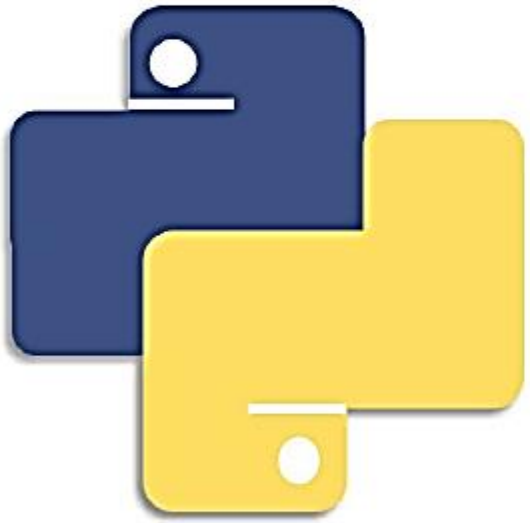
Python

By

Dr. rabih ali sbera

Department of Computer Engineering Techniques-2 nd

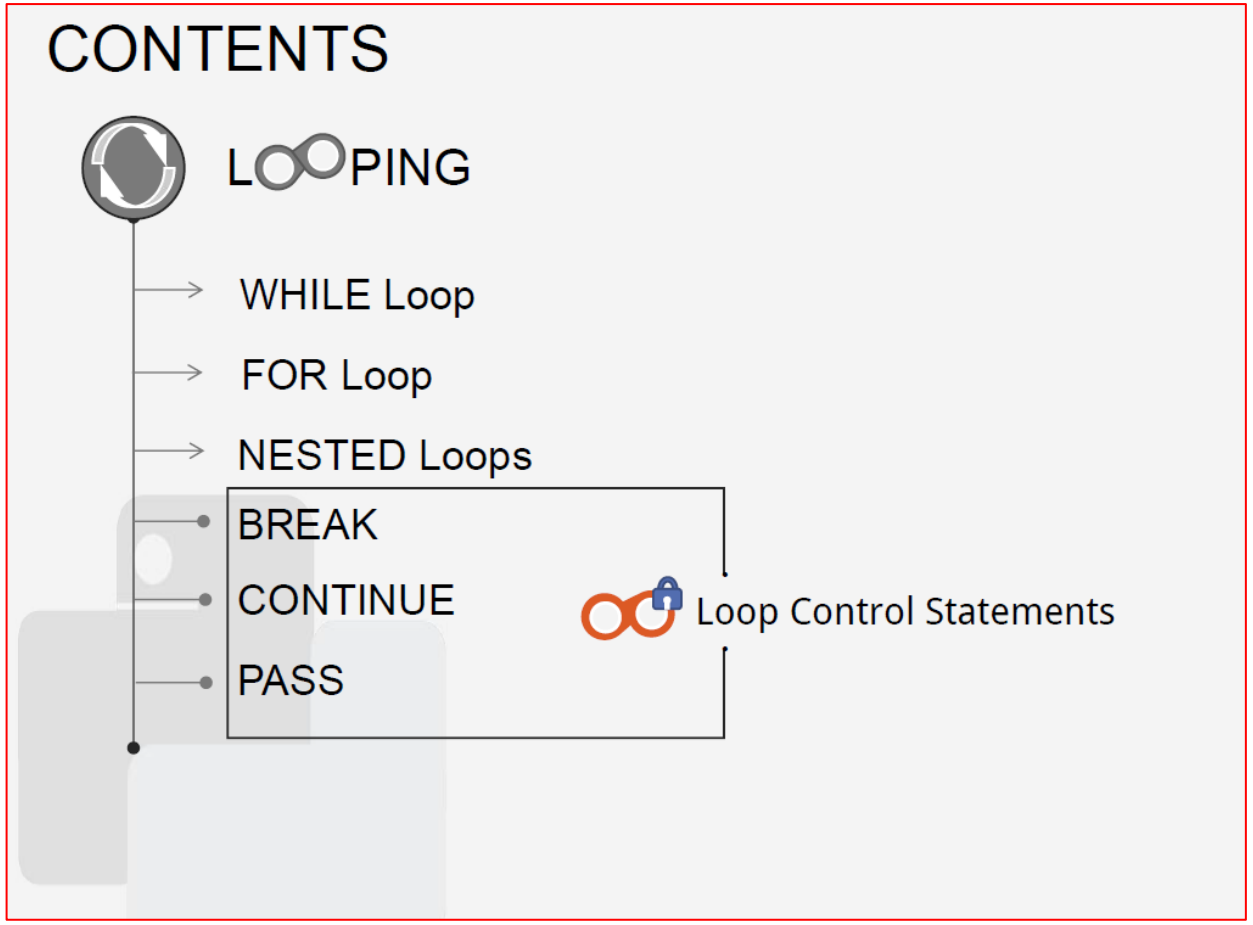
2024-2025



Python Programming

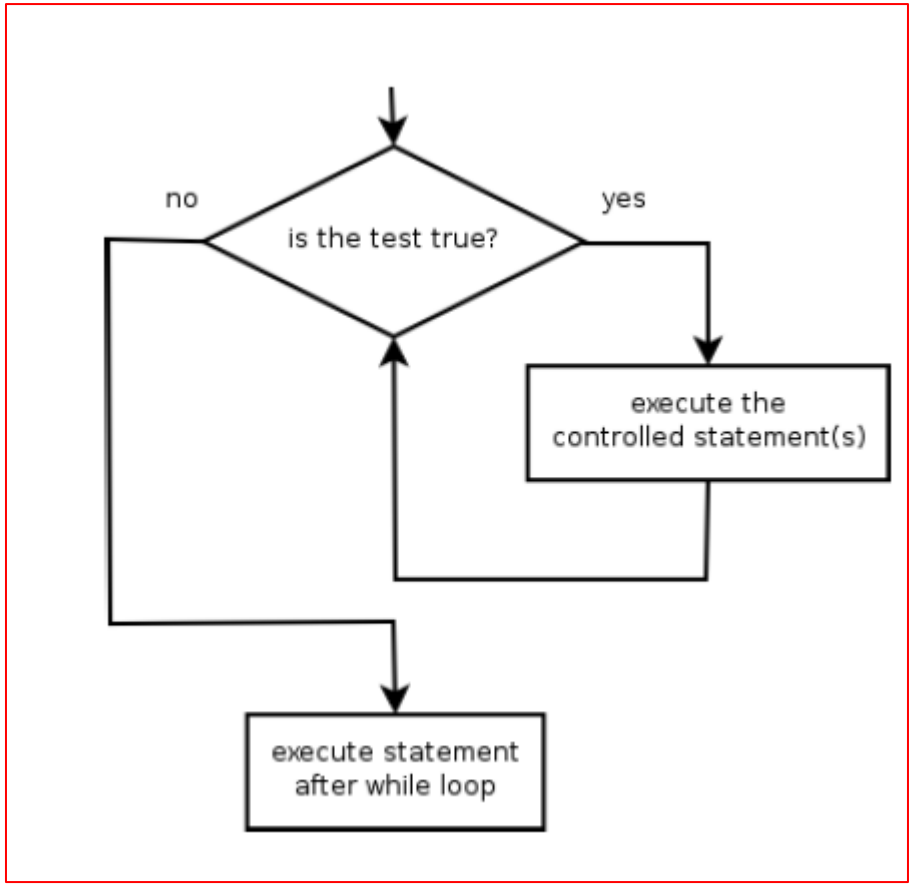
Looping & It's Control

loops


CONTENTS
LOOPING
WHILE Loop
FOR Loop
NESTED Loops
BREAK
CONTINUE
PASS
Loop Control Statements

While loop

- A while loop statement in Python programming language repeated executes a target statement as long as a given condition is true. A uniform indent is mandatory for rest of the statements to be in a loop.



Syntax:

```
while condition:
statements
```

While loop

Example: Printing Factorial of a number

```
num = int(input("Enter a number to calc it's factorial:"))
temp=1
fact=1
while temp <= num:
 fact*=temp
 temp=temp+1
 print("Factorial of", num, "is :",fact)
```

Output:

```
Enter a number to calc it's factorial:5
Factorial of 5 is : 1
Factorial of 5 is : 2
Factorial of 5 is : 6
Factorial of 5 is : 24
Factorial of 5 is : 120
```

While loop

Example:

```
while number < 200:
        print (number)
        number = number * 2
```

