```
#Range Function/Statement
a = list(range(1,10))
print(a)                        # Output => [1, 2, 3, 4, 5, 6, 7, 8, 9]
print(list(range(1,10)))        # Output => [1, 2, 3, 4, 5, 6, 7, 8, 9]
''' It prints range between Initial to End (excluding End)
means print(list(range(1,10))) prints range between 1 to 9 It doesn't
include last element. '''

b = list(range(1,12,2))         # It will print range between 1 to 11
with 2 jump.
print(b)                        # Output => [1, 3, 5, 7, 9, 11]

c = list(range(3,13,3))
print(c)                        # Output => [3, 6, 9, 12]
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9]
[1, 2, 3, 4, 5, 6, 7, 8, 9]
[1, 3, 5, 7, 9, 11]
[3, 6, 9, 12]
```

```
# Basic questions related to range function.
#! Question: Using range display/print the list of even numbers till
10.
''' Solution:  '''
even = list(range(2,11,2))
print(even)
# Output => [2, 4, 6, 8, 10]

''' Explanation:
We have to print even numbers upto 10 so
1. We have to take range between 2 to 11.
   Because If we take 10, range function will exclude 10. While we
have to print 10.
2. Use 2 jump, because if we start counting from an even number
   every 2nd number will be an even number.'''

print(even)
```

```
[2, 4, 6, 8, 10]
[2, 4, 6, 8, 10]
```

```
#! Question: Print the list of 1st 10 natural numbers in reverse using
range.
''' Solution: '''
revNumber = list(range(10,0,-1))
print(revNumber)

# Output => [10, 9, 8, 7, 6, 5, 4, 3, 2, 1]
```

```
''' Explanation:
=> We have to take range in reverse manner, so range will start from
10 to 0.
=> zero(0) will be excluded by default, so It will print list upto 1.
=> We have to use reverse jump using -1 to print reverse order of the
list.'''

[10, 9, 8, 7, 6, 5, 4, 3, 2, 1]

# LOOP
#! for Loop => It is an iterative process.
for x in range(1,11):
    print(x)
# Output =>
'''
1
2
3
4
5
6
7
8
9
10
'''


1
2
3
4
5
6
7
8
9
10

''' Python Indentation =>
    1. Indentation refers to the spaces at the beginning of a code
line.
    2. Python uses indentation to indicate a block of code. '''
# 1 Indent => 1 Tab or 4 Spaces

#List Comprehensions => The operations performed inside a list.
#! Question: Print the first 10 natural numbers using list
comprehensions.
# Solution:
a = [x for x in range(1,11)]   # List Comprehensions
```

```
print(a)                        # Output => [1, 2, 3, 4, 5, 6, 7, 8, 9,
10]
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

```
b = ['Apple','Banana','Dragon
Fruit','Guava','Grapes','Mango','Pineapple']
print(b)
print(len(b))          # Output => 7

#copy => Alternative of copy function using list comprehension.
a = [x for x in b]
print(a)               # Output => ['Apple', 'Banana', 'Dragon Fruit',
'Guava', 'Grapes', 'Mango', 'Pineapple']

#Filtering list
c = [x for x in b if 'G' in x]
print(c)               # Output => ['Guava', 'Grapes']
c = [x for x in b if 'le' in x]
print(c)               # Output => ['Apple', 'Pineapple']
c = [x for x in b if 'a' in x]
print(c)               # Output => ['Banana', 'Dragon Fruit', 'Guava',
'Grapes', 'Mango', 'Pineapple']
c = [x for x in b if 'a' not in x]
print(c)               # Output => ['Apple']
c = [x for x in b if 'A' in x]
print(c)               # Output => ['Apple']
c = [x for x in b if 'pp' in x]
print(c)               # Output => ['Apple', 'Pineapple']
c = [x for x in b if 'G' not in x]
print(c)               # Output =>['Apple', 'Banana', 'Dragon Fruit',
'Mango', 'Pineapple']
```

```
['Apple', 'Banana', 'Dragon Fruit', 'Guava', 'Grapes', 'Mango',
'Pineapple']
7
['Apple', 'Banana', 'Dragon Fruit', 'Guava', 'Grapes', 'Mango',
'Pineapple']
['Guava', 'Grapes']
['Apple', 'Pineapple']
['Banana', 'Dragon Fruit', 'Guava', 'Grapes', 'Mango', 'Pineapple']
['Apple']
['Apple']
['Apple', 'Pineapple']
['Apple', 'Banana', 'Dragon Fruit', 'Mango', 'Pineapple']
```

```
#Creating a list comprehension from scratch.
print(b)
a = []
for x in b:
    if 'pp' in x:
```

```
        a.append(x)
print(a)

# Output => ['Apple', 'Pineapple']
```

```
['Apple', 'Banana', 'Dragon Fruit', 'Guava', 'Grapes', 'Mango',
'Pineapple']
['Apple', 'Pineapple']
```

```
# Question :
''' lst = [1, 2, 3, 4, 5, 6],
    Between 3 & 4 insert 3 more numbers - 97,98,99 without using the
insert function.
    Expected Output => [1, 2, 3, 97, 98, 99, 4, 5, 6]. '''

#Solution:
lst = [1, 2, 3, 4, 5, 6]
num = [97, 98, 99]
ans = lst[:3] + num + lst[3:]
print(ans)

# Explanation:
'''
1. lst[:3] => [1, 2, 3], It will exclude 3rd index i.e 4
2. num = [97, 98, 99]
3. lst[3:] => from 3rd index to the last. => [4, 5, 6]
'''
print(lst[:3])
print(lst[3:])
```

```
[1, 2, 3, 97, 98, 99, 4, 5, 6]
[1, 2, 3]
[4, 5, 6]
```

```
#del [a:b] => Deletion through slicing.
a = [1, 2, 3, 97, 98, 99, 4, 5]
del a[3:6]
print(a)
# It will delete elements from 3rd index to 5th index,
# Because 6th index will be excluded from deletion. => Output => [1,
2, 3, 4, 5]
```

```
[1, 2, 3, 4, 5]
```