for loop :-

If we want to execute some action for every element present in some sequence (it may be string or collection) then we should go for for loop.

Syntax :-

```
for x in sequence:
    body
```

Ex :-

```
d = "dhapps"
for i in d:
    print(i)
```

Output :- (Try)

Ex :- for i in {1,2,3,4}:

```
print("hello world")
```

o/p :-
hello world
hello world
hello world
hello world

Ex :- for i in range(0, 10):

```
print(i)
```

o/p :-
0
1
2
3
4
5
6
7
8
9

Eg:
```
for i in range(0, 10, 2):
    print(i)
```

o/p:- 0
2
4
6
8

Eg:
```
for i in range(10, -1):
    print(i)
```

o/p:- 10
9
8
7
6
5
4
3
2
1

Sum of n numbers program:

using loop -

```
no = int(input("Enter a number"))
sum = 0
for i in range(0, no+1):
    sum = sum + i
print(sum)
```

o/p: enter a numbers
15

using while loop :-

```
n=int(input("enter a no"))
i=0
sum=0
while i<=n:
    sum = sum+i
    i=i+1
print(sum)
```

o/p :- enter a no5
15

Nested loop -

Ex :-
```
for i in range(2):
    print("im outer loop")
    for j in range(2):
        print("inner loop")
```

o/p :- Im outer loop
inner loop
inner loop
Im outer loop
inner loop
inner loop

Ex :-
```
i=0
while i<2:
    print("hi")
    j=0
    while j<2:
        print("hello")
        j=j+1
    i=i+1
```

o/p :-
hi
hello
hello
hi
hello
hello

Transfer Statements:-

python supports 2 types of Transfer statements.

- break
- Continue.

break:

We can use break statement inside loops to break loop execution based on some condition.

Ex:-

```
for i in range(10):
    if i==7:
        print("enough lets take break")
        break
    print(i)
```

O/p:
0
1
2
3
4
5
6
enough lets take break.

Continue :

It skips the current Execution

Ex:- for i in range(10):

```
    if i==7:
        print("lets skip")
        Continue
    print(i)
```

o/p:- 0
1
2
3
4
5
6
lets skip
8
9

Task: prime number program.