Unit 5

Dictionary, list, set, string and Tuple

★. list any six built-in-function of list. Or Explain list methods and its built in functions.

→ (યાદી Function અને list methods)

list :- Python માં list એ એક mutable data structure છે.

list ની અંદર data ને add કરવા કે remove કે modify કરવા માટે Python માં ઘણા built in Function આપેલ છે.

Built in Function :-

1. Append (item) :-

આ method list ની એકદમ end માં એક નવો item add કરવા માટે વપરાય છે.

Syntax : list. append (item)

Example :-

```
my_list = [10, 20, 30]
Print (my_list. append (40))
```

Out Put :

[10, 20, 30, 40]

2. Extend (iterable)

જ્યારે એક list ની અંદર બીજુ list add કરવું હોય, ત્યારે extend () નો ઉપયોગ ક

Syntax : list. extend (another_list)

Example :-

Print (my_list. extend ([50, 60]))

Output :-

[10, 20, 30, 40, 50, 60]

3. insert (index, item)

આ મેથડ ની ઉપયોગ list માં કોઈ ચોક્કસ position પર નવી item add / insert કરવા માટે થાય છે.

Syntax : list. insert (index, item)

Example :-

Print (my_list. insert (1, 15))

Output :-

[10, 15, 20, 40, 50, 60]

4. remove (item)

આ મેથડ list માંથી આપેલી item ની પહેલી Occurrence remove કરવા માટે ઉપયોગ થાય છે.

Syntax : list. remove (item)

Example :

Print (my_list. remove (30))

Output :-

[10, 15, 20, 40, 50, 60]

5. pop (index) :-

ਇਸ ਵਿਧੀ ਰਾਹੀਂ index ਦੇ item ਨੂੰ remove ਕੀਤਾ ਜਾਂਦਾ ਹੈ ਅਤੇ item return ਪਰਤ ਕਰਦਾ ਹੈ. ਜੇ index ਨੂੰ ਆਮਦਤ ਨਹੀਂ ਹੈ ਤਾਂ ਆਖਰੀ item remove ਕਰੇਗਾ.

Syntax: list. pop (index)

Example :-

Print (my_list. pop(2))

Output :-

[10, 15, 40, 50, 60]

6. sort ()

ਇਸ ਵਿਧੀ ਰਾਹੀਂ list ਦੇ items ਨੂੰ ascending ਜਾਂ descending order ਵਿਚ sort ਕਰਦੇ ਹਨ.

Syntax : list. sort ()

Example :-

Print (my_list. sort (reverse = True))

Output :-

[60, 50, 40, 15, 10]

Q.4. Explain tuple operations and functions. 4 Or Write a program to demonstrate tuple functions and operations. (4)

→ Explanation :-

- Python માં Tuple એ એક ordered અને immutable data structure છે. Tuple ની items ને round brackets () ની અંદર ગોઠવવામાં આવે છે.

- list ની જેમ આમાં items add, remove કે modify નથી કરી શકાતા, પણ આપણે આના આધારે પર અલગ-અલગ Operations અને Functions apply કરી શકીએ છીએ.

Operations :-

3) Write a Program to traverse the string using for and while loop.

→ Traversing using for loop.

```
Str1 = input(" Enter a string: ")
Print (" Traversing using for loop:")
for ch in str1 :
    Print (ch)
```

→ Traversing using while loop.

```
Str1 = input (" Enter a String: ")
i = 0
Print (" Traversing using while loop.")
while i < len (Str1):
    Print (Str1 [i])
    i = i+1
```

→ Output for loop.

Enter a string : Python

Traversing using for loop

P

y

t

h
o
n

-> Output While loop.

Enter a String : While

Traversing using while Loop.
w
h
i
l
e

Q.I Write a Program to Check if a Substring is Present in a given string. (04)

-> Code :-

```
# Substring Check in Python

main_string = input ("Enter the main string:")
sub_string = input ("Enter the Substring to search: ")

if sub_string in main_string :
```

→ Code :-

```
main_string = "Python Programming"
Sub_string = "Python"

if Sub_string in main string :
    Print ("~~Sub String~~ Sub string is Present")
else :
    Print ("Sub string is not Present")
```

Output :-

Sub string is Present

2. Write a Program to Perform the Following list Operations. (4)

(a) To swap given two elements in a list.

