In [ ]:

```
strings='Rashidh'
list=['Rashidh']
tuple= ('Rashidh')
set= {'Rashidh'}
```

In [2]:

```
l=[1,2,3]
l[0]=100
l
```

Out[2]:

```
[100, 2, 3]
```

In [4]:

```
t=(1,2,3)
t[0]=100
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[4], line 2
      1 t=(1,2,3)
----> 2 t[0]=100

TypeError: 'tuple' object does not support item assignment
```

In [6]:

```
Names= ['Ramesh','Suresh','Mahesh']
Age=[21,22,23]
# Ramesh age is 21
# Suresh age is 22
for i in range(len(Names)):
    print(f"{Names[i]} has age {Age[i]}")
```

```
Ramesh has age 21
Suresh has age 22
Mahesh has age 23
```

zip

In [7]:

```
zip(Names,Age)
```

Out[7]:

```
<zip at 0x2970a8d5780>
```

In [8]:

```
list(zip(Names,Age))
```

Out[8]:

```
[('Ramesh', 21), ('Suresh', 22), ('Mahesh', 23)]
```

In [9]:

```
for i in zip(Names,Age):
    print(i)
```

```
('Ramesh', 21)
('Suresh', 22)
('Mahesh', 23)
```

In [10]:

```
a=10,20
a
```

Out[10]:

```
(10, 20)
```

In [ ]:

```
a,b=10,20
```

In [11]:
```
for i,j in zip(Names,Age):
    print(f"the {i} age is {j}")
```

```
the Ramesh age is 21
the Suresh age is 22
the Mahesh age is 23
```

In [ ]:
```
# Ramesh ==== 21
# Suresh === 22
# Mahesh === 23
```

Dictionary

- Dictionary represnts with curly braces
- Dictionay has key: value pair
- {key:value}

In [12]:
```
d1={'Ramesh':21,'Suresh':22,'Mahesh':23}
d1
```

Out[12]:
```
{'Ramesh': 21, 'Suresh': 22, 'Mahesh': 23}
```

In [13]:
```
type(d1)
```

Out[13]:
```
dict
```

In [14]:
```
d2={'Ramesh':'21','Suresh':'22','Mahesh':'23'}
d2
```

Out[14]:
```
{'Ramesh': '21', 'Suresh': '22', 'Mahesh': '23'}
```

In [15]:
```
d3={21:'Ramesh'}
d3
```

Out[15]:
```
{21: 'Ramesh'}
```

In [16]:
```
d4={'Ramesh':21,'Ramesh':25}
d4 # latest value
```

Out[16]:
```
{'Ramesh': 25}
```

In [17]:
```
d5={'Ramesh':21,'Suresh':21}
d5 # value can be duplicate
# keys can not be duplicate
# keys are important
```

Out[17]:
```
{'Ramesh': 21, 'Suresh': 21}
```

In [18]:
```
d6={21:'Ramesh',21:'Suresh'}
d6
# keys are impoartnt
# duplicate keys are not allowed
```

Out[18]:
```
{21: 'Suresh'}
```

In [19]:
```
d7={'int':10,'float':10.5,'str':'str','boolea':True}
d7
```

Out[19]:
```
{'int': 10, 'float': 10.5, 'str': 'str', 'boolea': True}
```

In [20]:
```
d8={'list':[1,2,3]}
d8
```

Out[20]:
```
{'list': [1, 2, 3]}
```

In [21]:
```
d9={[1,2,3]:'list'}
d9
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[21], line 1
----> 1 d9={[1,2,3]:'list'}
      2 d9

TypeError: unhashable type: 'list'
```

