

```
In [1]: list1=['Ram','Raheem','Robert']  
list2=[25,30,35]
```

```
In [ ]: 'Ram has age 25'  
        'Raheem has age 30'  
        'Robert has age 35'
```

```
In [2]: for i,j in zip(list1,list2):  
        print(i,j)
```

```
Ram 25  
Raheem 30  
Robert 35
```

```
In [ ]: #{key:value}
```

```
In [3]: d1={'Ram':25,  
           'Raheem':30,  
           'Robert':35}  
  
d1  
  
#keys= 'Ram','Raheem','Robert'  
#values=25,30,35
```

```
Out[3]: {'Ram': 25, 'Raheem': 30, 'Robert': 35}
```

```
In [4]: d2={25:'Ram',30:'Raheem',35:'Robert'}  
d2  
  
#keys: 25,30,35  
#values: 'Ram','Raheem','Robert'
```

```
Out[4]: {25: 'Ram', 30: 'Raheem', 35: 'Robert'}
```

```
In [5]: d3={'even':[2,4,6],  
           'odd':[3,5,7]}  
  
d3
```

```
Out[5]: {'even': [2, 4, 6], 'odd': [3, 5, 7]}
```

```
In [ ]: d4={ [2,4,6]: 'even',  
             [3,5,7]: 'odd' }      # fail
```

```
In [8]: d4={(2,4,6): 'even',  
           (3,5,7): 'odd'}  
  
d4
```

```
Out[8]: {(2, 4, 6): 'even', (3, 5, 7): 'odd'}
```

```
In [9]: # {'key':<{}>}
d5={'item_list':{'fruiits':'Apple'}}
d5
```

```
Out[9]: {'item_list': {'fruiits': 'Apple'}}
```

```
In [10]: d6={{'fruiits':'Apple'}:'item_list'}
d6
```

```
-----
-
TypeError                                Traceback (most recent call las
t)
Cell In[10], line 1
----> 1 d6={{'fruiits':'Apple'}:'item_list'}
      2 d6

TypeError: unhashable type: 'dict'
```

```
In [11]: a=[1,2]
b=(1,2)
```

```
In [18]: a,b=[1,2]
c,d=(1,2)
d
```

```
Out[18]: 2
```

```
In [19]: a=1,2
a
```

```
Out[19]: (1, 2)
```

```
In [20]: d1={'A':1,'B':2,'A':1}
d1
```

```
Out[20]: {'A': 1, 'B': 2}
```

```
In [22]: d1={'A':1,'B':2,'A':3}
d1
```

```
Out[22]: {'A': 3, 'B': 2}
```

- Dictionary is a key:value pair
- at values position you can take any data type
- at keys position list and dictionary will fail
- Duplicates are not allowed
- If you will update a key value, latest value it will take

type

```
In [23]: type(d1)
```

```
Out[23]: dict
```

- str
- list
- dict
- int
- float
- bool
- complex
- tuple
- set

```
In [24]: d1={'Ram':25,
            'Raheem':30,
            'Robert':35}

max(d1)

# this maximum we are getting based on key or value
```

```
Out[24]: 'Robert'
```

```
In [25]: d1={'Ram':25,
            'Raheem':30,
            'Robert':3}

max(d1)
```

```
Out[25]: 'Robert'
```

- Maximum and minimum value based on key only

```
In [26]: min(d1)

'Ram',    ord('m')
'Raheem', ord('h')
'Robert'
```

```
Out[26]: 'Raheem'
```

```
In [27]: len(d1)
```

```
Out[27]: 3
```

```
In [28]: d1
```

```
Out[28]: {'Ram': 25, 'Raheem': 30, 'Robert': 3}
```

```
In [29]: sum(d1)
```

```
-----  
-  
TypeError                                Traceback (most recent call las  
t)  
Cell In[29], line 1  
----> 1 sum(d1)  
  
TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

```
In [32]: d2={100:'2',300:'4'}  
sum(d2)
```

```
Out[32]: 400
```

- if keys has numeric then we can do sum

```
In [35]: 'a'*'b'
```

```
-----  
-  
TypeError                                Traceback (most recent call las  
t)  
Cell In[35], line 1  
----> 1 'a'*'b'  
  
