PYTHON

From Simple to Complex
With
Examples

AYESHA NOREEN

Bachelor's in Software Engineering,

Master's in Computer Science

from COMSATS University, Islamabad

NOTE!!!

In these notes Screenshots of practice examples and coding are added. The code files are also available in code folder that contain .ipynb files that are created on Jupyter notebook.

Chapter10
Sets in Python

Sets are immutable, unordered, unindexed and does not allow duplication. Sets uses curly bracket. Set items are unchangable but we can add or remove items from a set. e.g.

If we have a list and we want to remove its duplication than we use this.

```
l=[1,2,3,4,4,4,5,6,6]
s2=list(set(l))
print(s2)
```

- **add(), remove(), discard(), clear(), copy() methods in set**

```
#add remove discard clear copy
s={1,2,3,4.0,1.0,1.1,'ayesha'} #we cannot store list ,dictionary,tuple in set
print(s)    #it does not prints 1.0 bcz 1.0 and 1 are same
#it cannot contain duplication {1, 2, 3, 4.0, 1.1, 'ayesha'}
s.add(5)
print(s) # it prints {1, 2, 3, 4.0, 1.1, 5, 'ayesha'}
s.remove(1.0)
print(s) # it prints {2, 3, 4.0, 1.1, 5, 'ayesha'}
#s.remove(8)
#print(s)  #give error bcz 8 is not present
s.discard(8)
print(s) #8 is not present but give nothing not error
s.clear()
print(s)  #give empty set { }
s1=s.copy()
print(s1)  #same as s
```

```
{1, 2, 3, 4.0, 1.1, 'ayesha'}
{1, 2, 3, 4.0, 1.1, 5, 'ayesha'}
{2, 3, 4.0, 1.1, 5, 'ayesha'}
{2, 3, 4.0, 1.1, 5, 'ayesha'}
set()
set()
```

- **In keyword, looping, union, intersection in set**

```
#in keyword, looping, union and intersection
s={'a','b','c'}
s1={1,2,3,4}
s2={3,4,5,6}
if 5 in s1:
    print("present")
else:
    print("Not present")
#it prints not present
if 2 in s1:
    print("present")
else:
    print("Not present")
#it prints present
for item in s1:
    print(item)
#it print each element of set
union_set=s1|s2
print(union_set)        #prints {1,2,3,4,5,6}
intersection_set=s1&s2
print(intersection_set) #prints {3,4}
```

• Set comprehension

Set comprehension is rarely used in programming

```
#make a set that print square of numbers from 1 to 10 by using set comprehension
square={i**2 for i in range(1,11)}
print(square)
#take a set of names and make a set that prints 1st letter of each name by using set comprehension
names={'ayesha','rimsha','sana'}
first_letter={i[0] for i in names}
print(first_letter)
#it prints
#{64, 1, 4, 36, 100, 9, 16, 49, 81, 25}
#{'s', 'a', 'r'} data is unordered bcz set contain unordered collection of data
```

```
{64, 1, 4, 36, 100, 9, 16, 49, 81, 25}
{'s', 'r', 'a'}
```