* Loops in python :-

Loops are used to repeat (or) iterate a block of code multiple times untill a given condition is satisfied.

types of loops :-

1) for loop :-

* used when we know the number of iterations in advance.

* It is often used to iterate over a sequence (list, tuple, string, range, etc).

Syntax:-

```
for variable in sequence:
    # code block.
    statement
```

var = iterative variable.

Ex:- ① value = "Besant Tech"
print (value)

o/p:- Besant Tech.

② value = "Besant Tech"
for a in value:
print (a)

o/p:-
B
e
s
a
n
t

T
e
c
h

* for loop with range:-

The range() function generates a sequence of numbers, which can be used in a for loop iteration.

Syntax:-

```
for variable in range (start, stop, step)
    # code block.
```

var < stop value - The last value -1

- start → (optional) starting num (default = 0)
- stop → ending number (not included)
- step → (optional) increment / decrement (default = 1).

Ex:-

1) simple range (only stop value)

```
for i in range (5):
    print (i)
```

o/p:- 0
1
2
3
4

2) Range with start and stop

```
for i in range (2,6):
    print (i)
```

o/p:- 2
3
4
5

3) Range with start, stop, step.

```
for i in range (1, 10, 2):
    print (i)
```

o/p:- 1
3
5
7
9

4) Range with negative step (counting backwards).

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

o/p:- 10
8
6
4
2

Ex:- * for i in range (0,10,1):
print ("Besant Tech")

i = 0, 1, 2, 3, ... 9

o/p:- Besant Tech
Besant Tech
.
.
.
.
.
Besant Tech.

* value = "Besant Tech"
print (value [0])

o/p :- B.

Besant Tech
0 1 2 3 4 5 6 7 8 9 10

* value = "Besant Tech"
for i in range (0, len(value), 1):
print (value [i])

The end value is given or not it takes default '1'.

o/p:- B
e
s
a
n
t
T
e
h.

* animals = ['monkey', 'lion', 'dog', 'cat', 'rabit', 'cow']
print (animals).

o/p:- ['monkey', 'Lion', 'dog', 'cat', 'rabit', 'cow']

* for i in animals:
print (i)

o/p:- monkey
lion
dog
cat
rabit
cow

* to add the string in given variable.
for i in animals:
print ("hi good afternoon", i)

o/p:- hi good afternoon monkey
" " : "
" " " "
" " " "
" " " "
lion
dog
cat
rabbit

* to print the value position with range in given i/p

```
value = "Besant Tech"
for i in range (0, len (value), 1):
    print (i, value [i])
```

o/p:-
```
0 B
1 e
2 s
3 a
4 n
5 t
6
7 T
8 e
9 c
10 h.
```

* `print (f"postion = {i}, value = {value[i]}")`

o/p:-
```
position =0, value =B
position =1, value =e.
position =2, value =S
position =3, value =a
position =4, value =n
position =5, value =t
position =6, value =
position =7, value =T
position =8, value =e
position =9, value =c
position =10, value =h.
```

* `value = "Besant Tech"`

```
for i in enumerate (value):
    print (i)
    print (f"position = {i[0]}, value = {i[1]}")
```

o/p:-
```
(0, 'B')
position =0, value =B.
(1, 'e')
position =1, value =e.
  :
  :
(10, 'h')
position =10, value =h.
```

* even num.

```
for i in range (2, 21, 2):
    print (i)
```

o/p:-
```
2
4
6
8
10
12
14
16
18
20
```

* even or odd number.

```
for i in range (1,11,1):
    if (i%2 == 0):
        print (f"{i} = even number")
    else:
        print (f"{i} = odd number")
```

o/p:-
1 = odd num
2 = even num
3 = odd
4 = even
5 = odd
6 = even
7 = odd
8 = even
9 = odd
10 = even.

*

```
height = int (input ("enter height in feet "))
if (height >3):
        print ('buy token")
else:
    print ("no token req").
```

o/p:- enter height in fet - 2
no token req.

*
```
height = int (input ("enter height in feet "))
if (height >3):
        print ("buy token")
        print (" no token req")
```

o/p:- enter height in feet
buy token
no token req.

