Python List:

1. Create your own lists otherwise use the List1 = ['hello','great','learning'] to operate the below Python list Operations?

```
A. Repetition (*)
B. Concatenation (+)
C. Membership
D. Length
E. Iteration
```

```
List1 = ['hello','great','learning']
```

1.A. Repetition:

```
List1 * 2
```

```
['hello', 'great', 'learning', 'hello', 'great', 'learning']
```

1.B. Concatenation (+):

```
List1 + List1
```

```
['hello', 'great', 'learning', 'hello', 'great', 'learning']
```

1.C. Membership:

```
'word' in List1
```

```
False
```

```
'word' not in List1
```

```
True
```

1.D. Length:

```
len(List1)
```

```
3
```

1.E. Iteration:

```
for char in List1:
  print(char)
```

```
hello
great
learning
```

2.1. Create your own lists to operate the below Python List Built-in functions?

otherwise use the

```
List1 = [15, 300, 2700, 821]
List2 = [12, 2]
List3 = [34, 567, 78]
```

```
A.  max(list)
B.  min(list)
C.  list(seq)
```

```
List1 = [15, 300, 2700, 821]
List2 = [12, 2]
List3 = [34, 567, 78]
```

2.A) max():

```
max(List1)
```

```
2700
```

```
max(List2)
```

```
12
```

```
max(List3)
```

```
567
```

2.B) min():

```
min(List1)
```

```
15
```

```
min(List2)
```

```
2
```

```
min(List3)
```

```
34
```

2.C) list:

```
list(List1)
```

```
[15, 300, 2700, 821]
```

```
list(List2)
```

```
[12, 2]
```

```
list(List3)
```

```
[34, 567, 78]
```

3. Create your own lists to operate the below Python List built-in methods

```
A.  list.append()
B.  list.clear()
C.  List.copy()
D.  list.count()
E.  list.extend()
F.  list.index()
G.  list.insert()
H.  list.pop()
I.  list.remove()
J.  list.reverse()
K.  list.sort()
```

```
Vol=["a","e","i","o"]
```

3.A) append:

```
Vol.append("u")
```

```
print(Vol)
```

```
['a', 'e', 'i', 'o', 'u']
```

3.B) list.clear():

```
Vol.clear()
```

```
print(Vol)
```

```
[]
```

3.C) List.copy()

```
Alp = Vol.copy()
```

```
Alp
```

```
['a', 'e', 'i', 'o']
```

3. D) list.count(obj):

```
Vol.count("a")
```

```
1
```

3.E) list.extend(seq):

```
Vol_other=["u"]
```

```
Vol.extend(Vol_other)
```

```
print(Vol)
```

```
['a', 'e', 'i', 'o', 'u']
```

3.F) list.index(obj):

```
Vol.index("i")
```

```
2
```

3.G) list.insert(index, obj):

```
Vol.insert(1,"b")
```

```
print(Vol)
```

```
['a', 'b', 'e', 'i', 'o', 'u']
```

3.H) list.pop(obj=list[-1]):

```
Vol.pop()
```

```
'u'
```

```
print(Vol)
```

```
['a', 'b', 'e', 'i', 'o']
```

3.I) list.remove(obj):

```
Vol.remove('b')
```

```
print(Vol)
```

```
['a', 'e', 'i', 'o']
```

3.j) list.reverse():

```
Vol.reverse()
```

```
print(Vol)
```

```
['o', 'i', 'e', 'a']
```

3.K) list.sort([func]):

```
Vol.sort()
```

```
print(Vol)
```

```
['a', 'e', 'i', 'o']
```

Python Tuple

Create your own Tuple to operate the below Python tuple Operations ?

otherwise use the

```
Tuple1 = ('a','b','c','d')
Tuple2 = ('e','f','g','h')
```

1. Tuple Operations

```
A. Repetition (*)
B. Concatenation (+)
C. Membership
D. Length
E. Iteration
```

```
Tuple1 = ('a','b','c','d')
Tuple2 = ('e','f','g','h')
```

