len() and Range() builtin function

In [ ]:
```
len()--> gives you the length of the sequence(list , string , tuple.set)
```

In [3]:
```
x="Manishakti  Python Good"
len(x)
```

Out[3]:
```
23
```

In [4]:
```
x=[10,20,30,40,50]
len(x)
```

Out[4]:
```
5
```

In [5]:
```
x=(10,20,30,40,50,60,70)
len(x)
```

Out[5]:
```
7
```

In [6]:
```
x={1,2,3,4,5}
len(x)
```

Out[6]:
```
5
```

In [ ]:
```
Range()-->is used to represnt a  sequence of  a number. Range is immutable
Range funcation will take three arguments:
First argument --> starting value --> default value -0
Second argument --> ending value+1
third argument --> Gap -> default -1

Syntax :
    range(start value , end value , gap)
```

In [18]:
```
#Variations of Range function
#Example1:
x=list(range(11)) #0,1,2,3,4,5,6,7,8,9,10
x
```

Out[18]:
```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

In [21]:
```
#Example2:
x=list(range(10,21,2)) #Generate numebr from 10-20
x
```

Out[21]:
```
[10, 12, 14, 16, 18, 20]
```

In [24]:
```
#example3:
x=list(range(0,20,3)) #0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19
# 0 2 4 6 8 10 12 14 16 18
#0,3,6,9,12
x
```

Out[24]:
```
[0, 3, 6, 9, 12, 15, 18]
```

What are Loops?

In [25]:
```
print("Hello World")
print("Hello World")
print("Hello World")
print("Hello World")
print("Hello World")
print("Hello World")
print("Hello World")
print("Hello World")
print("Hello World")
print("Hello World")
print("Hello World")
print("Hello World")
print("Hello World")
print("Hello World")
print("Hello World")
print("Hello World")
print("Hello World")
print("Hello World")
print("Hello World")
```

```
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
```

In [ ]:
```
Loops: When we want to execute a group of statement multiple time then we should always use loops
    Example: if we want to print hello world 100 times for that we simply use a for loop it will do
        it in just 3 lines
```

Types of Loops?

In [ ]:
```
In python we have two types of loops
1:For loop
2:while loops
```

In [ ]:
```
#For loop:if we want to execute a number of statements in some sequence then we should always
use for loop
Syntax of For loop :
    for i in sequence:
        body
i is known as iterator
```

In [33]:
```
x=[10,20,30,40,50,60,50,60]
for j in range(2,len(x)):
    print("Element present at index "+str(j)+" and the element is "+str(x[j]))
#J is nothing but known as iterator
```

```
Element present at index 2 and the element is 30
Element present at index 3 and the element is 40
Element present at index 4 and the element is 50
Element present at index 5 and the element is 60
Element present at index 6 and the element is 50
Element present at index 7 and the element is 60
```

In [34]:
```
x=[10,20,30,40,50,60,50,60]
for j in x:
    print("Element present at index "+str(j)+" and the element is "+str(j))
#J is nothing but known as iterator
```

```
Element present at index 10 and the element is 10
Element present at index 20 and the element is 20
Element present at index 30 and the element is 30
Element present at index 40 and the element is 40
Element present at index 50 and the element is 50
Element present at index 60 and the element is 60
Element present at index 50 and the element is 50
Element present at index 60 and the element is 60
```

In [ ]:
```
range(len(x))
range(5)#0 to 4
```

In [35]:
```
#Python program to print hello world 10 times:
for i in range(11):
    print("Hello world")
```

```
Hello world
Hello world
Hello world
Hello world
Hello world
Hello world
Hello world
Hello world
Hello world
Hello world
Hello world
```

In [36]:
```
#Python program to print even numebrs between 0 to  30
for i in range(0,31,2):
    print(i)
```

```
0
2
4
6
8
10
12
14
16
18
20
22
24
26
28
30
```

In [38]:
```
#Python program to print odd numebrs between 0 to  30
for i in range(1,30,2):#1,3,5,7,9,10
    print(i)
```