TypeError: can't multiply sequence by non-int of type 'str'
```

- type
- len
- max
- min
- sum

in

```
In [40]: d1={'Ram':25,  
            'Raheem':30,  
            'Robert':35}  
  
# 'Ram':25 in d1 fail  
  
'Ram' in d1 # works  
  
#25 in d1 # Fails
```

```
Out[40]: True
```

```
In [39]: for i in d1:  
        print(i)
```

Ram
Raheem
Robert

```
In [38]: str1='apple'  
        'a' in str1  
  
        l1=[1,2,3]  
        1 in l1  
  
        for i in l1:  
            print(i)
```

1
2
3

index

```
In [42]: l1=[10,20,30,40]  
        l1[0]  
  
        str1='apple'  
        str1[0]
```

Out[42]: 'a'

```
In [44]: d1={'Ram':25,  
            'Raheem':30,  
            'Robert':35}  
  
        d1['Ram']  
  
        # can we get values using for loop
```

Out[44]: 25

```
In [49]: for key in d1:  
        print("The age of {} is {}".format(key,d1[key]))
```

The age of Ram is 25
The age of Raheem is 30
The age of Robert is 35

```
In [51]: for i in range(len(l1)):  
        print(i,l1[i])
```

0 10
1 20
2 30
3 40

```
In [52]: for key in range(len(d1)):
          print(key)
          #print("The age of {} is {}".format(key,d1[key])) # possibile / not pos

0
1
2
```

Creating a empty dictionary and update

```
In [54]: s=''

for i in 'apple':
    s=s+i

print(s)
```

apple

```
In [55]: l=[i for i in range(10)]

l=[]
for i in range(10):
    l.append(i)

l
```

Out[55]: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

```
In [56]: d1={}
          d1['fruite']='Apple'
          d1
```

Out[56]: {'fruite': 'Apple'}

```
In [57]: d1={}
          d1["First name"]="Nikita"
          d1["sir name"]="Parolekar"
          d1["Nativeplace"]="Dhule"
          d1["State"]="Maharashtra"
          d1
```

Out[57]: {'First name': 'Nikita',
 'sir name': 'Parolekar',
 'Nativeplace': 'Dhule',
 'State': 'Maharashtra'}

```
In [ ]: # WAP create a dictionary based on two lists
# names=['Ram','Raheem','Robert']
# age=[25,30,35]
# d1={'Ram':25,'Raheem':30,'Robert':35}

names=['Ram','Raheem','Robert']
age=[25,30,35]
```

```
In [58]: names=['Ram','Raheem','Robert']
age=[35,25,30]
dict1={}
for i,j in zip(names,age):
    dict1[i]=j
dict1

#####
dict1={name:age for name,age in zip(names,age)}
```

```
Out[58]: {'Ram': 35, 'Raheem': 25, 'Robert': 30}
```

```
In [73]: dict1={}
for i in range(len(names)):
    dict1[names[i]]=age[i]

#####
dict1={names[i]:age[i] for i in range(len(names))}
```

```
Ram 25
Raheem 30
Robert 35
```

```
In [68]: list1=[]
for i in range(len(names)):
    list1.append(i)

list1=[i for i in range(len(names))]
list1
```

```
Out[68]: [0, 1, 2]
```

```
In [ ]: #WAP take 5 random numbers
        # and create a dictionary with even and odd

        # Output: {'even':[20,22], 'odd':[19,21,23]}

        # step-1: take empty dictionary
        # step-2: take two even and odd list
        # step-3: import random
        # step-4: for i in range(5):
        # step-5:     num=random.randint(a,b)
        # step-6:     if <even>:
        # step-7:         append the values in even list
        # step-8:     else:
        # step-9:         append the values in odd list
        # step-10: create dictionary
```

