* I/P - t = (1, 2, 3, 4)

```
for i in tup:
    Print(i)
```

O/P - 1
2
3
4

* I/P - t = (1, 2, 3, 4)

```
for i in range(len(tup)):
    Print(i)
```

O/P - 0
1
2
3

* I/P - t = (1, 2, 3, 4, 5)

```
sum = 0
for item in tup:
    sum += item
Print('product = {}'.format(prod))
```

O/P - product = 120.

```
* I/P - lst = [10, 20, 30, 40, 50, 60]
       for index in range(len(lst)):
           Print(lst[index])

O/P - 10
      20
      30
      40
      50
      60.

* I/P - print('LIST ITERATION')
       lst = [10, 20, 30, 40, 50, 60]
       for element in lst:
           Print(element, end = ' ')
       Print()

       Print('TUPLE ITERATION')
       tup = (10, 20, 30, 40, 50, 60)
       for element in tup:
           Print(element, end = ' ')
       Print()

       Print('STRING ITERATION')
       mystr = 'python'
       for element in mystr:
           Print(element, end = ' ')
       Print()

       Print('DICTIONARY ITERATION')
```

```
dict = { }
dict ['xyz'] = 123
dict ['abc'] = 345
for key in dict:
    Print(key, dict[key])
```

O/P -

```
LIST ITERATION:
10 20 30 40 50 60
TUPLE ITERATION:
10 20 30 40 50 60
STRING ITERATION:
P y t h o n
DICTIONARY ITERATION:
xyz = 123
abc = 345
```

FOR-ELSE:

The else keyword in a for loop specifies a block of code to be executed when the loop is finished.

* I/P -

```
for x in range(6):
    Print(x)
else:
    Print("FINISHED")
```

o/p - 0
1
2
3
4
5
FINISHED

* i/p - tup = (1, 2, 3, 4, 5)

```
Prod = 1
for item in tup:
    Prod *= item
else:
    Print('You have exhausted the list')
Print('product = {}'.format(prod))
```

o/p - You have exhausted the list.

Product = 120.

NESTED LOOP:

A nested loop is a loop inside a loop. the "inner loop" will be executed one time for each iteration of the "outer loop".

*I/p - adj = ["red", "big", "tasty"]

fruits = ["apple", "banana", "cherry"]

```
for x in adj:
    for y in fruits:
        Print(x, y)
```

o/p - red apple
red banana
red cherry
big apple
big banana
big cherry
tasty apple
tasty banana
tasty cherry.

BREAK:

The break keyword is used to break out of "a for loop or a while loop".

NOTE: Break statements are only allowed inside loops.

*I/P - i = 1
while i < 9:
    Print(i)
    if (i == 3):
        break
    i += 1

O/P - 1
2
3

CONTINUE:

The continue keyword is used to end the current iteration in a 'for' loop or in a 'while' loop", and continues to the next iteration.

*I/P - i = 0
while i < 9;
    i += 1
    if i == 3:
        continue
    Print(i)

O/P - 1
2
4
5
6
7
8
9.

*I/p - While True:

num = int(input(" Entere a number to check if its even or odd:

if num % 2 == 0:

Print ("yes, you ~~printed~~ entered an even no.")

else:

break

Print ("wrong no., End")

O/p - Entere a number to check if its even or odd
2

Yes, you entered an even number.

Entere a number to check if its even or odd

wrong number, End.

*I/p - lst = [1, 2, 3, 4, 5, 1, 2, 3, 4]

ans = 0

for elements in lst:

ans = ans ^ elements

Print (ans)

O/p - 5

```
* I/P -  if(number_of_div ==12):
            Print("Prime")
        else:
            Print("Not a prime")
```

O/P - Not a prime.

```
* I/P -  num = 9
        number_of_div = 0
        for i in range(1, num+1):
            if(num % i ==0):
                number_of_div = number_of_div+1
        Print(number_of_div)
```

O/P - 3.

WHY ELSE:

```
* I/P -  lst=[1, 2, 3, 4, 5]
        for item in lst:
            if item == 4:
                break
            Print(item)
        else:
            Print('HI')
```

O/P - 1
2
3