In [22]:
```
d10={'tuple':(1,2,3)}
d10
```

Out[22]:
```
{'tuple': (1, 2, 3)}
```

In [23]:
```
d11={(1,2,3):'tuple'}
d11
```

Out[23]:
```
{(1, 2, 3): 'tuple'}
```

In [ ]:
```
# keys are impoartnt
list mutable
tuple immutable
Ramesh ====
```

In [ ]:
```
d1={'Ramesh':21,'Suresh':22,'Mahesh':23} # w
d2={'Ramesh':'21','Suresh':'22','Mahesh':'23'} # w
d3={21:'Ramesh'} # w
d4={'Ramesh':21,'Ramesh':25} # w  keys
d5={'Ramesh':21,'Suresh':21} # w
d6={21:'Ramesh',21:'Suresh'} # w
d7={'int':10,'float':10.5,'str':'str','boolea':True} # w
d8={'list':[1,2,3]} # w
d9={[1,2,3]:'list'} # f
d10={'tuple':(1,2,3)} # w
d11={(1,2,3):'tuple'} # w
d12={[1,2,3]:[1,2,3]}
d13={(1,2,3):(1,2,3)}
d14={'Name':{'Ramesh':25}}
d15={{'Ramesh':25}:'Name'} # w
```

In [2]:
```
#d9={[1,2,3]:'list'}
# keys are important
# keys should be unique
# list is applied at keys position
# list is mutable ==== fail
```

```
#d11={(1,2,3):'tuple'} # w

d15={{'Ramesh':25}:'Name'}
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[2], line 9
      1 #d9={[1,2,3]:'list'}
      2 # keys are important
      3 # keys should be unique
   (...)
      6 
      7 #d11={(1,2,3):'tuple'} # w
----> 9 d15={{'Ramesh':25}:'Name'}

TypeError: unhashable type: 'dict'
```

- len
- min
- max
- sorted
- revresed

In [3]:
```
d1={'Ramesh':21,'Suresh':22,'Mahesh':23}
d1
```

Out[3]:
```
{'Ramesh': 21, 'Suresh': 22, 'Mahesh': 23}
```

In [4]:
```
len(d1)
```

Out[4]:
```
3
```

In [5]:
```
max(d1) # keys are important
```

Out[5]:
```
'Suresh'
```

In [6]:
```
min(d1) # keys are important
```

Out[6]:
```
'Mahesh'
```

In [7]:
```
sorted(d1)
```

Out[7]:
```
['Mahesh', 'Ramesh', 'Suresh']
```

In [9]:
```
list(reversed(d1))
```

Out[9]:
```
['Mahesh', 'Suresh', 'Ramesh']
```

In [ ]:
```
for i in [1,2,3]:
    print(i,end=' ')

for i in '123':
    print(i,end=' ')

for i in {'n1':1,'n2':2,'n3':3}:
    print(i)
```

In [10]:

```
d1={'Ramesh':21,'Suresh':22,'Mahesh':23}
d1
```

Out[10]:

```
{'Ramesh': 21, 'Suresh': 22, 'Mahesh': 23}
```

In [11]:

```
d1[0]
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
Cell In[11], line 1
----> 1 d1[0]

KeyError: 0
```

In [12]:

```
d1['Ramesh']
```

Out[12]:

```
21
```

In [13]:

```
d1['Suresh']
```

Out[13]:

```
22
```

if want to access values, first we need to use key as index

In [ ]:

```
d1['Ramesh']
d1['Suresh']
d1['Mahesh']

d1[i]
```

In [15]:

```
for i in d1:
    print(i,d1[i])
```

```
Ramesh 21
Suresh 22
Mahesh 23
```

In [16]:

```
d1={'Ramesh':21,'Suresh':22,'Mahesh':23}
for i in d1:
    print(f"The {i} age is {d1[i]}")
```

```
The Ramesh age is 21
The Suresh age is 22
The Mahesh age is 23
```

In [ ]:

```
Names= ['Ramesh','Suresh','Mahesh']
Age=[21,22,23]
for i in range(len(Names)):
    print(f"{Names[i]} has age {Age[i]}")
```

How to create empty dictionary

In [17]:

```
s=''
s=s+'p'
s
```

Out[17]:

```
'p'
```

In [18]:

```
l=[]
l.append(10)
l
```

Out[18]:

```
[10]
```

In [19]:

```
d={}
d['Fruite']='Apple'
d
```

Out[19]:

```
{'Fruite': 'Apple'}
```

create a dictionary using list

In [20]:

```
Names= ['Ramesh','Suresh','Mahesh']
Age=[21,22,23]
d={}
d['Ramesh']=21
d['Suresh']=22
d['Mahesh']=23

d[key]=value
```