```
In [1]: import os
```

```
In [2]: os.getcwd()
```

```
Out[2]: 'C:\\Users\\omkar\\Documents'
```

dictionary methods


```
In [3]: dir({})
```

```
# str=''
# list=[]
# dict={}
```

```
Out[3]: ['__class__',
          '__class_getitem__',
          '__contains__',
          '__delattr__',
          '__delitem__',
          '__dir__',
          '__doc__',
          '__eq__',
          '__format__',
          '__ge__',
          '__getattr__',
          '__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']
```

items – keys – values

```
In [4]: d1={'Ram':25,  
          'Raheem':30,  
          'Robert':35}
```

```
d1
```

```
Out[4]: {'Ram': 25, 'Raheem': 30, 'Robert': 35}
```

```
In [8]: #items  
items=d1.items()  
items
```

```
Out[8]: dict_items([('Ram', 25), ('Raheem', 30), ('Robert', 35)])
```

```
In [9]: type(items)
```

```
Out[9]: dict_items
```

```
In [10]: #keys  
keys=d1.keys()  
keys
```

```
Out[10]: dict_keys(['Ram', 'Raheem', 'Robert'])
```

```
In [11]: type(keys)
```

```
Out[11]: dict_keys
```

```
In [12]: # values  
values=d1.values()  
values
```

```
Out[12]: dict_values([25, 30, 35])
```

```
In [13]: type(values)
```

```
Out[13]: dict_values
```

```
In [15]: l1=[25,30,35]  
l1.append(400)  
l1
```

```
Out[15]: [25, 30, 35, 400]
```

```
In [20]: values      # I want to convert into a list  
values_list=list(values) # then you can apply list methods  
values_list
```

```
Out[20]: [25, 30, 35]
```

```
In [21]: keys_list=list(keys)  
keys_list
```

```
Out[21]: ['Ram', 'Raheem', 'Robert']
```

```
In [ ]: # I will give the dictionary
        # can you extract keys and values in a list

        # I will give two list keys and values
        # can you create a dictionary
```

```
In [22]: d1={'Ram':25,
            'Raheem':30,
            'Robert':35}

        keys=list(d1.keys())
        values=list(d1.values())
```

```
In [24]: keys,values
```

```
Out[24]: (['Ram', 'Raheem', 'Robert'], [25, 30, 35])
```

```
In [25]: {i:j for i,j in zip(keys,values)}
```

```
Out[25]: {'Ram': 25, 'Raheem': 30, 'Robert': 35}
```

```
In [27]: d1={}
        for i in range(len(keys)):
            d1[keys[i]]=values[i]    # d1[keys[0]]=values[0]    d1['Ram']=25

        d1
```

```
Out[27]: {'Ram': 25, 'Raheem': 30, 'Robert': 35}
```

```
In [28]: dict(zip(keys,values))
```

```
Out[28]: {'Ram': 25, 'Raheem': 30, 'Robert': 35}
```

```
In [ ]: s1='virat.kohli@blr.com,Rohit.sharma@mumbai.com,kl.rahul@lucknow.com'

        # {'first_name':['virat','Rohit','KL'],
        #   'second_name':['kohli','sharma','rahul'],
        #   'company':['blr','mumbai','lucknow']}
```

```
In [32]: s1.split(',')
```

```
Out[32]: ['virat.kohli@blr.com', 'rohit.sharma@mumbai.com', 'kl.rahul@lucknow.com']
```

```
In [44]: str1='virat.kohli@blr.com'
d1={}
f_name=[str1[:str1.find('.')] for str1 in s1.split(',')]
s_name=[str1[str1.find('.')+1:str1.find('@')] for str1 in s1.split(',')]
c_name=[str1[str1.find('@')+1:str1.find('.',str1.find('.')+1)] for str1 in
f_name,s_name,c_name

d1['first_name']=f_name
d1['second_name']=s_name
d1['company_name']=c_name

d1
```

```
Out[44]: {'first_name': ['virat', 'rohit', 'kl'],
'second_name': ['kohli', 'sharma', 'rahul'],
'company_name': ['blr', 'mumbai', 'lucknow']}
```