1.A. Repetition:

```
Re_Tuple= Tuple1 * 2 , Tuple2 * 2
```

```
print(Re_Tuple)
```

```
(('a', 'b', 'c', 'd', 'a', 'b', 'c', 'd'), ('e', 'f', 'g', 'h', 'e', 'f', 'g', 'h'))
```

1.B. Concatenation (+):

```
Add_Tuple= Tuple1+Tuple2
```

```
print(Add_Tuple)
```

```
('a', 'b', 'c', 'd', 'e', 'f', 'g', 'h')
```

1.C) Membership:

```
'a' in Tuple1
```

```
True
```

```
'b' in Tuple2
```

```
False
```

```
'g' not in Tuple1
```

```
True
```

1.D) Length:

```
len(Tuple1)
```

```
4
```

```
len(Tuple2)
```

```
4
```

1.E) Iteration:

```
for alph in Tuple1:
  print(alph)
```

```
a
b
c
d
```

```
for alph in Tuple2:
  print(alph)
```

```
e
f
g
h
```

2.Create your own Tuple to operate the below Python tuple inbuilt functions ?

otherwise use the

```
Tuple1 = (1,4,2,4,5,6,3,5,4,6,77,8,7,7,876,89,8765,4,5,1,876,9,3456,4234)
```

```
A. max(Tuple)
B. min(Tuple)
C. Tuple(seq)
```

```
Tuple1 = (1,4,2,4,5,6,3,5,4,6,77,8,7,7,876,89,8765,4,5,1,876,9,3456,4234)
```

2.A) max():

```
max(Tuple1)
```

```
8765
```

2.D) min():

```
min(Tuple1)
```

```
1
```

2.E) Tuple:

```
tuple(Tuple1)
```

```
(1,
 4,
 2,
 4,
 5,
 6,
 3,
 5,
 4,
 6,
 77,
 8,
 7,
 7,
 876,
 89,
 8765,
 4,
 5,
 1,
 876,
 9,
 3456,
 4234)
```

Python Set

Create your own Set to operate the below Python Set Operations ?

otherwise use the below set:

```
Set1 = {1,4,2,4,5,6,3,5,4,6,77,8,7,7,876}
Set2 = {3,432,5,6,4,6,7,6,5,6,54,567,5}
```

1. Set Operations

```
A. Union
B. Intersection
C. Difference
D. Symmetric difference
```

```
Set1 = {1,4,2,4,5,6,3,5,4,6,77,8,7,7,876}
Set2 = {3,432,5,6,4,6,7,6,5,6,54,567,5}
```

- 1.A) Union:

```
Set1.union(Set2)
```

```
{1, 2, 3, 4, 5, 6, 7, 8, 54, 77, 432, 567, 876}
```

- 1.B. Intersection

```
Set1.intersection(Set2)
```

```
{3, 4, 5, 6, 7}
```

- 1.C. Difference

```
Set1.difference(Set2)
```

```
{1, 2, 8, 77, 876}
```

- 1.D. Symmetric difference:

```
Set1.symmetric_difference(Set2)
```

```
{1, 2, 8, 54, 77, 432, 567, 876}
```

Python Dictionary

- Create your own Dictionary to operate the below Python Built-in Dictionary functions

  otherwise use the below Dictionary:

```
dict = {'Name': 'Student', 'Age': 27};
```

```
A.  len(dict)
B.  str(dict)
C.  type(variable)
```

```
dict_1 = {'Name': 'Student', 'Age': 27};
```

- 1.A) len(dict):

```
len(dict_1)
```

```
2
```

- 2.B) str(dict):

```
str(dict_1)
```

```
'{'Name': 'Student', 'Age': 27}'
```

- 1.C) type(variable):

```
type("Name")
```

```
str
```

- 2.Create your own Dictionary to operate the below Python Built-in Dictionary methods

  otherwise use the below Dictionary:

```
dictionaries = {0:" Data",1: "GREAT", 2: "LEARNING",3:"Python",4:"Happy"}
```

```
A.  dic.clear()
B.  dict.copy()
C.  dict.fromkeys()
D.  dict.get(key[, value])
E.  dict.items()
F.  dict.keys()
```

```
G.  dict.setdefault()
H.  dict.update()
I.  dict.values()
```

```
dic = {0:" Data",1: "GREAT", 2: "LEARNING",3:"Python",4:"Happy"}
```

