Dictionary

- dic is represented with curly brackets{}
- string use quotes
- list uses []
- tuple use()
- dictionary is always a pair key and value

In [1]:

```
# syntax
# dict= {<key>:<value>}
```

In [2]:

```
dict1={'manish':21,'subham':25,'akhil':20}  #keys are manish,subham,akhil
                                            # values are 21 25 20
dict1
```

Out[2]:

```
{'manish': 21, 'subham': 25, 'akhil': 20}
```

In [3]:

```
type(dict1)
```

Out[3]:

```
dict
```

In [7]:

```
# d1={'manish':21,'subham':25,'akhil':20}     #works
# d2={'manish':'21','subham':'25','akhil':'20'} #works
# d3={21:'manish',20:'subham',23:'akhil'}      #not works
# d4={20:25}
# d5={'fruites':['apple','cherry']}
# d6={['apple','cherry']:'fruites'}
# d7={'ajay':20,'ajay':30}
# d8={20:'ajay',30:'ajay'}
```

- **keys are important**
- keys can have the latest value
- no duplicates key
- key should be one
- values can be duplicate which means different keys can have same value
- list cannot be a key
- tuple can be a key

In [8]:

```
d1={'manish':21,'subham':25,'akhil':20}
min(d1)
```

Out[8]:

```
'akhil'
```

In [9]:

```
max(d1)
```

Out[9]:

```
'subham'
```

```
In [10]: len(d1)

Out[10]: 3
```

```
In [11]: d1={'manish':21,'subham':25,'akhil':20}
         sorted(d1)

Out[11]: ['akhil', 'manish', 'subham']
```

```
In [12]:  for i in reversed(d1):
              print(i)

         akhil
         subham
         manish
```

```
 In [ ]: d1={'manish':21,'subham':25,'akhil':20}
         'manish' in d1   # only string elements gives true if its presents
         'subham'   in d1  #true
         'raju' not in d1  #true
         'ram' in d1      #False
```

for loop

```
In [13]: for i in d1:
             print(i)

         manish
         subham
         akhil
```

mutable

- added at last of the dictionary

```
In [15]: d1={'manish':21,'subham':25,'akhil':20}
         a=d1[0]='ramu'
         d1

Out[15]: {'manish': 21, 'subham': 25, 'akhil': 20, 0: 'ramu'}
```

```
In [16]: d1={'manish':21,'subham':25,'akhil':20}
         d1.popitem()
         d1

Out[16]: {'manish': 21, 'subham': 25}
```

In [ ]:

```
# 'clear',
#  'copy',
#  'fromkeys',
#  'get',
#  'items',
#  'keys',
#  'pop',
#  'popitem',
#  'setdefault',
#  'update',
#  'values']
```

In [17]:

```
dir(dict)
```

Out[17]:
```
['__class__',
 '__class_getitem__',
 '__contains__',
 '__delattr__',
 '__delitem__',
 '__dir__',
 '__doc__',
 '__eq__',
 '__format__',
 '__ge__',
 '__getattribute__',
 '__getitem__',
 '__getstate__',
 '__gt__',
 '__hash__',
 '__init__',
 '__init_subclass__',
 '__ior__',
 '__iter__',
 '__le__',
 '__len__',
 '__lt__',
 '__ne__',
 '__new__',
 '__or__',
 '__reduce__',
 '__reduce_ex__',
 '__repr__',
 '__reversed__',
 '__ror__',
 '__setattr__',
 '__setitem__',
 '__sizeof__',
 '__str__',
 '__subclasshook__',
 'clear',
 'copy',
 'fromkeys',
 'get',
 'items',
 'keys',
 'pop',
 'popitem',
 'setdefault',
 'update',
 'values']
```

In [19]:
```python
l=['apple' ,'cat', 'dog', 'ball']
#sort it without using sorting

l=[10,15,2,25,89]
#maximum value

l=[1,2,3,4,]
l2=[10,20,30,40]
#output:[11,22,33,44]

l1=[1,2,3,4,5]
l2=[10,20,30,40]
#output[11,22,33,44,5]
```

```
s='hello how are you im good at python in naresh it'
# find the maximum length word
# minimum length word
# repeated words
```

In [20]:

```
l=[1,2,3,4,]
l2=[10,20,30,40]
out=[]
#output:[11,22,33,44]

for i in range(len(l)):

       out.append(l[i]+l2[i])
out
```

Out[20]:

```
[11, 22, 33, 44]
```

In [21]:

```
l1=[1,2,3,4,5]
l2=[10,20,30,40,0 ] #0
#output[11,22,33,44,5]
n=len(l1)
out=[]

for i in range(n):

        out.append(l1[i]+l2[i])

out
```

Out[21]:

```
[11, 22, 33, 44, 5]
```

In [22]:

```
l=[10,15,2,25,89]
max(l),min(l)
```

Out[22]:

```
(89, 2)
```

In [23]:

```
l=['apple' ,'cat', 'dog', 'ball']
#sort it without using sorting
for i in l:
    a=sorted(l)
a
```

Out[23]:

```
['apple', 'ball', 'cat', 'dog']
```

indexing

In [24]:

```
d1={'manish':21,'subham':25,'akhil':20}
d1['manish']
```

Out[24]:

```
21
```

- Dictionary is a **key value pair**
- there is not concept of index as number

- if we want to retrive any value,we need provide key as index
- we access values using only corresponding **keys**

In [25]:
```
d1={'manish':21,'subham':25,'akhil':20}
print(d1['manish'])
print(d1['subham'])
print(d1['akhil'])
```

```
21
25
20
```

In [26]:
```
for i in d1:
                     # to print keys
    print(i)
```

```
manish
subham
akhil
```