```
In [30]: s1='virat.kohli@blr.com,rohit.sharma@mumbai.com,kl.rahul@lucknow.com'
l1=s1.split(',')
first_name=[]
second_name=[]
company=[]
d2={}
for i in l1:
    #print(i)
    first_name.append(i[0:i.find('.')])
    second_name.append(i[i.find('.')+1:i.find('@')])
    company.append(i[i.find('@')+1:i.find('.',i.find('.')+1)])

d2['first_name']=first_name
d2['second_name']=second_name
d2['company']=company
d2
```

```
Out[30]: {'first_name': ['virat', 'rohit', 'kl'],
'second_name': ['kohli', 'sharma', 'rahul'],
'company': ['blr', 'mumbai', 'lucknow']}
```

words-frequency

```
In [ ]: str1='can can you canner can you able to can canner'

#{'can':4,'you':2,'canner':2,'able':1,'to':1}

# d={}
# step-1: split the str1 =====> you will get a list
# step-2: iterate through loop ====> each word will print
# step-3: list1.count(<word>): number
# step-4: make the dictionary
```

```
In [46]: str1='can can you canner can you able to can canner.'
        l1=str1.split(' ') # step-1
        d1={}

        for i in l1:      # step-2
            d1[i]=l1.count(i) # step-3

        d1
```

Out[46]: {'can': 4, 'you': 2, 'canner': 2, 'able': 1, 'to': 1}

```
In [47]: str1='can can you canner can you able to can canner.'
        str1.split()
```

Out[47]: ['can', 'can', 'you', 'canner', 'can', 'you', 'able', 'to', 'can', 'canner']

```
In [55]: keys=list(d1.keys())
```

```
In [56]: values=list(d1.values())
```

```
In [58]: keys,values
```

Out[58]: (['can', 'you', 'canner', 'able', 'to'], [4, 2, 2, 1, 1])

```
In [60]: i=max(values)
```

```
In [61]: values.index(i)
```

Out[61]: 0

```
In [63]: keys[values.index(max(values))]
```

Out[63]: 'can'

```
In [ ]: d1={'a':20,'b':30,'c':40}
        d2={'a':50,'b':100,'c':200}

        # o/p={'a':70,'b':130,'c':240}
```

```
In [ ]: d1={'a':20,'b':30,'c':40,'d':500}
        d2={'a':50,'b':100,'c':200}

        # o/p={'a':70,'b':130,'c':240,'d':500}
```

```
In [64]: d1 = {'a':20,'b':30,'c':40,'d':500}
         d2 = {'a':50,'b':100,'c':200}
         for i in d2:
             d1[i]+=d2[i]
         d1
```

```
Out[64]: {'a': 70, 'b': 130, 'c': 240, 'd': 500}
```

```
In [65]: d1 = {'a':20,'b':30,'c':40}
         d2 = {'a':50,'b':100,'c':200}
         for i in d1:
             d1[i]+=d2[i]
         d1
```

```
Out[65]: {'a': 70, 'b': 130, 'c': 240}
```

```
In [67]: d1={'a':20,'b':30,'c':40}
         d2={'a':50,'b':100,'c':200}

         d3={}
         if len(d1)==len(d2):
             for i in d1:
                 d3[i]=d1[i]+d2[i]
                 # d3['a']=d1['a']+d2['a']

         d3

         # which ever is max length iterate through that dictionary
```

```
Out[67]: {'a': 70, 'b': 130, 'c': 240}
```

```
In [68]: for i in range(min(len(d1),len(d2))):
         d1[list(d1.keys())[i]]+=d2[list(d2.keys())[i]]
         d1
```

```
Out[68]: {'a': 70, 'b': 130, 'c': 240}
```

```
In [69]: d1 = {'a':20,'b':30,'c':40,'d':500}
         d2 = {'a':50,'b':100,'c':200}
         for i in range(min(len(d1),len(d2))):
             d1[list(d1.keys())[i]]+=d2[list(d2.keys())[i]]
         d1
```

```
Out[69]: {'a': 70, 'b': 130, 'c': 240, 'd': 500}
```

```
In [ ]: # strings /list/dictionary
         # Python developer
```

```
In [ ]: # this week finish python
         # statistics will start (10days)
         # EDA with python
```

```
In [ ]:
```

In []:

In []:

In []:

In []:

In []:

In []: