LOOPS
OR
CONTROL STATEMENT

ITERATION

CONTROL STATEMENT (LOOPING STATEMENT)

- Program statement are executed sequentially one after another. In some situations, a block of code needs of times
- These are repetitive program codes, the computers have to perform to complete tasks.
- The following are the loop structures available in python
  - While statement
  - for ….loop statement
  - Nested loop statement

while loop statement

- A while loop statement in python programming language repeatedly executes a target statement as long as a given condition is true.
- Syntax:
  - while expression:
    - statement(s)

Examples of a while loop

- Write a program to find the sum of number
- n=int(input("enter no"))
- s=0
- while(n>0):
  - s=s+n
  - n=n-1
- print("the sum is",s)

OUTPUT

enter no 5
the sum is 15

```
a=1
while a<5:
  print a*a
  a=a+1
Output:
1 4 9 16
```

```
a=1
while a<5:
  print a,'*',a,'=',a*a
  a=a+1
Output:
1*1=1
2*2= 4
3*3=9
4*4=16
```

In while loop- Infinite loop

```
while 1: print("*") #Infinite loop
while True: print("*") #Infinite loop
while 1:
    print("*") #Infinite loop
while True:
    print("*") #Infinite loop
```

Using else statement with while loops

- Python supports have an else statement associated with a loop statement
- If the else statement is used with a while loop, the else statement is executed when the condition false.

```
Program to illustrate the else in while loop
c=0
while c<3:
        print("inside loop")
        c=c+1
else:
        print("outside loop")
```

OUTPUT

```
inside loop
inside loop
inside loop
outside loop
```

for loop statement

- The for loop is another repetitive control structure, and is used to execute a set of instructions repeatedly, until the condition becomes false.
- The for loop in python is used to iterate over a sequence(list,tuple,string) or other iterable objects. Iterating over a sequence is called traversal.
- Syntax:
  - for val in expression:
    - Body of the for loop

collection → String | Tuple | List | Dictionary

range() → range (begin,end-1,step_value)

- For VN in [ ] :

range(start,end-1,step_value)

end =n-1

```
range(n)        #n>end
range(a,n)      #a>start,n>end
range(a,n,sv)   #a>start,n>end,sv>step value
```

Ex.

```
range(5)      ➡ 0,1,2,3,4
range(1,5)    ➡ 1,2,3,4
range(1,8,2)  ➡ 1,3,5,7
range(0,7)    ➡ 0,1,2,3,4,5,6
```