Out[20]:

```
{'Ramesh': 21, 'Suresh': 22, 'Mahesh': 23}
```

In [21]:

```
for i,j in zip(Names,Age):
    print(i,j)
```

```
Ramesh 21
Suresh 22
Mahesh 23
```

In [22]:

```
d={}
for i,j in zip(Names,Age):
    d[i]=j
d
```

Out[22]:

```
{'Ramesh': 21, 'Suresh': 22, 'Mahesh': 23}
```

How to make a list from dictionary

- dictionary has key and values
- so we can make two lists
- keys and values

In [29]:

```
keys=[]
values=[]
for i in d:
    keys.append(i)
    values.append(d[i])
keys,values
```

Out[29]:

```
(['Ramesh', 'Suresh', 'Mahesh'], [21, 22, 23])
```

In [ ]:

```
Names= ['Ramesh','Suresh','Mahesh']
Age=[21,22,23]
d={}
for i,j in zip(Names,Age):
    d[i]=j

#######################################
keys=[]
values=[]
for i in d:
    keys.append(i)
    values.append(d[i])
keys,values
```

mutable-immutable

In [31]:

```
d1={'Ramesh':21,'Suresh':22,'Mahesh':23}
d1['Ramesh']=32
d1
#############
s='hello'
s[0]='HH' # error
################
l=[1,2,3]
l[0]=100 # ans
```

Out[31]:

```
{'Ramesh': 32, 'Suresh': 22, 'Mahesh': 23}
```

- dictionary is mutable
- list is mutable
- string is immutable
- tuple is immutable

In [32]:

```
d1={'Fruites':{'Apple':'Kahsmir'}}
d1
```

Out[32]:

```
{'Fruites': {'Apple': 'Kahsmir'}}
```

In [33]:

```
d2={'Fruites':{'Apple':['Kahsmir']}}
d2
```

Out[33]:

```
{'Fruites': {'Apple': ['Kahsmir']}}
```

In [34]:

```
d3={'Fruites':[{'Apple':'Kahsmir'}]}
d3
```

Out[34]:

```
{'Fruites': [{'Apple': 'Kahsmir'}]}
```

In [37]:

```
d4={'Fruites':['Kahsmir']}
d4
```

Out[37]:

```
{'Fruites': ['Kahsmir']}
```

In [40]:

```
d1={'Fruites':{'Apple':'Kahsmir'}}
d1['Fruites']['Apple']
```

Out[40]:

```
'Kahsmir'
```

In [43]:

```
d2={'Fruites':{'Apple':['Kahsmir']}}
d2['Fruites']['Apple'][0]
```

Out[43]:

```
'Kahsmir'
```

In [49]:

```
d3={'Fruites':[{'Apple':'Kahsmir'}]}
d3['Fruites'][0]['Apple']
```

Out[49]:

```
'Kahsmir'
```

In [51]:

```
d4={'Fruites':['Kahsmir']}
d4['Fruites'][0]
```

Out[51]:

```
'Kahsmir'
```

In [59]:

```
d5={'Fruites':['Apple',{'Kashmir':['India']}]}
d5['Fruites'][1]['Kashmir'][0]
```

Out[59]:

```
'India'
```

In [66]:

```
d6={'Fruites':
    {'Benganapply':
     {'Mango':
      {'Nagpur':
       {'MH':
        {'shivaji':
         'Shambaji'
        }
       }
      }
     }
    }
   }
d6
```

Out[66]:

```
{'Fruites': {'Benganapply': {'Mango': {'Nagpur': {'MH': {'shivaji': 'Shambaj
i'}}}}}}
```

In [72]:

```
d6['Fruites']['Benganapply']['Mango']['Nagpur']['MH']['shivaji']
```

Out[72]:

```
'Shambaji'
```

In [73]:

```
dir({})
```

Out[73]:

```
['__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 [ ]:

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

```

# clear copy keys values items pop popitem
```