→ Code :-

```
my_list = [10, 20, 30, 40]

Print (" Before swapping :", my_list)

my_list[1], my_list[3] = my_list[3], my_list[1]

Print ("After swaping :", my_list)
```

Output :-

Before swapping : [10, 20, 30, 40]
After Swapping : [10, 40, 30, 20]

(b) To find the sum of element in a list

-> Code :-

```
my_list = [10, 20, 30, 40]

total = sum(my_list)

Print("Sum of elements in the list is:", total)
```

Output :-

Sum of elements in the list is : 100

3. Write a program to find the smallest and largest element in a given list. (03)

-> ~~Wr~~ Code :-

```
list = [10, 20, 25, 5, 40]

Smallest = min(list)
largest = max(list)

Print("Smallest element is :", smallest)
Print("Largest element is :", Largest)
```

Output :-

Smallest element is : 5
Largest element is : 40

6. Write a program to demonstrate the set fu. & of.

4. Explain different set functions and operations with example. (07)

-> • Set :- Set એ Python માં unique values ની unordered collection છે. set માં duplicate values આપોઆપ દૂર થાય છે. set નો ઉપયોગ data ને store અને manage કરવા માટે થાય છે.

s = { 1, 2, 3 }

• Set methods (Built in functions)

1. add (x) :- set માં એક નવી value ઉમેરવા માટે ઉપયોગ થાય છે.

Code :-

```
s.add(6)
Print(s)
```

Output :-

{ 1, 2, 3, 6 }

2. Update (iter) :- એક થી વધુ values set માં ઉમેરવા માટે ઉપયોગ થાય છે.

Code :-

```
s.update([7, 8, 9])
Print(s)
```

Output :-

{ 1, 2, 3, 7, 8, 9 }

3. Remove (x) :- Set માંથી value દૂર કરે છે જો value હાજર ન હોય તો error આપે છે.

Code :-

```
S.remove(3)
Print(S)
```

Output :-

{1, 2}

4. discard (x) :- Set માંથી value દૂર કરે છે પણ value હાજર ન હોય તો error આપતું નથી.

Code :-

```
S.discard(1)
Print(S)
```

Output :-

{2, 3}

5. Pop() :- Set માંથી random value remove કરે છે.

Code :-

```
S.pop()
Print(S)
```

Output :-

{1, 3}

- Set Operations :-

1. union () or | :- ઘણા sets નો એક value duplicate થઈ જાય છે.

Code :-

```
A = {1, 2}
B = {2, 3}

Print (A | B)
```

Output :-

{1, 2, 3}

2. intersection () or & :- ઘણા sets માંથી common values આવે છે.

Code :-

```
A = {1, 2, 3}
B = {2, 3, 4}

Print (A & B)
```

Output :-

{2, 3}

3. difference () or - :- પ્રથમ set માં હોય પણ બીજા set માં ના હોય એવી values આપે છે.

Code :-

```
A = {1, 2, 3}
B = {2, 3}

Print (A-B)
```

Output :-

```
{1}
```

4. issubset () :- જો કોઈ set ના બધા elements બીજા set માં હોય તો True આપે છે.

Code :-

```
A = {1, 2}
B = {1, 2, 3}

Print (A. issubset (B))
```

Output :- True

5. issuperset () :- જો કોઈ set બીજા set ના બધા elements ધરાવે તો True આપે છે.

Code :-

```
A = {1, 2, 3}
B = {1, 2}

Print (A. issuperset (B))
```

Output :-

True

5. Explain different dictionary functions and methods with example. (07)

→ Dictionary :- Dictionary એ python માં data ની key-value pair ના સ્વરૂપમાં store કરવાનું data structure છે. દરેક key unique હોય છે.

```
Student = {
    "name" : "Rahul",
    "roll" : 101,
    "marks" : 85
}
```

Dictionary functions અને methods :-

1. Keys () :-

Dictionary માં હાજર બધા keys મેળવવા માટે ઉપયોગ થાય છે. Data structure માં દરેક keys બરાબર છે કે નહીં જોવા માટે વપરાય છે.

Syntax :- dictionary.keys()

Code :-

```
Print (Student.keys())
```

Output :-

```
dict_keys (['name', 'roll', 'marks'])
```

2. values () :- Dictionary ની બધા values મેળવવા માટે ઉપયોગ થાય છે. stored data ને access અને display કરવા માટે ઉપયોગી છે.

Syntax :- dictionary. values ()

Code :-
Print (Student. values ())

Output :-
dict_values (['Rahul', 101, 85])

3. items () :- Dictionary ની key-value pairs મેળવવા માટે ઉપયોગ થાય છે. loop દ્વારા સંયુક્ત dictionary data access કરવા માટે ઉપયોગી છે.