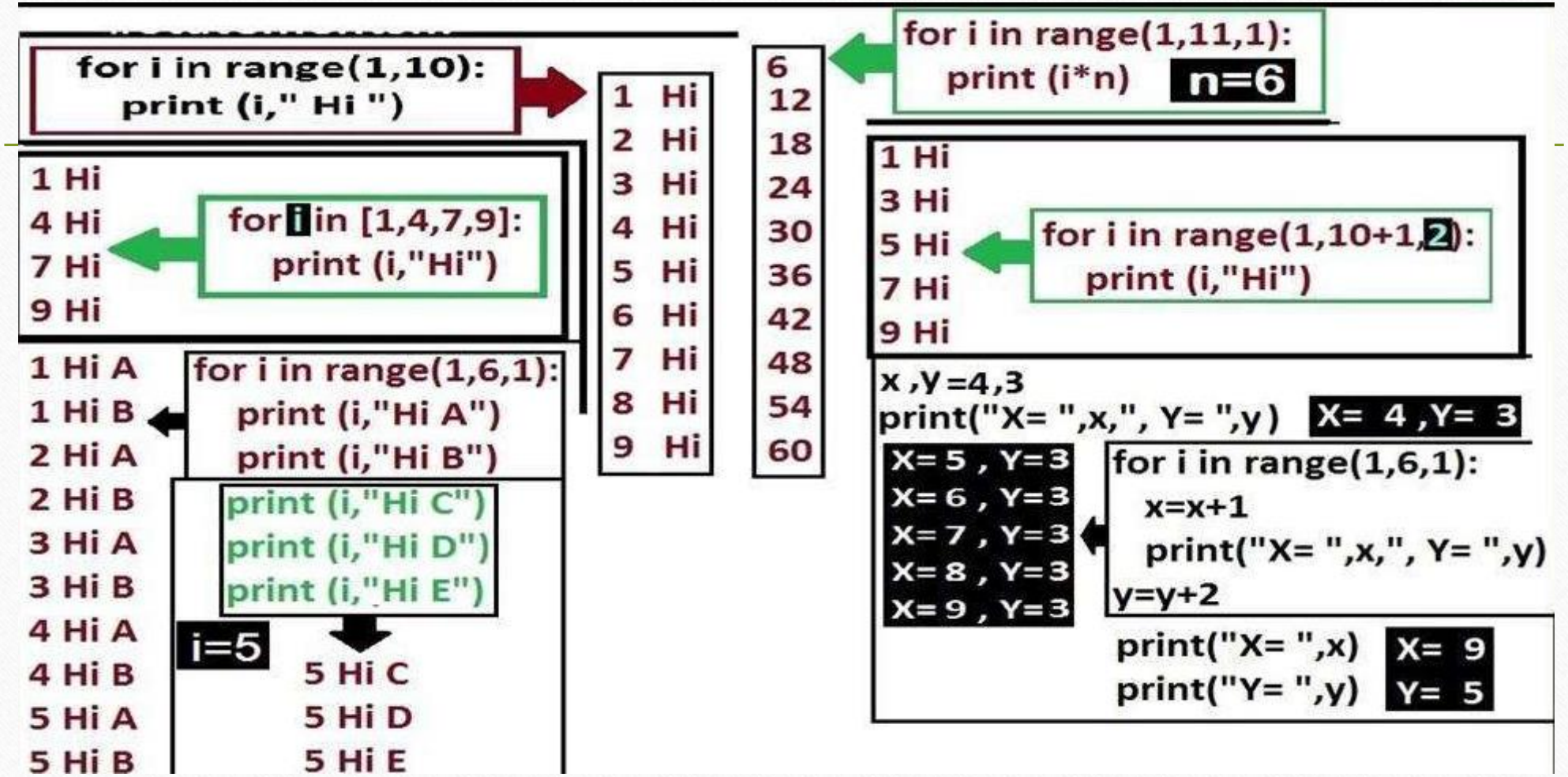
```
for i in [1,4,7]:
    print(a,end=",")
```
Output: 1,4,7,

```
for i in (1,3,5,6):
    print(i,end=",")
```
Output: 1,3,5,6,

```
for ch in "DotPyEdu":
    print(ch,end=",")
```
Output: D,o,t,P,y,E,d,u,

```
for i in {1:"PYTHON",2:"PHP",3:"JAVA"}:
    print(i,end=",")
```
Output:1:"PYTHON",2:"PHP",3:"JAVA"

```
for VN range(SV,EV-1,Interval)
    #Statements...
```



for loop and for loop with else clause

For – else clause

For loops also may have the optional **else** clause

- for *var in Group* :
  Statements

  else :
  Statements

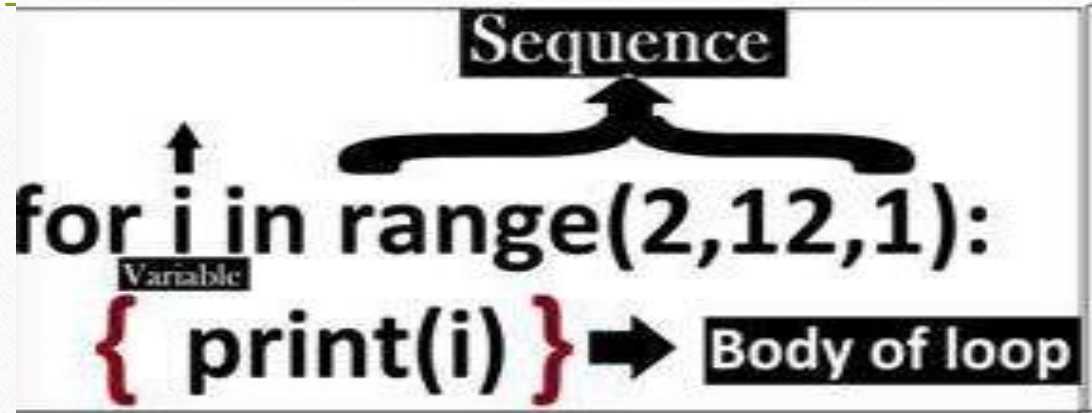
```
>>>for x in range(9):
        print x
   else :
        y = x
>>> print y
```

```
for i in range(10):
      print("Inside Loop for ",i)
else:
    print("Loop is Over Now ")
    print("The finel value of i is  ",i)
```

