannot access d

In [107…

```
d={100:"Pankaj",200:"Kumar",300:"Yadav"}
print(d)

del d
print(d)
```

```
{100: 'Pankaj', 200: 'Kumar', 300: 'Yadav'}
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
C:\Users\PANKAJ~1\AppData\Local\Temp/ipykernel_7156/4240557939.py in <module>
      3 
      4 del d
----> 5 print(d)

NameError: name 'd' is not defined
```

6. Important functions of dictionary

In [108…

```
print(dir(d))
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
C:\Users\PANKAJ~1\AppData\Local\Temp/ipykernel_7156/1732628335.py in <module>
----> 1 print(dir(d))

NameError: name 'd' is not defined
```

A. dict()

To create a dictionary

In [109…

```
d = dict()
print(d,type(d))

d1 = dict({'name':'Pankaj','Branch':'CSE'})
print(d1,type(d1))

d2 = dict([('Subject','Python'),('Topic','Dict'),('Level','Easy')])
#It creates dictionary with the given list of tuple elements
print(d2,type(d2))
```

```
{} <class 'dict'>
{'name': 'Pankaj', 'Branch': 'CSE'} <class 'dict'>
{'Subject': 'Python', 'Topic': 'Dict', 'Level': 'Easy'} <class 'dict'>
```

B. len()

Returns the number of items in the dictionary

In [110…

```
print(len(d),len(d1),len(d2))
```

```
0 2 3
```

C. clear()

    To remove all elements from the dictionary

In [111…

```
d={100:"Pankaj",200:"Kumar",300:"Yadav"}
print(d)

d.clear()
print(d)
```

```
{100: 'Pankaj', 200: 'Kumar', 300: 'Yadav'}
{}
```

D. get()

    To get the value associated with the key

d.get(key)

    If the key is available then returns the corresponding value otherwise returns
    None.It wont raise any error.

d.get(key,defaultvalue)

    If the key is available then returns the corresponding value otherwise returns
    default value.

In [112…

```
d = {'name':'Pankaj','branch':'cs','degree':'btech'}
print(d)

print(d.get('name'))
print(d.get('Home'))
print(d.get('Home','India'))
```

```
{'name': 'Pankaj', 'branch': 'cs', 'degree': 'btech'}
Pankaj
None
India
```

E. pop()

d.pop(key)

    It removes the entry associated with the specified key and returns the
    corresponding value
    If the specified key is not available then we will get KeyError

In [113…

```
d = {'name':'Pankaj','branch':'cs','degree':'btech'}
print(d)

print(d.pop('name'))
print(d)
print(d.pop('Home'))
```

```
{'name': 'Pankaj', 'branch': 'cs', 'degree': 'btech'}
Pankaj
```

```
{'branch': 'cs', 'degree': 'btech'}
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
C:\Users\PANKAJ~1\AppData\Local\Temp/ipykernel_7156/258224932.py in <module>
      4 print(d.pop('name'))
      5 print(d)
----> 6 print(d.pop('Home'))

KeyError: 'Home'
```

F. popitem():

```
It removes an arbitrary item(key-value) from the dictionaty and returns it
```

In [114…

```
d = {'name':'Pankaj','branch':'cs','degree':'btech'}
print(d)

print(d.popitem())
print(d)
print(d.popitem())
print(d)
```

```
{'name': 'Pankaj', 'branch': 'cs', 'degree': 'btech'}
('degree', 'btech')
{'name': 'Pankaj', 'branch': 'cs'}
('branch', 'cs')
{'name': 'Pankaj'}
```

In [115…

```
d = {}
print(d.popitem())
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
C:\Users\PANKAJ~1\AppData\Local\Temp/ipykernel_7156/2731748297.py in <module>
      1 d = {}
----> 2 print(d.popitem())

KeyError: 'popitem(): dictionary is empty'
```

G. keys()

```
It returns all keys associated with dictionary
```

In [116…

```
d = {'name':'Pankaj','branch':'cs','degree':'btech'}
print(d.keys())
```

```
dict_keys(['name', 'branch', 'degree'])
```

H. values()

```
It returns all values associated with the dictionary
```

In [117…

```
d = {'name':'Pankaj','branch':'cs','degree':'btech'}
print(d.values())
```

```
dict_values(['Pankaj', 'cs', 'btech'])
```

I. items()

```
It returns list of tuples representing key-value pairs
```

In [118…

```
d = {'name':'Pankaj','branch':'cs','degree':'btech'}
print(d.items())
```

```
dict_items([('name', 'Pankaj'), ('branch', 'cs'), ('degree', 'btech')])
```

J. copy():

```
To create exactly duplicate dictionary(cloned copy)
```

d1=d.copy();

In [119…

```
d = {'name':'Pankaj','branch':'cs','degree':'btech'}
d1 = d.copy()

print(d,id(d))
print(d1,id(d1))
```

```
{'name': 'Pankaj', 'branch': 'cs', 'degree': 'btech'} 1416805054080
{'name': 'Pankaj', 'branch': 'cs', 'degree': 'btech'} 1416805056384
```