Syntax :- dictonary. items ()

Code :-
Print (Student. items())

Output :-
dict_items ([('name', 'Rahul'), ('roll', 101), ('marks', 85)])

4. get () :- કોઈ specific key ની value safely મેળવવા માટે ઉપયોગ થાય છે. key ખોટી હોય તો error આપવા કરતાં value return કરે છે.

Syntax :- dictionary. get (key)

Code :-

```
Print ( Student. get ("name"))
```

Output :-

Rahul.

5. Update () :- Dictionary માં નવી entry ઉમેરવા અથવા જુની value બદલવા માટે ઉપયોગ થાય છે. Data ને modify અને manage કરવા માટે મદદરૂપ થાય છે.

Syntax :- dictionary. update ({key: value})

Code :-

```
Student. update ({"grade": "B"})
Print (student)
```

Output :- { 'name' : 'Rahul',
'roll' : 101 ,
'marks' : 85 ,
'grade' : 'B' }

6. Pop () :- Dictionary માંથી Specific key-value pair દૂર કરવા માટે ઉપયોગ થાય છે. અનાવશ્યક data delete કરવા માટે ઉપયોગી છે.

Syntax :- dictionary. Pop (key)

Example :-

```
Student. Pop ("roll")
Print (Student)
```

Output :-

{'name' : 'Ruhvi'}

7. Clear () :- Dictionary ની બધાં data remove કરવા માટે ઉપયોગ થાય છે. Dictionary ને empty બનાવવા માટે ઉપયોગ છે.

syntax :- dictionary. clear ()

example :-

```
Student. Clear ()
print (Student)
```

Output :-

{}

Q. 6 Write a program to demonstrate the dictionaries functions and operations. (07)

→ Code :-

```
Student = {
        "name" : "Jerry",
        "roll" : 2024,
        "marks" : 84
}

Print ("Original Dictionary :")
Print (Student)

Print ("Keys:", Student. keys ())

Print ("Values:", Student. values ())

Print ("Items :", Student. items ())

Print ("Name :", Student. get ("name"))

Student. Update ({"City" : "Ahmedabad"})
Print (Student)

Student. POP ("roll")
Print (Student)

new_dict = Student. Copy ()
Print (new_dict)
```

```
Print (" length of Dictionary:", len(student))

Print (student.clear())
```

Output :-

```
Original Dictionary:
{'name': 'Jerry', 'roll': 2024, 'marks': 84}

Keys: dict_keys(['name', 'roll', 'marks'])

Values: dict_values(['Jerry', 2024, 84])

Items: dict_items([('name', 'Jerry'),
                   ('roll', 2024),
                   ('marks', 84)])

Name : Jerry

{'name': 'Jerry', 'roll': 2024, 'marks': 84,
 'city': 'Ahmedabad'}

{'name': 'Jerry', 'marks': 84, 'city':
 'Ahmedabad'}

{'name': 'Jerry', 'marks': 84, 'city':
 'Ahmedabad'}

length of Dictionary: 0

{}
```

7. Explain tuple Operations and Functions. OR Write a program to demonstrate tuple functions and operations. (4).

→ Explaination :-

- Python માં Tuple એ એક Ordered અને immutable data structure છે. Tuple ની items ને round brackets () ની અંદર લખવામાં આવે છે.

- list ની જેમ આમાં items add, remove કે modify નથી કરી શકતા, પણ આપેલ આમાં પર અલગ-અલગ Operations અને Functions apply કરી શકીએ છીએ.

Operations :-

```
t = (10, 20, 30, 20)

Print ("Tuple = ", t)

# len()
Print ("Length = ", len(t))

# max()
Print ("Maximum value = ", min(t))

# min()
Print ("minimum value = ", min(t))
```

```
# Count
Print ("Count of 20 = ", t.count(20))

# index()
Print ("Index of 30 = ", t.index(30))

# Concatenation
t2 = (40, 50)
Print ("Concatenation = ", t + t2)

# Repetition
Print ("Repetition = ", t * 2)
```

Output :

Tuple = (10, 20, 30, 20)

Length = 4
Maximum Value = 30
Minimum Value = 10

Count of 20 = 2
Index of 30 = 2

Concatenation = (10, 20, 30, 20, 40, 50)

Repetition = (10, 20, 30, 20, 10, 20, 30, 20)