* write a program to find factorial of given number.

ex:- 5! = 5*4*3*2*1 = 120.

print o/p inside the loop:-

```
num = int (input ("enter a value:"))
fact = 1
for i in range (1, num+1, 1):
    print ("i =", i)
    fact = fact * i
    print (fact).
```

o/p:- enter a value.: 5
i = 1
1
i = 2
2
i = 3
6
i = 4
24
i = 5
120.

print o/p outside the loop.

```
num = int (input ("enter a value:"))
fact = 1.
for i in range (1, num+1, 1):
    print ("i=", i)
    fact = fact * i
print (fact).
```

o/p :- enter a value : 5
120

write a program to calculate sum of num, sum of even num sum of odd num from 1 to 10.

sum num = 55 even num = 35 odd num = 25

```
total_sum = 0
even_sum = 0
odd_sum = 0
for i in range (1, 11, 1):
    total_sum = total_sum + i
    if (i%2 == 0):
        even_sum = even_sum + i
    else:
        odd_sum = odd_sum + i
print (f"total sum = {total_sum}, even sum = {even_sum},
        odd sum = {odd_sum}")
```

o/p :- total sum = 55
even sum = 30
odd sum = 25

* print 1st 10 fibanacci num.

0, 1, 1, 2, 3, 5, 8, 13, 21, 34

```
a = 0
b = 1
s = 0
for i in range (1, 11, 1):
    print (a)
    s = a + b
    a, b = b, s
```

o/p :-
0
1
1
2
3
5
8
13
21
34

* WHILE LOOP :-

a while loop is used to repeatedly executed a block of code as long as the given condition is true.

It is called an entry-controlled loop. because the condition is checked before the loop body executes.

Syntax:- # initialization

```
while condition:
    # statements
    # update (increment / decrement).
```

steps of execution:-

* initialization - set a starting value for loop control variable.
* conditional check:- if true → execute the loop body. if false → exit
* statements execution - code inside the loop runs.
* update - loop variable is changed (increment / decrement).
* the process repeates untill the condition becomes false.

Ex:-

```
i = 1              # initialization
while i <=5:       # condition
    print ("Hello", i)
    i +=1          # increment.
```

o/p:- Hello 1
Hello 2
Hello 3
Hello 4
Hello 5
Hello 6

* If the condition never becomes false, the loop runs infinite time
* used when the number of itterations is not known in advance
* must have an update stp (inc/dec), otherwise. infinite loop may occur

Ex:-

```
i =1
while (i<=10):
    print ("hi good moring")
    i= i+1.
```

o/p:- hi good morning
hi good morning
:
hi good morning.

*

```
num = int (input ("enter a value"))
i=1
while (i <=10):
    print (f"{num} x{i} = {num * i}")
    i = i+1
```

o/p:- enter a value 5
5x1=5
5x2=10
5x3=15
:
5x10=50

*

```
num = int (input ("enter a value"))
i=1
while (i<=10):
    print (f"{ num * i}")
    i = i+1
```

o/p:- enter a value 5
5
10
15
:

* outer loop & inner loop

outer loop :-

Controls rows (how many lines to print)

inner loop :-

Controls columns (how many values to print in each row).

Ex:-

```
rows = 4        # outer → no. of Rows
cols = 5        # inner → no. of cols

for i in range (rows):          # outer
    for j in range (cols):      # inner
        print ("*", end = " ")  # print in same row
    print ()                    # move to next line.
```

o/p :-

```
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
```

* outer loop (for i in range (rows))
* inner loop (for j in range (cols)
* After inner loop finishes, print ().

Ex:-

```
* for i in range (1,11,1):
      for j in range (1,11,1):
          print (i*j, end = "\t")
      print ().
```

o/p :-

```
1   2   3   4   5   .  .  .  .  .  10
2   4   6   .   .   .  .  .  .  .  20
3   6   .   .   .   .  .  .  .  .  30
4   8   .   .   .   .  .  .  .  .  40
5   .   .   .   .   .  .  .  .  .  50
.   .   .   .   .   .  .  .  .  .  .
.   .   .   .   .   .  .  .  .  .  .
10  20  30  40  50  .  .  .  .  .  100
```

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

o/p :-

```
i =                 j =
1 1 1 1 1           1 2 3 4 5
2 2 2 2 2           1 2 3 4 5
3 3 3 3 3           1 2 3 4 5
4 4 4 4 4           1 2 3 4 5
5 5 5 5 5           1 2 3 4 5
```

```
* for i in range (1,6,1):
      for j in range (1,6,1):
          if ((i*j) % 2 == 0):
              print (i*j, end = "\t")
          else:
              print (" ", end = "\t")
      print ("\n")

          (or)

      m = i*j
      if (m % 2 == 0):
          print (m, end = "\t").
```

o/p :-

```
        2           4
2       4       6       8       10
        6           12
4       8       12      16      20
        10          20.
```

```
* for i in range (1,6):
      for j in range (1, i+1, 1):
          print ("*", end = " ");
      print ()
```

o/p:-
```
*
* *
* * *
* * * *
* * * * *
```

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

o/p:-
```
* * * * *
* * * *
* * *
* *
*
```

```
* for i in range (1,6,1):
      for j in range (1, i+1, 1):
          print (j, end = " ");
      print ()
```

o/p:-
```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

```
* for i in range (5,0,-1):
      for j in range (1, i+1, 1):
          print (j, end = " ");
      print()
```

(or)

```
* for i in range (1,6,1):
      for j in range (1, 7-i, 1):
          print (j, end = " ");
      print ().
```

o/p:-
```
1 2 3 4 5
1 2 3 4
1 2 3
1 2
1
```

Ex:- print even num (1 to 10).

```
i = 2
while i <= 10:
    print (i)
    i += 2
```

o/p:-
```
2
4
6
8
10
```

Reverse count down.

```
i = 5
while i >= 1:
    print (i)
    i -= 1
```

o/p:-
```
5
4
3
2
1
```

* If condition is true → it becomes an infinite loop.

```
i = 1
while i <= 5:
    print (i)   # forgot i += 1
```

o/p:- It will never stop printing 1.

* NESTED LOOPS in python :-

A nested loop means one loop inside another loop.

* The outer loop runs first
* for each time the outer loops runs, the inner loop runs completely
* After the inner loop finishes, control goes back to the outer loop.

* ~~for loop inside for loop~~

simple nested loop.

```
for i in range (2):          # outer loop → run 2 times
    for j in range (3):      # inner loop → run 3 times for each Outer loop
        print (i,j)
```

o/p :-
```
0 0
0 1
0 2
1 0
1 1
1 2
```

* outer loop runs 2 times (i = 0, 1) 2 x 3 = 6 executions
* inner loop run 3 times each outer loop (j = 0, 1, 2).

* pattern using nested loop.

```
for i in range (3):          # outer → rows
    for j in range (3):      # inner → columns
        print ("*", end = " ")
    print ()
```

o/p :-
```
* * *
* * *
* * *
```

1) for loop inside for loop :-

It means we are using one for loop inside another for loop.

* The outer loop controls the rows (or bigger cycle).
* The inner loop control the colums (or smaller cycle).
* for each iteration of the outer loop, the inner loop run completely.

ex :- square of stars :-

```
for i in range (3):          # outer → run 3 times
    for j in range (3):      # inner → run 3 times each outer loop
        print ("*", end = " ")
    print ()                 # move to next line after inner loop.
```

o/p :-
```
* * *
* * *
* * *
```

0 - i = 0 → inner → j = 0, 1, 2 . o/p :- * * *

i = 1 inner - runs again . J = 0, 1, 2 → * * *

i = 2 inner - " " J = 0, 1, 2 * * *

Number triangle :-

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

o/p :-
```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

Explain :-

i = 1 → p - 1

i = 2 → p - 1 2

i = 3 → p - 1 2 3

i = 4 → p - 1 2 3 4

i = 5 → p - 1 2 3 4 5

Multiplication :-

```
for i in range (1,6):        # 0 → R (1 to 5)
    for j in range (1,6):    # 1 → C (1 to 5)
        print (i * j, end = "\t")
    print ().
```

o/p :-

1	2	3	4	5
2	-	-	-	10
3	-	-	-	15
4	-	-	-	20
5	10	15	20	25

2) While loop inside while loop :-

A while loop inside another while loop is called a nested while loop.

It means that one loop (the inner loop) is placed inside another loop (the outer loop).

The inner while loop executes completely for each iteration of the outer loop while loop.

in other words for every single time the outer loop runs once, the inner loop runs all its iterations.

(or)

A nested while loop is used when you need to repeate a set of statements multiple times inside another repeating process.

Syntax :-

```
# initialize outer loop variable
i = starting_value.
while condition_for_outer_loop:
    # initialize inner loop variable
    j = starting value.
    while condition_for_inner_loop:
        # inner loop stmt
        J += 1      # increment inner loop variable.
# outer loop st.
i += 1      # increment outer loop var.
```

Ex:

```
i = 1
while i <= 3:
    j = 1
    while j <= 2:
        print(f"i = {i}, j = {j}")
        j += 1
    i += 1.
```

O/P :- i=1, j=1
i=1, j=2
i=2, j=1
i=2, j=2
i=3, j=1
i=3, j=2

Explo :- 1) i=1, out cond. True → inner loop start
j=1 → print → j=2 → print → inner loop ends.
2) outer loop incre → i=2, again inner loop run fully
3. Repeat until j>3 → outer loop stop.

total outer 3 × inner 2 = 6.

Stars pattern :-

```
i = 1
while i <= 5:
    j = 1
    while j <= i:
        print("*", end=" ")
        j += 1
    print()      # new line after inner loop finishes.
    i += 1.
```

O/P :-
*
* *
* * *
* * * *
* * * * *

*

```
i = 1
while (i <= 10):
    j = 1
    while (j <= 10):
        print(i*j, end="\t")
        j = j + 1
    print()
i = i + 1
```

O/P :-

1	2	3	4	5	6	7	8	9	10
2	-	-	-	-	-	-	-	-	20
3	-	-	-	-	-	-	-	-	30
4	-	-	-	-	-	-	-	-	40
5	-	-	-	-	-	-	-	-	50
6	-	-	-	-	-	-	-	-	60
7	-	-	-	-	-	-	-	-	70
8	-	-	-	-	-	-	-	-	80
9	-	-	-	-	-	-	-	-	90
10	-	-	-	-	-	-	-	-	100

```
* i=1
while (i<=5):
    j=1
    while (j<=i):
        print (i, end=" ")
        j=j+1
    print ()
    i=i+1
```

o/p:-
```
1
2 2
3 3 3
4 4 4 4
5 5 5 5 5
```

```
* i=1
while (i<=5):
    j=1
    while (j<=i):
        print (j, end=" ")
        j=j+1
    print ()
    i=i+1
```

o/p:-
```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

* to print even num in multiples of 2 to pyramid shape.

```
num=2 (1) for odd num.
i=1
while (i<=5):
    j=1
    while (j<=i):
        print (num, end=" ")
        num=num+2
        j=j+1
    print ()
    i=i+1
```

o/p:-
```
2
4 6
8 10 12
14 16 18 20
22 24 26 28 30
```

```
* i=1
while (i<=5):
    j=1
    while (j<=i):
        print ("*", end=" ")
        j=j+1
    print ()
    i=i+1

i=1
while (i<=4):
    j=4
    while (j>=i):
        print ("*", end=" ")
        j-=1
    print ()
    i+=1
```

o/p:-
```
*
* *
* * *
* * * *
* * * * *
* * * *
* * *
* *
*
```

```
* for i in range (1,6,1):
        print ("*"*i, end=" ")
        print ("  "*(2*(5-i)), end=" ")
        print ("*"*i, end="\n")
for i in range (4,0,-1):
        print ("*"*i, end=" ")
        print ("  "*(2*(5-i)), end=" ")
        print ("*"*i, end="\n")
```

o/p:-
```
*         *
* *       * *
* * *     * * *
* * * *   * * * *
* * * * * * * * * *
* * * *   * * * *
* * *     * * *
* *       * *
*         *
```

```
print ("*"*i, end=' ')
print ()
```

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

o/p:-

```
    *
   * *
  * * *
 * * * *
* * * * *
```

```
* for i in range (1,6,1):
    for j in range (5, i, -1):
        print (" ", end = " ")
    for k in range of (1,i+1,1):
        print ("*", end = " ")
    print ()
for i in range (4,0,-1):
    for j in range (5, i, -1):
        print (" ", end =" ")
    for k in range (1, i+1, 1):
        print ("*", end =" ")
    print ()
```

o/p:-

```
    *
   * *
  * * *
 * * * *
* * * * *
 * * * *
  * * *
   * *
    *
```

3) write a syntax of for loop inside while loop

A for loop inside a while loop means that the for loop is nested with in the while loop.

* The while loop repeatedly executes a block of codes as long as a condition is true.
* The for loop execute a bloock of code for each item in a sequence
* when a for loop is placed inside a while loop, the for loop runs completely every time the while loop executes once.

Syntax:-

```
# initializing of while loop variable.
while_cond_var = initial_value.