K. setdefault():

d.setdefault(k,v)

```
If the key is already available then this function returns the corresponding value.
If the key is not available then the specified key-value will be added as new item
to the dictionary.
```

In [120…

```
d = {'name':'Pankaj','branch':'cs','degree':'btech'}
print(d)

print(d.setdefault('name','Pankaj'))
print(d)

print(d.setdefault('Home','Bihar'))
print(d)
```

```
{'name': 'Pankaj', 'branch': 'cs', 'degree': 'btech'}
Pankaj
{'name': 'Pankaj', 'branch': 'cs', 'degree': 'btech'}
Bihar
{'name': 'Pankaj', 'branch': 'cs', 'degree': 'btech', 'Home': 'Bihar'}
```

L. update():

d.update(x)

```
All items present in the dictionary x will be added to dictionary d
```

In [121…

```
x = {'name':'Pankaj','branch':'cs','degree':'btech'}
print(x)

d1.update(x)
print(d1)

d = {'Student':'S1'}
```

```
print(d)
d.update(x)
print(d)
```

```
{'name': 'Pankaj', 'branch': 'cs', 'degree': 'btech'}
{'name': 'Pankaj', 'branch': 'cs', 'degree': 'btech'}
{'Student': 'S1'}
{'Student': 'S1', 'name': 'Pankaj', 'branch': 'cs', 'degree': 'btech'}
```

dict.update(key1=value1, key2=value2)

dict.update({'key1':'value1', 'key2': 'value2'})

dict.update([(k1, v1), (k2, v2), (k3, v3)])

M. fromkeys()

In [122…

```
names = [ 'ram', 'lakshman', 'hanuman', 'seeta']

count = dict.fromkeys(names, 0)

print(count)
print(type(count))
```

```
{'ram': 0, 'lakshman': 0, 'hanuman': 0, 'seeta': 0}
<class 'dict'>
```

In [123…

```
d = {}

nd = d.fromkeys(['name', 'age', 'country'], None)
print(nd)
```

```
{'name': None, 'age': None, 'country': None}
```

7. Dictionary Comprehension

```
    Comprehension concept applicable for dictionaries also
```

In [124…

```
squares={x:x*x for x in range(1,6)}
print(squares)

doubles={x:2*x for x in range(1,6)}
print(doubles)
```

```
{1: 1, 2: 4, 3: 9, 4: 16, 5: 25}
{1: 2, 2: 4, 3: 6, 4: 8, 5: 10}
```

In [125…

```
print(*[ func for func in dir(dict) if func[0].islower() ], sep='\n')
```

```
clear
copy
fromkeys
get
items
keys
pop
popitem
setdefault
update
values
```

8. Dictionary From keyboard

In [126…

```
d=eval(input("Enter dictionary:"))
s=sum(d.values())
print("Sum= ",s)
```

```
Enter dictionary:{'a':10,'b':30,'c':50}
Sum=  90
```

9. Practice

In [127…

```
info= {
    'name': 'python',
    'father': 'Guido Van Rossum',
    'website': 'www.python.org',
    'scope': [ 'ml', 'ds', 'web', 'automation'],
    'versions': [ '1.X', '2.X', '3.X' ]
}
```

In [128…

```
info['name'] = 'Python Programming'
```

In [129…

```
from pprint import pprint
pprint(info)
```

```
{'father': 'Guido Van Rossum',
 'name': 'Python Programming',
 'scope': ['ml', 'ds', 'web', 'automation'],
 'versions': ['1.X', '2.X', '3.X'],
 'website': 'www.python.org'}
```

In [130…

```
d = { 'name': ['sachin'], 'name': 'rahul',
    'name': 'akhil'}
print(d)
```

```
{'name': 'akhil'}
```

In [131…

```
d = { 'name': [ 'ram', 'shyam', 'ghyanshya']}
print(d)
```

```
{'name': ['ram', 'shyam', 'ghyanshya']}
```

In [132…

```
d = { 'name': []}

d['name'].append('ram')
d['name'].append('shyam')
d['name'].append('ghyanshyam')

print(d)
```

```
{'name': ['ram', 'shyam', 'ghyanshyam']}
```

In [133…

```
d = { 'name': []}

d['name'] = 'ram'
d['name'] = 'shyam'
d['name'] = 'ghyanshyam'
```

```
print(d)
```

```
{'name': 'ghyanshyam'}
```

In [134…

```
info= {
    'name': 'python',
    'father': 'Guido Van Rossum',
    'website': 'www.python.org',
    'scope': [ 'ml', 'ds', 'web', 'automation'],
    'versions': [ '1.X', '2.X', '3.X' ]
}
```