In [27]:
```
for i in d1:            # to print value
    print(d1[i])
```

```
21
25
20
```

In [28]:
```
for i in d1:
    print(f"{i} age is {d1[i]}")      # keys and values are together
```

```
manish age is 21
subham age is 25
akhil age is 20
```

In [29]:
```
# How to access the elements
d1={'fruties':['apple']}
d1['fruties'][0]
```

Out[29]:
```
'apple'
```

In [30]:
```
d1={'Fruites':{"Apple":['Kashmir']}}
d1['Fruites']['Apple'][0]
```

Out[30]:
```
'Kashmir'
```

In [31]:
```
d1={'Fruites':{"Apple":[
                     {'Kashmir':'india'},
                     {'Hyd':'TS'}
                     ]
              }           }
d1['Fruites']
```

Out[31]:
```
{'Apple': [{'Kashmir': 'india'}, {'Hyd': 'TS'}]}
```

In [32]:
```
d1={'Fruites':{"Apple":[
                     {'Kashmir':'india'},
                     {'Hyd':'TS'}
                     ]
               }
```

```
   }
d1['Fruites']['Apple'][0]['Kashmir']
```

Out[32]:

```
'india'
```

In [33]:

```
d1={'Fruites':{"Apple":[
                     {'Kashmir':'india'},
                     {'Hyd':'TS'}
                     ]
               }
   }
d1['Fruites']['Apple'][1]['Hyd']
```

Out[33]:

```
'TS'
```

In [34]:

```
d1={'Fruites':{'Mango':{'Benganpalli':{'Nagpur market':{'MH':{'Shivaji':'the bos
d1['Fruites']['Mango']['Benganpalli']['Nagpur market']['MH']['Shivaji']
```

Out[34]:

```
'the boss'
```

How to create empty dictionary

- it is very imp to know
- how to make empty string
- how to make empty dictionary

In [35]:

```
str1='python'
str2=''
for i in str1:
    str2=str2+i
print(str2)
```

```
python
```

In [36]:

```
l=[1,2,3]
l1=[]
for i in l:
    l1.append(i)
l1
```

Out[36]:

```
[1, 2, 3]
```

In [37]:

```
d={}
d['ramesh']=50
d['suresh']=100
d['sathish']=150
d['manish']=200
d
```

Out[37]:

```
{'ramesh': 50, 'suresh': 100, 'sathish': 150, 'manish': 200}
```

coversion of string to list

In [38]:

```
str1='hai hello how are you'
l=str1.split()
```

```
l
```

Out[38]:

```
['hai', 'hello', 'how', 'are', 'you']
```

join:list to string

In [39]:

```
l=['hai','how','are','you']
new_str='  '.join(l)
new_str
```

Out[39]:

```
'hai  how  are  you'
```

we have two lists to convert to dictionary

In [40]:

```
name=['ramesh','suresh','raju']
age=[25,20,21]
d={}
for i,j in zip(name,age):
    d[i]=j
d
```

Out[40]:

```
{'ramesh': 25, 'suresh': 20, 'raju': 21}
```

In [41]:

```
str='how are you' #string to list to dict
l=str.split()
items=[1,2,3]
d={}
for i,j in zip(l,items):
    d[i]=j
d
```

Out[41]:

```
{'how': 1, 'are': 2, 'you': 3}
```

methods

In [ ]:

```
# dir('')  string
# dir([])  list
# dir({})  dictionary
# dir(( )) tuple
```

In [42]:

```
dir({})
```

```
Out[42]: ['__class__',
          '__class_getitem__',
          '__contains__',
          '__delattr__',
          '__delitem__',
          '__dir__',
          '__doc__',
          '__eq__',
          '__format__',
          '__ge__',
          '__getattribute__',
          '__getitem__',
          '__getstate__',
          '__gt__',
          '__hash__',
          '__init__',
          '__init_subclass__',
          '__ior__',
          '__iter__',
          '__le__',
          '__len__',
          '__lt__',
          '__ne__',
          '__new__',
          '__or__',
          '__reduce__',
          '__reduce_ex__',
          '__repr__',
          '__reversed__',
          '__ror__',
          '__setattr__',
          '__setitem__',
          '__sizeof__',
          '__str__',
          '__subclasshook__',
          'clear',
          'copy',
          'fromkeys',
          'get',
          'items',
          'keys',
          'pop',
          'popitem',
          'setdefault',
          'update',
          'values']
```

item-keys- values

- items means key and values come together
- keys means only dictionary keys
- values means only dictionary values

```
In [43]: d={}
         d['ramesh']=50
```

```
          d['suresh']=100
          d['sathish']=150
          d['manish']=200
          l=d.items()
          l

Out[43]:  dict_items([('ramesh', 50), ('suresh', 100), ('sathish', 150), ('manish', 20
          0)])

In [44]:  type(l)

Out[44]:  dict_items

In [45]:  list(l) #conversion of dict to list

Out[45]:  [('ramesh', 50), ('suresh', 100), ('sathish', 150), ('manish', 200)]

In [46]:  l1=d.keys()
          l1

Out[46]:  dict_keys(['ramesh', 'suresh', 'sathish', 'manish'])

In [47]:  type(l1)

Out[47]:  dict_keys

In [48]:  list(l1)  # conversion of keys to list

Out[48]:  ['ramesh', 'suresh', 'sathish', 'manish']

In [49]:  l2=d.values()
          l2

Out[49]:  dict_values([50, 100, 150, 200])

In [50]:  type(l2)

Out[50]:  dict_values

In [51]:  list(l2) # convert to list

Out[51]:  [50, 100, 150, 200]

In [ ]:
```