Output:

```
2
4
8
16
32
64
128
```

While loop

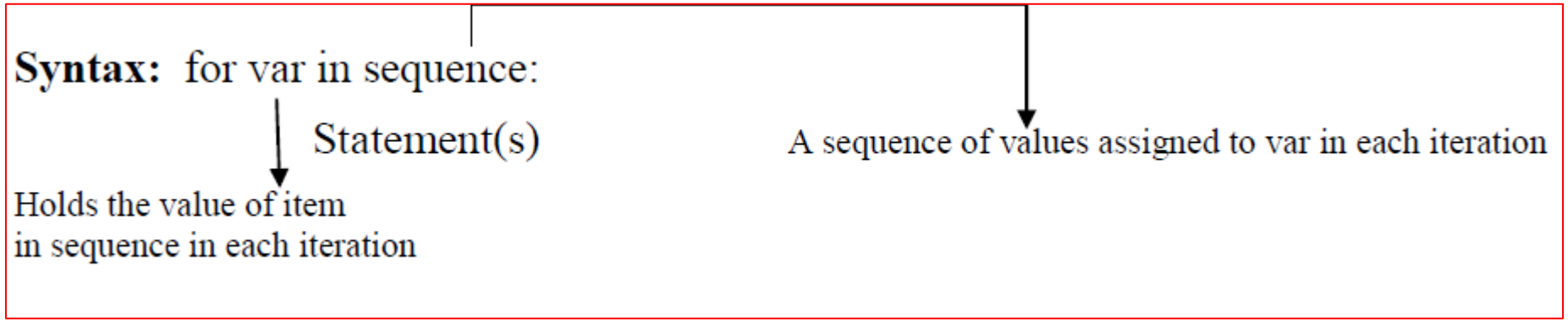
```
n=1
while n<=10:
    if(n%2!=0):
        print(">", n)
      n=n+1
  else:
  print("Bye")
```

Output:

```
> 1
bye
> 3
bye
> 5
bye
> 7
bye
> 9
bye
```

For loop

Python **for loop** is used for repeated execution of a group of statements for the desired number of times. It iterates over the items of lists, tuples, strings, the dictionaries and other iterable objects;

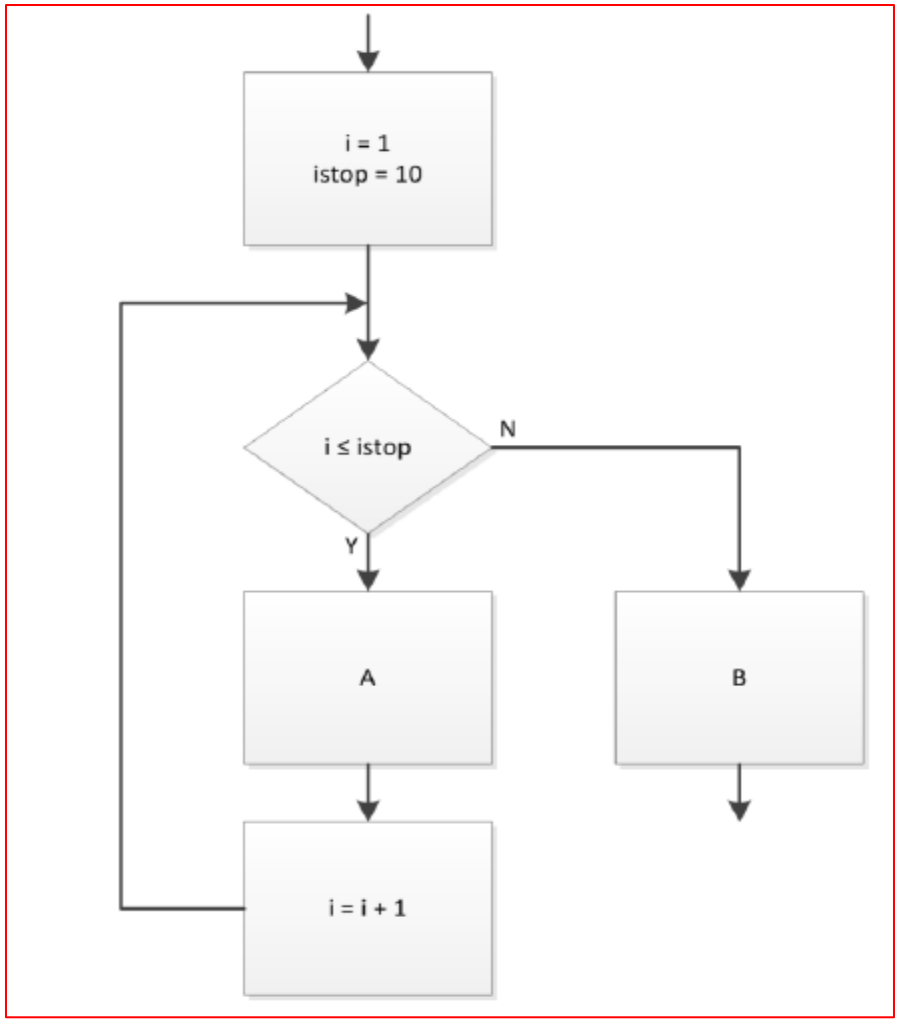


For loop

for loops are **counted loops**

Number of loop iterations is known and is constant

Here loop executes 10 times Stopping value not necessarily hard-coded Could depend on an input or vector size, etc.



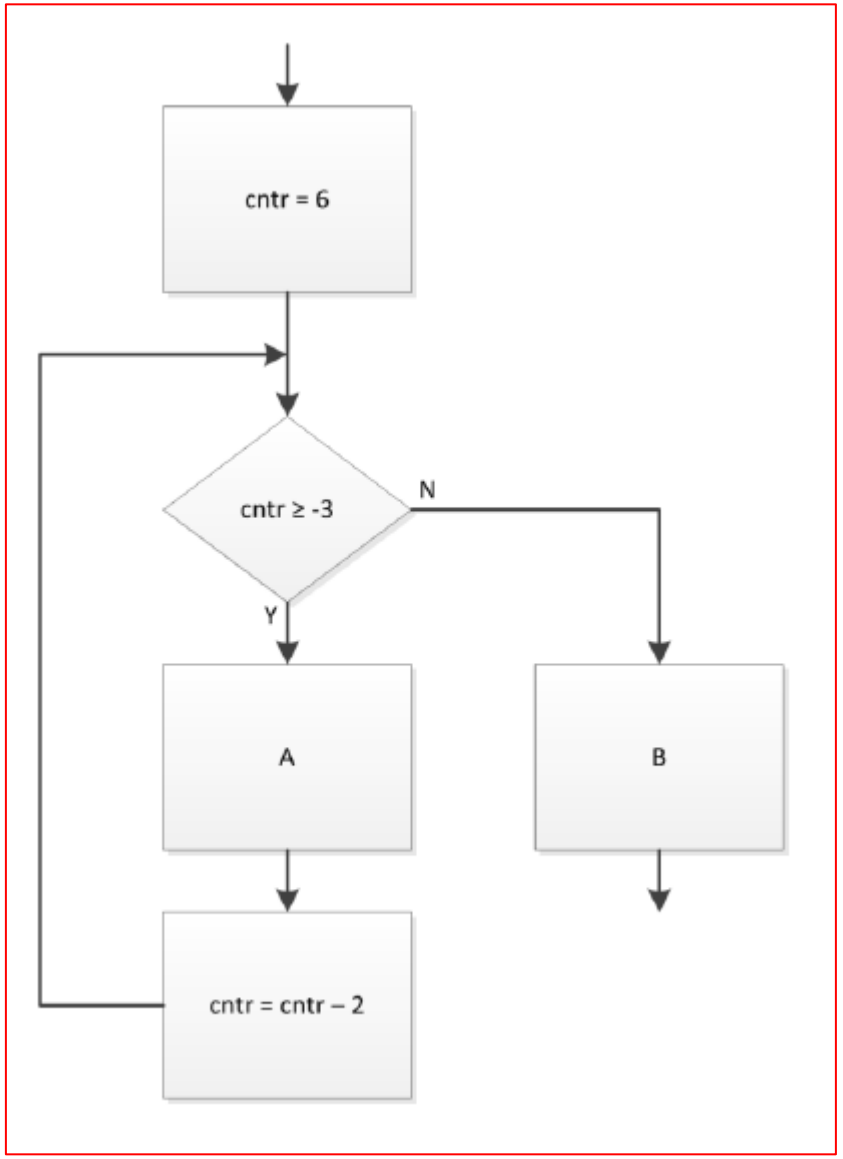
For loop

Loop counter may start at value other than 1

Increment size may be a value other than 1

Loop counter may count backwards

Iteration	cntr	Process
1	6	A
2	4	A
3	2	A
4	0	A
5	-2	A
6	-4	B



For loop

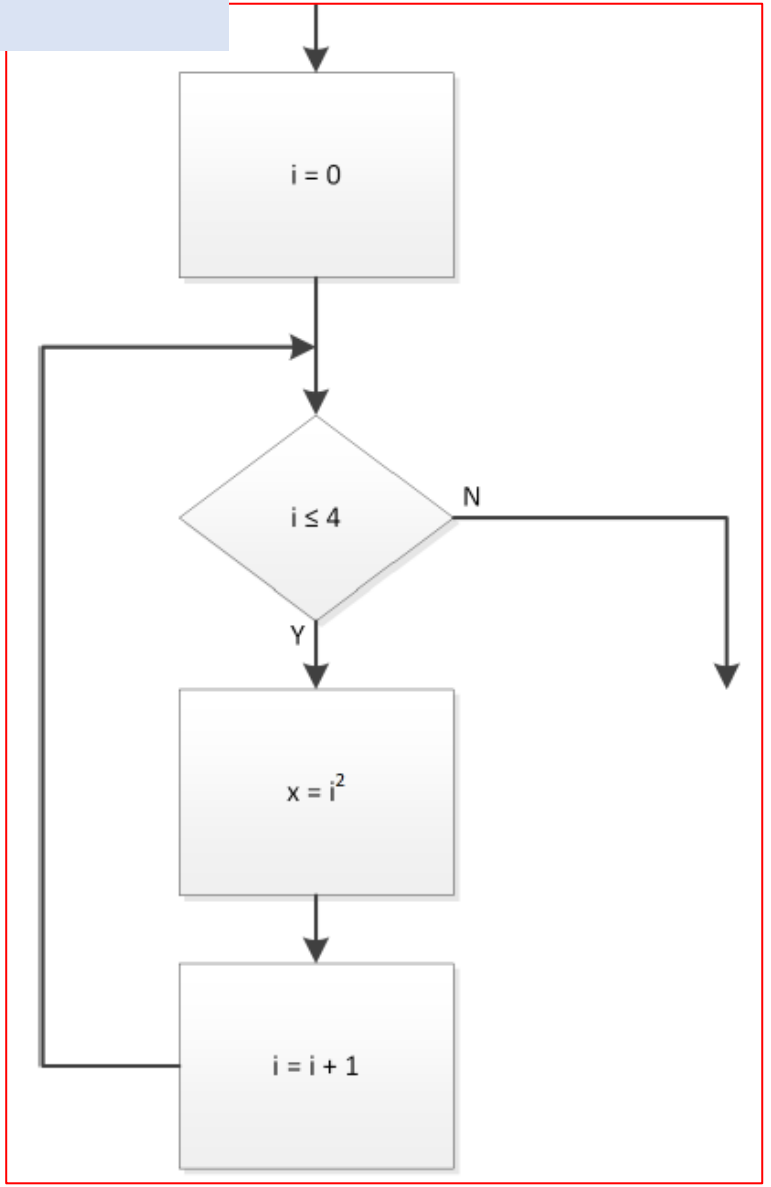
- for Loop – Example 1

Here, the loop counter, i, is used to update a variable, x, on each iteration

Iteration	i	x
1	0	0
2	1	1
3	2	4
4	3	9
5	4	16

When loop terminates, and flow proceeds to the next process step, x = 16

A scalar No record of previous values of x



For loop

- for Loop – Example 2

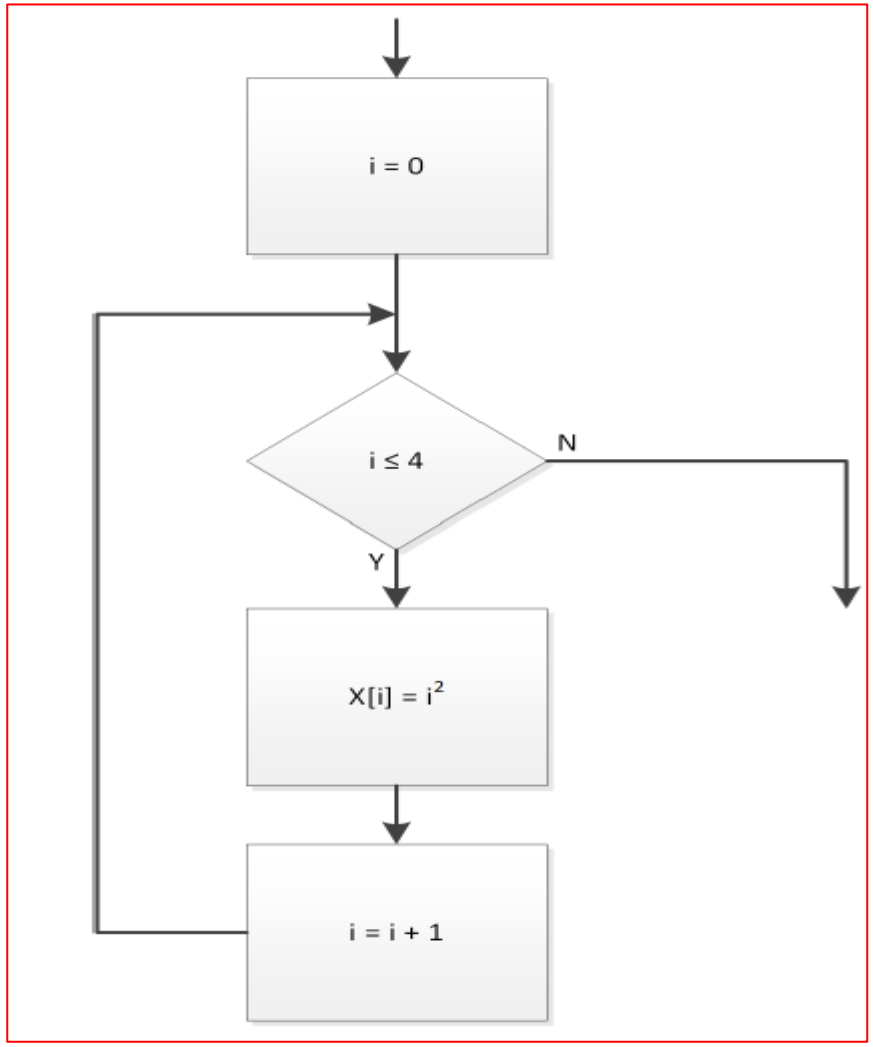
Now, modify the loop process to store values of x as a ***vector***

Use loop counter to index the vector

i	X[i]	x
0	0	[0]
1	1	[0, 1]
2	4	[0, 1, 4]
3	9	[0, 1, 4, 9]
4	16	[0, 1, 4, 9, 16]

When loop terminates, x = [0, 1, 4, 9, 16]

A ***vector*** x grows with each iteration



For loop

```
numbers = [1, 2, 4, 6, 11, 20]
seq=0
for val in numbers:
  seq=val*val
  print(seq)
```

```
1
4
16
36
121
400
```

```
s="“
for i in range(5):
  s+= "*"
  print(s)
else:
  print("Goodbye")
```

```
*
**
***
****
*****
Goodbye
```

```
for i in range(5):
  s+= "*"
  print(s)
  print("Goodbye")
```

```
******
Goodbye
*******
Goodbye
********
Goodbye
*********
Goodbye
**********
Goodbye
```

For loop

- **Iterating over a list:**

```
#list of items
list = ['M','R','C','E','T']
i = 1
#Iterating over the list
for item in list:
    print ('college ',i,' is ',item)
    i = i+1
```

```
college  1  is  M
college  2  is  R
college  3  is  C
college  4  is  E
college  5  is  T
```

Iterating over a Tuple:

```
tuple = (2,3,5,7)
print ('These are the first four prime numbers ')
#Iterating over the tuple
for a in tuple:
  print (a)
```

```
These are the first four prime numbers
2
3
5
7
```

For loop

```
#creating a dictionary
college = {"ces":"block1","it":"block2","ece":"block3"}
#Iterating over the dictionary to print keys
print ('Keys are:')
for keys in college:
  print (keys)
#Iterating over the dictionary to print values
print ('Values are:')
for blocks in college.values():
  print(blocks)
```

```
Keys are:
ces
it
ece
Values are:
block1
block2
block3
```

Iterating over a String:

```
#declare a string to iterate over
college = 'MRCET'
#Iterating over the string
for name in college:
  print (name)
```

```
M
R
C
E
T
```

Nested For loop

- When one Loop defined within another Loop is called Nested Loops.

Syntax:
for val in sequence:
for val in sequence:

```
for i in range(1,6):
  for j in range(0,i):
   print(i, end=" ")
   print('')
```

```
1
2 2
3 3 3
4 4 4 4
5 5 5 5 5
```

```
for i in range(1,6):
  for j in range(0,i):
   print(i, end=" ")
  print('')
```

```
1
2 2
3 3 3
4 4 4 4
5 5 5 5 5
```

```
for i in range(1,6):
 for j in range(5,i-1,-1):
  print(i, end=" ")
 print('')
```

```
1 1 1 1 1
2 2 2 2
3 3 3
4 4
5
```