Comprehension

Thursday, July 31, 2025 7:10 PM

It is a phenomenon of creating the new collection by reducing the number of instructions.

It can be performed only on mutable datatypes.

There are 3 types of comprehension:

i. List Comprehension
ii. Set Comprehension
iii. Dictionary Comprehension

1. List Comprehension: It is a phenomenon of creating new list by reducing the number of instructions.

   Syntax:

   i. Var=[val for var in collection if condition] #writing condition is optional
      or
   ii. Var=[TSB if condition else FSB for var in collection]
      or
   iii. Var=[val1,val2 for var1 in collection1 for var2 in collection 2]

   Eg:

```
#WAP to create the list consists of 1 to 10 int values
'''out=[i for i in range(1,11)]
print(out)'''

'''print([i for i in range(1,11)])'''

#WAP to create a list consists of square of each int between 1 to 20 if it is
#multiple of 3
'''print([i**2 for i in range(1,21) if i%3==0])'''

#WAP to store the square of val if it is even else store cube in between 1 to 10
'''print([i**2 if i%2==0 else i**3 for i in range(1,11)])'''

#WAP to get the following output
'''[('A',1),('A',2),('A',3),('B',1),('B',2),('B',3)'''
'''print([(i,j) for i in 'AB' for j in [1,2,3]])'''

#WAP to get the following output
#s='programs on comprehension'
#out=['programs','on','cn']
'''s='programs on comprehension'
s1=s.split()
print([i if len(i)%2==0 else i[0]+i[-1] for i in s1])'''
```

2. Set Comprehension: It is the phenomenon of creating new set with less number of instructions.

Difference from Set:

i. In list '[]' is used but in set '{}' is used.
ii. In list the duplicate value gets stored but in set no duplicate values allowed.
iii. In list data is stored in sequence but in set it is stored randomly.

Syntax:

i. Var={val for var in collection if condition} #writing condition is optional
or
ii. Var={TSB if condition else FSB for var in collection}
or
iii. Var={val1,val2 for var1 in collection1 for var2 in collection 2}

Eg:

```
#WAP to find the sqrt of all the int between 10 to 100
'''import math
print({math.sqrt(i) for i in range(10,101)})'''

#WAP to get the following output
#s='Data science For Business'
#out={'DATA','ecneics','FOR','BUSINESS'}
'''s='Data science For Business'
s1=s.split()
print({i.upper() if 'A'<=i[0]<='Z' else i[::-1] for i in s1})'''

#WAP to make each student opt every subject from the set
'''sn=['A','B','C']
sb=['Python','DS','DA']
print({(i,j) for i in sn for j in sb})'''
```

- Zip(): It is used in the dictionary comprehension.
  When we want to traverse through multiple collection using single for loop, then zip() is used.

  Syntax:
  For (var1, var2, var3…) in zip(coll1, coll2, coll3,…..):
  S B

  Where number of variables == Number of collection

  It will consider the collection with least length and will execute accordingly.

  Eg:

```
for (i,j,k) in zip('ABCD','123','456'):
   print(i,j,k)
```

3. Dictionary Comprehension: It is the phenomenon of creating new dictionary with less number of instructions.

Syntax:

```
Var={k:v for var in collection if condition} #if condition is optional
                Or
Var={k:v for var1,var2 in zip(coll1, coll2)}
                Or
Var={k:v1 if condition else v2 for var1,var2 in zip(coll1, coll2)}
```

Eg:

```
#WAP to get the following output
#out={1:1,2:4,3:9,4:16,5:25}
'''print({i:i**2 for i in range(1,6)})'''
```

```
#WAP to get the following output
#s='hii hello how'
#out={'hii':3.......}
#out1=('howw':4}
'''s='hii hello howw'
print({i:len(i) for i in s.split()})
print({i:len(i) for i in s.split() if len(i)%2==0})'''
```

```
#WAP to map 2 lists in the form of dictionary
'''l1=[1,2,3,4]
l2=[10,20,30,40,50,60]
print({i:j for i,j in zip(l1,l2)})'''
```

```
#WAP to get the following output
'''l=['abcd','nayan','data','aba']
#out={'abcd':'dcba','nayan':5.....}
print({i:len(i) if i==i[::-1] else i[::-1] for i in l})'''
```

```
##WAP to get the following output
#out={1:1,2:4,3:27,4:256,5:3125}
'''print({i:i**i for i in range(1,6)})'''
```

Assignment:

1. WAP to get the following output
   o/p: {1:1,2:2,3:3,4:4,5:25,6:6,7:7,8:8,9:9,10:100}

2. WAP to get the output as
   Key: char like A,B,C,D etc
   Value: ASCII value of the char