In [135…

```
keys = list( info.keys() )
print(keys)
print()

values = list( info.values() )
pprint(values)
print()

items = list( info.items() )
pprint(items)
print()

d = dict(items)
print(d)
```

```
['name', 'father', 'website', 'scope', 'versions']

['python',
 'Guido Van Rossum',
 'www.python.org',
 ['ml', 'ds', 'web', 'automation'],
 ['1.X', '2.X', '3.X']]

[('name', 'python'),
 ('father', 'Guido Van Rossum'),
 ('website', 'www.python.org'),
 ('scope', ['ml', 'ds', 'web', 'automation']),
 ('versions', ['1.X', '2.X', '3.X'])]

{'name': 'python', 'father': 'Guido Van Rossum', 'website': 'www.python.org', 'scope': ['m
l', 'ds', 'web', 'automation'], 'versions': ['1.X', '2.X', '3.X']}
```

In [136…

```
key = 'name'
value = info.get(key, "no such information")
print(value)

key = 'abc'
value = info.get(key, "no such information")
print(value)

key = 'abc'
value = info.get(key)
print(value)
```

```
python
no such information
None
```

In [ ]:

In [137…

```
db = {
    '14earcs094': {
        'name': 'sachin yadav',
        'branch': 'cs',
        'section': 'B',
        'session': '2014-2018',
        'ph_no': [9782131159, 1234567890],
        'email': [ 'sachinyadav3496@gmail.com']
    },
    '14earcs70': {
        'name': 'vijay jangid',
        'branch': 'cs',
        'section': 'A',
        'session': '2014-18',
        #...
    }
}
```

In [138…

```
#data['2014-18']['cs']['B']['14earcs094']
#
pprint(db['14earcs094'])
```

```
{'branch': 'cs',
 'email': ['sachinyadav3496@gmail.com'],
 'name': 'sachin yadav',
 'ph_no': [9782131159, 1234567890],
 'section': 'B',
 'session': '2014-2018'}
```

In [ ]:

```

```

In [139…

```
data = {
    '2014-18':{
        'cs': {
            'A': {},
            'B': {
                '14earcs094':{
                    'name': 'sachin',
                    'ph_no': ['1234567890']
                }
            },
            'IT': {},
            'II': {}
        },
        'ec': {},
        'ee': {}
    }
}
```

In [140…

```
data['2014-18']['cs']['B']['14earcs094']
```

Out[140…

```
{'name': 'sachin', 'ph_no': ['1234567890']}
```

dict.update(key1=value1, key2=value2)

dict.update({'key1':'value1', 'key2': 'value2'})

dict.update([(k1, v1), (k2, v2), (k3, v3)])

In [141…

```
info= {
    'name': 'python',
    'father': 'Guido Van Rossum',
    'website': 'www.python.org',
    'scope': [ 'ml', 'ds', 'web', 'automation'],
    'versions': [ '1.X', '2.X', '3.X' ]
}

pprint(info)
```

```
{'father': 'Guido Van Rossum',
 'name': 'python',
 'scope': ['ml', 'ds', 'web', 'automation'],
 'versions': ['1.X', '2.X', '3.X'],
 'website': 'www.python.org'}
```

In [142…

```
info.update(name='python', modules=['flask', 'django',
                                    'sklearn', 'numpy'])
```

In [143…

```
pprint(info)
```

```
{'father': 'Guido Van Rossum',
 'modules': ['flask', 'django', 'sklearn', 'numpy'],
 'name': 'python',
 'scope': ['ml', 'ds', 'web', 'automation'],
 'versions': ['1.X', '2.X', '3.X'],
 'website': 'www.python.org'}
```

In [144…

```
info.update([ ('name', 'PYTHON'),
             ('package_url', 'www.pypi.org'),
             ('company', 'Python Software Foundation') ])
```

In [145…

```
pprint(info)
```

```
{'company': 'Python Software Foundation',
 'father': 'Guido Van Rossum',
 'modules': ['flask', 'django', 'sklearn', 'numpy'],
 'name': 'PYTHON',
 'package_url': 'www.pypi.org',
 'scope': ['ml', 'ds', 'web', 'automation'],
 'versions': ['1.X', '2.X', '3.X'],
 'website': 'www.python.org'}
```

In [146…

```
info = {
    'name': 'iphone',
    'price': 130000,
    'model': '13 pro max',
    'size': '128 gb',
}
```

In [147…

```
value = info.setdefault('color', 'blue')
print(value)
```

```
blue
```

In [148…

```
from pprint import pprint
```

In [149…

```
pprint(info)
```

```
{'color': 'blue',
 'model': '13 pro max',
 'name': 'iphone',
 'price': 130000,
 'size': '128 gb'}
```

In [150…

```
value = info.setdefault('color', 'red')
print(value)
```

```
blue
```

In [151…

```
pprint(info)
```

```
{'color': 'blue',
 'model': '13 pro max',
 'name': 'iphone',
 'price': 130000,
 'size': '128 gb'}
```

10. Problem

Q. Write a program to find number of occurrences of each vowel present in the given string?

In [152…

```
word=input("Enter any word: ")
vowels={'a','e','i','o','u'}
d={}
for x in word:
    if x in vowels:
        d[x]=d.get(x,0)+1

for k,v in sorted(d.items()):
    print(k,"occurred ",v," times")
```

```
Enter any word: Pankaj Kumar Yadav
a occurred  5  times
u occurred  1  times
```

In [ ]: