WORKING WITH PYTHON SET DATASTRUCTURE..!!

Python Sets
SET is an unordered collection of unique elements

How to create a set?
1 Insertion order is not preserved.But we can sort the elements.
2 Indexing and slicing not allowed for the set.
3 Heterogeneous elements are allowed.
4 We can represent set elements within curly braces and with comma seperation
5 We can apply mathematical operations like union,intersection,difference etc

There are currently two built-in set types:
1 set ==> The set type is mutable
2 frozenset. ==> It is immutable and hashable

Example:
```
FruitBasket={"Apple","Mango","orange","banana","Apple","Mango"}
print(type(FruitBasket))
print(FruitBasket)
```

Example:
We should not use empty curly brackets, treat as dict.!
```
PySet={}
print(type(PySet)) #<class 'dict'>
```

Set not allowed duplicates but list allows:
```
Numbers=[1,2,3,4,1,2,3,4,5]
UniqueNumbers=set(Numbers)
print(UniqueNumbers) #{1,2,3,4,5}
```

Iteration Over Sets
We can move over each of the items in a set using a loop. However, since sets are unorded

Example:
```
NumSet = set([0, 1, 2, 3, 4, 5])
for n in NumSet:
  print(n)
```

Different Python Set Methods

1 add()	2 clear()		
3 copy()		4 difference()	
5 difference_update()	6 discard()		
7 intersection()		8 intersection_update()	
9 isdisjoint()	10 issubset()		
11 issuperset()	12 pop()		
13 remove()	14 symmetric_difference()		
15 symmetric_difference_update()			
16 union()	17 update()		

Python Set add():
It adds a given element to a set. If the element is already present, it doesn't add that element.

Syntax:
set.add(elem)

Example:
```
PySet={}
print(PySet)
PySet.add(1)
print(PySet)#AttributeError: 'dict' object has no attribute 'add'
```

Example:
```
PySet={1}
print(PySet)#{1}
PySet.add(1)
print(PySet)#{1}
```

Example:
```
PySet={1}
print(PySet)
PySet.add(1,2)
print(PySet)
#TypeError: add() takes exactly one argument (2 given)
```

Example:
```
PySet={1}
print(PySet)
PySet.add([1,2])
print(PySet)
#TypeError: unhashable type: 'list'
```

Example:
```
PySet={1}
print(PySet)#{1}
PySet.add((1,2))
print(PySet)#{(1, 2), 1}
```

clear():
It is used to clear all elements from a set.

Syntax:
set.clear()

Example:
```
PySet={1}
print(PySet)#{1}
PySet.clear()
print(PySet)#set()
```

Python Set update():
It adds elements from a set to the set.
OR
A |= B or A.update(B)
Adds all elements of array B to the set A.

Syntax
A.update(B)

NOTE:
A and B are two sets. The elements of set B are added to the set A.

Example:

```
DataSet=set()
DataSet.update(["Data Science", "Big Data"])
print(DataSet)
```

Example:

```
A={10,20,30}; B=[40,50,60,10]
A.update(B,range(5))
print(A)
```