Dictionary

In [ ]:

```
# Dictionary

1- denote : {}

2- Dictionary is mutable(can be modified after creation)

3- its not follow the order (unordered )

4- it is store the data as key and value format.  eg . {"name" : "John","age": 2
```

In [1]:

```
# Dictionary Operations

# empty Dictionary

d = {}
print(d)

student = {"name": "Bob","age" : 25,"grade" : "A"}
```

```
{}
```

In [2]:

```
lst = [('a',1),('b',2)]

# converting list of tuples to dict
dict(lst)
```

Out[2]:

```
{'a': 1, 'b': 2}
```

In [4]:

```
d_com = {x: x**2 for x in range(5)} #
d_com
```

Out[4]:

```
{0: 0, 1: 1, 2: 4, 3: 9, 4: 16}
```

In [5]:

```
# Accessing
student = {"name": "Bob","age" : 25,"grade" : "A"}
```

In [10]:

```
student["grade"]
```

Out[10]:

```
'A'
```

In [22]:

```
student.get("grade","Unknown key defined")
```

Out[22]:

```
'A'
```

In [26]:

```
# modifying

student["location"] = "Mumbai"
print(student)
```

```
{'name': 'Bob', 'age': 25, 'grade': 'A', 'location': 'Mumbai'}
```

In [27]:

```
print(student)
```

```
{'name': 'Bob', 'age': 25, 'grade': 'A', 'location': 'Mumbai'}
```

In [ ]:

```python
del student["grade"]
del student
```

In [31]:

```python
# dict methods
student = {'name': 'Bob', 'age': 25, 'grade': 'A', 'location': 'Mumbai'}

print(student.keys())
print(student.values())
print(student.items())
print(student.pop("location"))
```

```
dict_keys(['name', 'age', 'grade', 'location'])
dict_values(['Bob', 25, 'A', 'Mumbai'])
dict_items([('name', 'Bob'), ('age', 25), ('grade', 'A'), ('location', 'Mumba
i')])
Mumbai
```

In [32]:

```python
student
```

Out[32]:

```
{'name': 'Bob', 'age': 25, 'grade': 'A'}
```

In [ ]:

```python
# []---->list
# ()-----> tuple
# {} ---> dict curly bracket

student[]
student.get() # function
print() --->
```

In [34]:

```python
# Membership
"name" in student #
```

Out[34]:

```
True
```

In [35]:

```python
lst = [1,2,43,4,5,6,6,7]

43 in lst
```

Out[35]:

```
True
```

In [36]:

```python
"name" not in student #
```

Out[36]:

```
False
```

In [37]:

```python
student
```

Out[37]:

```
{'name': 'Bob', 'age': 25, 'grade': 'A'}
```

In [38]:

```python
student["grade"]
```

Out[38]:

```
'A'
```

In [39]:

```python
student.get("grade")
```

Out[39]:

```
'A'
```

Sets

In [ ]:

```
1.  set is denoted by {}

2- Set is mutable

3- there is no key and values associated, only value will be there.

4- No duplicate value allowed

5- no follow the specific order
```

In [43]:

```
s = set()
print(s)
```

```
{1}
```

In [41]:

```
s1  = {1}
type(s1)
```

Out[41]:

```
set
```

In [44]:

```
numbers = set([1,24,3,4,5,5])
print(numbers)
```

```
{1, 3, 4, 5, 24}
```

In [45]:

```
numbers = {1,24,3,4,5,5}
print(numbers)
```

```
{1, 3, 4, 5, 24}
```

In [47]:

```
values = set("Hello")
print(values)
```

```
{'Hello'}
```

In [51]:

```
# set accessing
numbers = {1,24,3,4,5,5}
for num in numbers:
    print(num)
```

```
1
3
4
5
24
```

In [62]:

```
# sets operation
animal = {"lion","dear","elephant","wolf"}
animal.add("Zebra")
animal
```

Out[62]:

```
{'Zebra', 'dear', 'elephant', 'lion', 'wolf'}
```

In [63]:

```
animal
```

Out[63]:

```
{'Zebra', 'dear', 'elephant', 'lion', 'wolf'}
```

In [64]:

```
animal.update(["Zebra","monkey"])
```

In [65]:

```
animal
```

Out[65]:

```
{'Zebra', 'dear', 'elephant', 'lion', 'monkey', 'wolf'}
```

In [66]:

```
animal.remove("Zebra")
```

In [67]:

```
animal
```

Out[67]:

```
{'dear', 'elephant', 'lion', 'monkey', 'wolf'}
```

In [69]:

```
'dear' not in animal
```

Out[69]:

```
False
```

In [75]:

```
set1 = {1,2,3,4}
set2 = {2,3,4,5}

#union
union = set1 | set2
```

In [76]:

```
union
```

Out[76]:

```
{1, 2, 3, 4, 5}
```

In [77]:

```
#union
intersection = set1 & set2
intersection
```

Out[77]:

```
{2, 3, 4}
```

In [79]:

```
#union
difference = set2 - set1
difference
```

Out[79]:

```
{5}
```

In [ ]:

```

```