Dictionary -- is a mapping datatype-- Key Value Pair for example English Dictionary

Mutable Data type

dict = {key1:val1,key2:val2,key3:val3}

cruly braces, comma, pair

```
emp={'emp1':1,'name':'Ishita','designation':'PGT','sub':'cs'}
```

```
emp
```

```
{'emp1': 1, 'name': 'Ishita', 'designation': 'PGT', 'sub': 'cs'}
```

```
dt={} # empty dictionary
type(dt)
```

```
dict
```

```
dt['data']=400
dt
```

```
{'data': 400}
```

```
dt['data2']=600
dt
```

```
{'data': 400, 'data2': 600}
```

Keys should be immutable

```
d1={1:2,2:2,3:3}
d2={"abc":"xyz","pqr":"stu"}
#d3={[1,2,3]:"list1",[4,5,6]:"list3"}
d4={(1,2,3):"list1",(4,5,6):"list3"}
```

Accessing elements through keys

```
emp['name']
```

```
'Ishita'
```

```
emp['sub']
```

```
'cs'
```

```
emp
```

Traversing a dictionary

```
emp.keys()
```

```
dict_keys(['emp1', 'name', 'designation', 'sub'])
```

```
emp.values()
```

```
dict_values([1, 'Ishita', 'PGT', 'cs'])
```

```
(emp.keys())list
```

```
['emp1', 'name', 'designation', 'sub']
```

```
list(emp.values())
```

```
[1, 'Ishita', 'PGT', 'cs']
```

```
for val in emp:
  print(val,emp[val],end=" ")
```

```
emp1 1 name Ishita designation PGT sub cs
```

Unordered and not a sequence. Unique keys

```
data1={1:100,2:200,3:300,4:200}
data1
```

```
{1: 100, 2: 200, 3: 300, 4: 200}
```

```
data2={1:100,2:200,3:300,2:400} # NO ERROR
data2
```

```
{1: 100, 2: 400, 3: 300}
```

Mutable

```
emp['sub']="English"
emp
```

```
{'emp1': 1, 'name': 'Ishita', 'designation': 'PGT', 'sub': 'English'}
```

```
emp['sal']=78000 # Adding a new key value pair
emp
```

```
{'emp1': 1,
 'name': 'Ishita',
 'designation': 'PGT',
 'sub': 'English',
 'sal': 78000}
```

Creating dictionary with dict() FUNCTION

```
std1=dict(roll=1,name="ABC",per=70)
std2=dict({'roll':1,'name':"XYZ",'per':70})
std2
```

```
{'roll': 1, 'name': 'XYZ', 'per': 70}
```

NESTED DICTIONARY

```
classroom = {'Alice': {'math': 90, 'english': 85, 'history': 92}, 'Bob': {'math': 88, 'english': 79,
```

```

```

```
alice_math_grade = classroom['Alice']['math']
alice_math_grade
```

```
90
```

```
classroom['Bob']['english']=89
classroom['Bob']['english']
classroom['David']= {'math': 70, 'english': 80, 'history': 62}
classroom
```

```
{'Alice': {'math': 90, 'english': 85, 'history': 92},
 'Bob': {'math': 88, 'english': 89, 'history': 95},
 'David': {'math': 70, 'english': 80, 'history': 62}}
```

Deleting

```
emp
```

```
{'emp1': 1,
 'name': 'Ishita',
 'designation': 'PGT',
 'sub': 'English',
 'sal': 78000}
```

```
del emp['sal']
emp
```

```
{'emp1': 1, 'name': 'Ishita', 'designation': 'PGT', 'sub': 'English'}
```

```
#del emp['DOJ']
emp
```

```
{'emp1': 1, 'name': 'Ishita', 'designation': 'PGT', 'sub': 'English'}
```

POP

```
#emp.pop()  TypeError
emp.pop('sub')
#emp.pop('add')
#emp.pop('add','JKD') # no error, key value pair is first added then popped
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
/tmp/ipython-input-3456094620.py in <cell line: 0>()
      1 #emp.pop()  TypeError
----> 2 emp.pop('sub')
      3 #emp.pop('add')
      4 #emp.pop('add','JKD') # no error, key value pair is first added then popped

KeyError: 'sub'
```

in, not in --> only of key, dict.values()

```
emp
'Ishita' in emp.values()
```

```
True
```

len() clear()

```
'name'  in emp
#'Ishita' in emp.values()
```

```
emp.clear()
len(emp)
```

```
0
```

get() Most Important Topic

```
#emp['name']
#emp['nm']
#emp.get('name')
#emp.get('nm')
#emp.get('nme','Akshita')
```

```
'Akshita'
```

```
emp
```

```
{'emp1': 1, 'name': 'Ishita', 'designation': 'PGT', 'sub': 'cs'}
```

Items as key value tuple

```
(emp.items())
```

```
dict_items([('emp1', 1), ('name', 'Ishita'), ('designation', 'PGT'), ('sub', 'cs')])
```

```
for key,val in emp.items():
  print(key,val,sep="#$",end="   ")
```

```
emp1#$1   name#$Ishita   designation#$PGT   sub#$cs
```

```
for val in emp.values():
  print(val,end="   ")
```

```
1   Ishita   PGT   cs
```

```
for k in emp:
  print(k,end="  ")
```

```
emp1  name  designation  sub
```

```
for k in emp:
  print(k,emp[k],end="  ")
```

```
emp1 1  name Ishita  designation PGT  sub cs
```

```
list(emp.items())
```

```
[('emp1', 1), ('name', 'Ishita'), ('designation', 'PGT'), ('sub', 'cs')]
```

POPITEM()

```
emp
emp.popitem()
```

```
('sub', 'cs')
```

Update

```
d1={1:'Apple',2:'Banana',3:'Papaya'}
d2={4:'Orange',5:'Kiwi'}
```

```
d1.update(d2)
d1
```

```
{1: 'Apple', 2: 'Banana', 3: 'Papaya', 4: 'Orange', 5: 'Kiwi'}
```

```
d1={1:'Apple',2:'Banana',3:'Papaya'}
d3={2:'Orange',5:'Kiwi'}
d1.update(d3)
d1
```

```
{1: 'Apple', 2: 'Orange', 3: 'Papaya', 5: 'Kiwi'}
```

fromkeys()

```
nd1=dict.fromkeys([2,3,4,5],100)
nd2=dict.fromkeys([2,3,4,5],(100,200))
```

```
nd1
```

```
{2: 100, 3: 100, 4: 100, 5: 100}
```

```
nd2
```

```
{2: (100, 200), 3: (100, 200), 4: (100, 200), 5: (100, 200)}
```

```
nd1[6]=200
nd1
```

```
{2: 100, 3: 100, 4: 100, 5: 100, 6: 200}
```

setdefault()

```
nd1.setdefault(7,300)
nd1
```

```
{2: 100, 3: 100, 4: 100, 5: 100, 6: 200, 7: 300}
```

```
nd1[7]=400
nd1
```

```
{2: 100, 3: 100, 4: 100, 5: 100, 6: 200, 7: 400}
```

```
nd1.setdefault(9,9000)
```

```
9000
```

```
nd1
```

```
{2: 100, 3: 100, 4: 100, 5: 100, 6: 200, 7: 400, 9: 9000}
```

```
nd1.setdefault(6,700)
```

```
200
```