# while loop cond
while while_cond_var < some_value:
    # for loop inside while loop.
    For loop_variable in range (start, end, step):    # for loop iterates over a sequence
        #statement(s) inside for loop
        print (loop_var)
```

statement inside while loop outside for loop.
print ("end of one while iteration")
increment or decrement of while loop var
while_cond_var += 1 # or -=1.

Ex:-
```
i=1  # initialize
while i<=3:  # while loop runs as long as i<=3
    print (f"while loop iteration {i}")
    for j in range (1,4):  # for loop inside while loop, j = 1,2,3.
        print (f" for loop value : {j}")
    i += 1  # increment while loop var.
```

O/P:-
```
while loop iteration 1
 for loop value : 1
 "   "    "  : 2
 "   "    "  : 3
while loop iteration 2
 for loop value : 1
 "   "    "  : 2
 "   "    "  : 3
while loop iteration 3.
 for loop value : 1
 "   "    "  : 2
 "   "    "  : 3.
```

*
```
i=1  # initialize
while i<=5:  # while loop runs as long as i<=5.
    for j in range(i):  # j goes from 0 to i-1
        print ("*", end=" ")  # print star in same line
    print()  # move to next line after each for loop
    i += 1  # inc while loop var.
```

O/P:-
```
*
* *
* * *
* * * *
* * * * *
```

4) while loop inside for loop:-

while loop inside a for loop is a nested loop structure where:

* The for loop runs a fixed number of iterations over a sequence (like range() or a list).
* for each iteration of the for loop, a while loop executes repeatedly as long as its condition is true.

Syntax:-
```
# for loop
for var in range (start, stop, step):  # for loop iterates over a sequence
    while_var = initial_value.  # initialize while loop inside for loop var
    while while_var < some_val:  while loop inised for loop.
# statement inside while loop.  print (f" For loop value : {var}, while loop value:
                                      {while_var}")
```

```
# incre / dec of while loop var
while-var +=1  # or while - var -=1
print (f"end of iteration {var} of for loop") # stm inside for loop but
                                               out side while loop.
```

Ex:-

```
for i in range (1,4): # outer loop
    j =1      # initialize variable for inner while loop.
    while j <=2:  # inner while loop
        print (i,j)
        j+=1      # inc j to avoid infinite loop.
```

o/p :- 11
12
21
22
31
32

* write a program to check the given num is a prime or not.

```
num = int (input ("enter a number :"))
count =0
for i in range (2, num, 1)
    if (num %. i ==0):
        count = count +1
    else :
        count =count
print (count)
if (count ==0):
    print (" It is a prime num")
else :
    print (" it is not a prime")
```

o/p :- enter a num : 5
0
It is a prime num

* write a prg to count even, odd, natural num frol 1 to 20.

```
num =int (input ("enter a num"))
count =0
for i in range (1,21,1):
    if (num):
        count = count

if (count):
    print ("It is a even num")
else :
    print ("It is a odd num").
```

o/p :- enter a num 9
It is a odd num

*

```
en =0
on =0
nn =0
for i in range (1,21,1):
    nn =nn+1
    if (i%.2 ==0):
        en =en +1
    else :
        on =on+1
print (f" natural num ={nn}\n even num = {en}\n odd num ={on}")
```

o/p :- natural = 20
even = 10
odd = 10

* n1 = int (input ("enter a num"))
n2 = int (input ("enter a num"))
en = 0
on = 0
nn = 0
for i in range (n1, n2+1, 1):
    nn = nn + 1
    if (i % 2 == 0):
        en = en + 1
    else:
        on = on + 1
print (f" natural num = {nn} \n even num = {en} \n odd num = {on}")

o/p:- enter a num 8
enter a num 10
natural num = 3
even num = 2
odd num = 1

* num = int (input ("enter a num:"))
count = 0
for i in range (2, num, 1):
    if (num % i == 0):
        print (f"{num} is not a prime")
        break
else:
    print (num, "is a prime num").

o/p:- enter a num: 9
9 is not a prime

* for i in range (2, 11, 1):
    for j in range (2, i, 1):
        if (i % j == 0):
            break
    else:
        print (i)

o/p:- 2
3
5
7.