```
1
3
5
7
9
11
13
15
17
19
21
23
25
27
29
```

In [41]:
```
x="Hello I am a Python Developer"
for i in x:
    print(i,end="")
```

```
Hello I am a Python Developer
```

In [42]:
```
x="Hello I am a Python Developer"
for j in range(len(x)):
    print("Element present at index "+str(j)+" and the element is "+str(x[j]))
```

```
Element present at index 0 and the element is H
Element present at index 1 and the element is e
Element present at index 2 and the element is l
Element present at index 3 and the element is l
Element present at index 4 and the element is o
Element present at index 5 and the element is 
Element present at index 6 and the element is I
Element present at index 7 and the element is 
Element present at index 8 and the element is a
Element present at index 9 and the element is m
Element present at index 10 and the element is 
Element present at index 11 and the element is a
Element present at index 12 and the element is 
Element present at index 13 and the element is P
Element present at index 14 and the element is y
Element present at index 15 and the element is t
Element present at index 16 and the element is h
Element present at index 17 and the element is o
Element present at index 18 and the element is n
Element present at index 19 and the element is 
Element present at index 20 and the element is D
Element present at index 21 and the element is e
Element present at index 22 and the element is v
Element present at index 23 and the element is e
Element present at index 24 and the element is l
Element present at index 25 and the element is o
Element present at index 26 and the element is p
Element present at index 27 and the element is e
Element present at index 28 and the element is r
```

In [43]:
```
for i in range(4,41,4):
    print(i)
```

```
4
8
12
16
20
24
28
32
36
40
```

In [ ]:
```
#Factorial program
5!= 5*4*3*2*1=120
6!=6*5*4*3*2*1
```

In [56]:
```
#5!= 5*4*3*2*1*0->
n=int(input())
result=1

for i in range(1,n+1):
    result=result*i  #result=1   #result=2 ,  result 6  result 24  result 120
print(result)
```

```
120
6689502913449127057588118054090372586752746333138029810295671352301633557244962989366874165271984981308157637893214090552534408589408121859898481114389650005964960521256960000000000000000000000000000000
```

In [55]:
```

```

Nested Loops Concepts

In [ ]:
```
loop inside loop
for inside for
```

In [60]:
```
for i in range(3):
    for  j in range(3):
        print("I value is "+str(i)+" j value is "+str(j))
```

```
I value is 0 j value is 0
I value is 0 j value is 1
I value is 0 j value is 2
I value is 1 j value is 0
I value is 1 j value is 1
I value is 1 j value is 2
I value is 2 j value is 0
I value is 2 j value is 1
I value is 2 j value is 2
```

In [ ]:
```
Nested for loops basically used for patterns
```

In [ ]:
```
#Armstrong NUMBER:
153 --> 1**3+5**3+3**3 -->27+125+1 =153
```

In [82]:
```
x="153453"
for i in range(len(x)):
    print(int(x[i])%10)
```

```
1
5
3
4
5
3
```

In [ ]:
```
n=5
*****
*****
*****
*****
*****
```

In [81]:
```
#Square pattern
for i in range(5):
    for j in range(5):
        print("#",end="")
    print()
```

```
#####
#####
#####
#####
#####
```

In [83]:
```

```

```
WARNING: Ignoring invalid distribution -ecorator (c:\users\praty\anaconda3\lib\site-packages)
WARNING: Ignoring invalid distribution -ecorator (c:\users\praty\anaconda3\lib\site-packages)
ERROR: Could not find a version that satisfies the requirement opencv (from versions: none)
ERROR: No matching distribution found for opencv
WARNING: Ignoring invalid distribution -ecorator (c:\users\praty\anaconda3\lib\site-packages)
WARNING: Ignoring invalid distribution -ecorator (c:\users\praty\anaconda3\lib\site-packages)
WARNING: Ignoring invalid distribution -ecorator (c:\users\praty\anaconda3\lib\site-packages)
```

In [ ]:
```

```