2.A) dic.clear():

```
dic.clear()
```

```
print(dic)
```

```
{}
```

2.B) dict.copy():

```
A=dic.copy()
```

```
print(A)
```

```
{0: ' Data', 1: 'GREAT', 2: 'LEARNING', 3: 'Python', 4: 'Happy'}
```

2.C) dict.fromkeys():

```
dict.fromkeys(dic,'Day')
```

```
{0: 'Day', 1: 'Day', 2: 'Day', 3: 'Day', 4: 'Day'}
```

2.D) dict.get(key[, value]) :

```
print(dic)
```

```
{0: ' Data', 1: 'GREAT', 2: 'LEARNING', 3: 'Python', 4: 'Happy'}
```

```
dic.get(1)
```

```
'GREAT'
```

2.E) dict.items()

```
dic.items()
```

```
dict_items([(0, ' Data'), (1, 'GREAT'), (2, 'LEARNING'), (3, 'Python'), (4, 'Happy')])
```

2.F) dict.keys() :

```
dic.keys()
```

```
dict_keys([0, 1, 2, 3, 4])
```

2.G) dict.setdefault():

```
dic.setdefault(5 , 'End')
```

```
'End'
```

```
print(dic)
```

```
{0: ' Data', 1: 'GREAT', 2: 'LEARNING', 3: 'Python', 4: 'Happy', 5: 'End'}
```

2.H) dict.update():

```
dic.update({1:"A"})
```

```
print(dic)
```

```
{0: ' Data', 1: 'A', 2: 'LEARNING', 3: 'Python', 4: 'Happy', 5: 'End'}
```

2.L) dict.values():

```
dic.values()
```

```
dict_values([' Data', 'A', 'LEARNING', 'Python', 'Happy', 'End'])
```

Conditional Statements

```
1. IF
2. IF-ELSE
3. If...Elif...Else
```

1. Write if statement 13 is greater than 25?

```
no1=13
if no1>25 :
  print('13 is greater')
if 25>no1 :
  print('25 is greater')
```

```
25 is greater
```

2. Write a if else statement to find if the number is divisible by 25?

```
number=50
if number % 25 == 0:
    print(f"The number {number} is divisible by 25.")
else:
    print(f"The number {number} is not divisible by 25.")
```

```
The number 50 is divisible by 25.
```

```
number=95
if number % 25 == 0:
    print(f"The number {number} is divisible by 25.")
else:
    print(f"The number {number} is not divisible by 25.")
```

```
The number 95 is not divisible by 25.
```

3.Using the three variables 'a = 154; b = 2451; c = 6054',

Write a If...Elif...Else statement to find the greatest number

```
a = 154
b = 2451
c = 6054

if a >= b and a >= c:
  greatest_number = a
elif b >= a and b >= c:
  greatest_number = b
else:
  greatest_number = c

print('the greatest number is', greatest_number)
```

```
the greatest number is 6054
```

while Loop

1. Write a code to print (1,10) using while loops

```
count = 1
while count <= 10:
  print(count)
  count += 1
```

```
1
2
3
4
5
6
7
8
9
10
```

For Loop:

1. Write a code to print string using for loop?

```
my_string = "Python"

for char in my_string:
  print(char)
```

```
P
y
t
h
o
n
```

2. Write a code for (1,10) after equal to 3 break the loop?

```
for number in range(1, 11):
  print(number)

  if number == 3:
    break
```

```
1
2
3
```

3. Write a code for (1,10) after equal to 3 continue the loop?

```
for number in range(1, 11):
  if number <= 3:
    continue
  print(number)
```

```
4
5
6
7
8
9
10
```

```
for number in range(1, 11):
  if number == 3:
    continue
  print(number)
```

```
1
2
4
5
6
7
8
9
10
```