Output

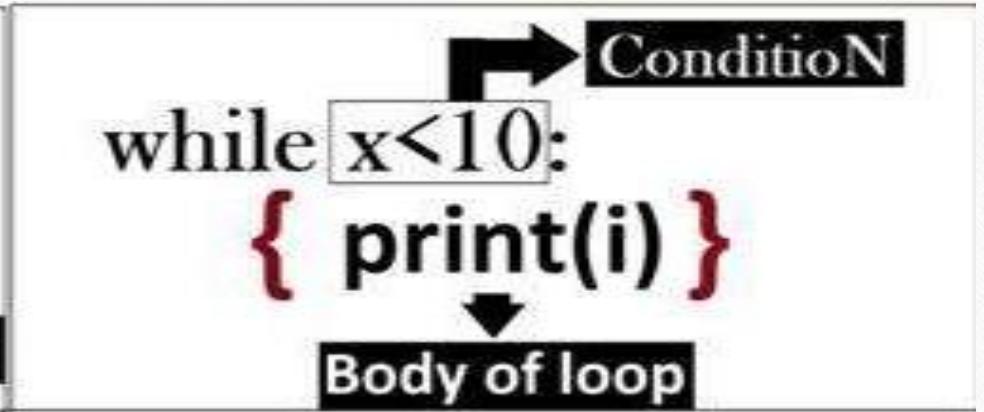
```
Inside Loop for  0
Inside Loop for  1
Inside Loop for  2
Inside Loop for  3
Inside Loop for  4
Inside Loop for  5
Inside Loop for  6
Inside Loop for  7
Inside Loop for  8
Inside Loop for  9
Loop is Over Now
The finel value of i is   9
```

Difference between for and while loop



Ex: for i in range(1,5,1):
print(i)

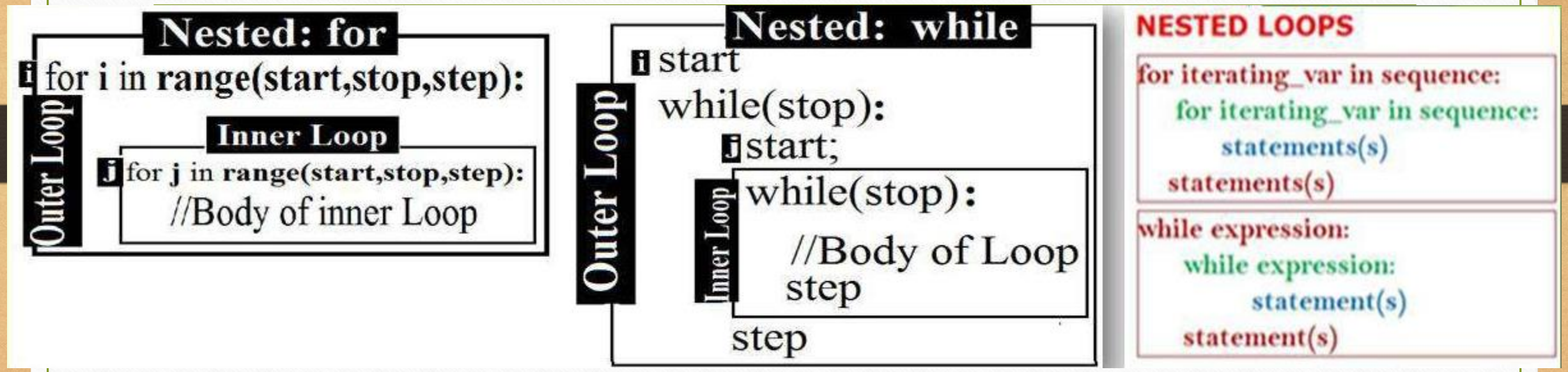
OUTPUT → 1 2 3 4 5



Ex: x=1 whiel x<=5:
print(x)
x+=1 #x=x+1

OUTPUT → 1 2 3 4 5

Nested Loop


Nested: for
i for i in range(start,stop,step):
Outer Loop
Inner Loop
j for j in range(start,stop,step):
//Body of inner Loop
Nested: while
i start
while(stop):
Outer Loop
j start;
Inner Loop
while(stop):
//Body of Loop
step
step
NESTED LOOPS
for iterating_var in sequence:
for iterating_var in sequence:
statements(s)
statements(s)
while expression:
while expression:
statement(s)
statement(s)

Examples of Nested Loop

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

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

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

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

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