



SNS COLLEGE OF ENGINEERING

Kurumbapalayam(Po), Coimbatore – 641 107

Accredited by NAAC-UGC with 'A' Grade

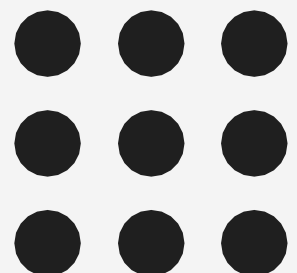
Approved by AICTE, Recognized by UGC & Affiliated to Anna University, Chennai

**Department of Artificial Intelligence and
Data Science**

**Course Name – Computational Thinking and
Python Programming**

I Year / I Semester

Unit 3-CONTROL FLOW, FUNCTIONS





ITERATION/CONTROL STATEMENTS:

- **state**
- **while**
- **for**
- **break**
- **continue**
- **pass**

State:

Transition from one process to another process under specified condition with in a time is called state.

-

While loop:

-

While loop statement in Python is used to repeatedly executes set of statement as long as a given condition is true.

In while loop, test expression is checked first. The body of the loop is entered only if the test_expression is True. After one iteration, the test expression is checked again. This process continues until the test_expression evaluates to False.

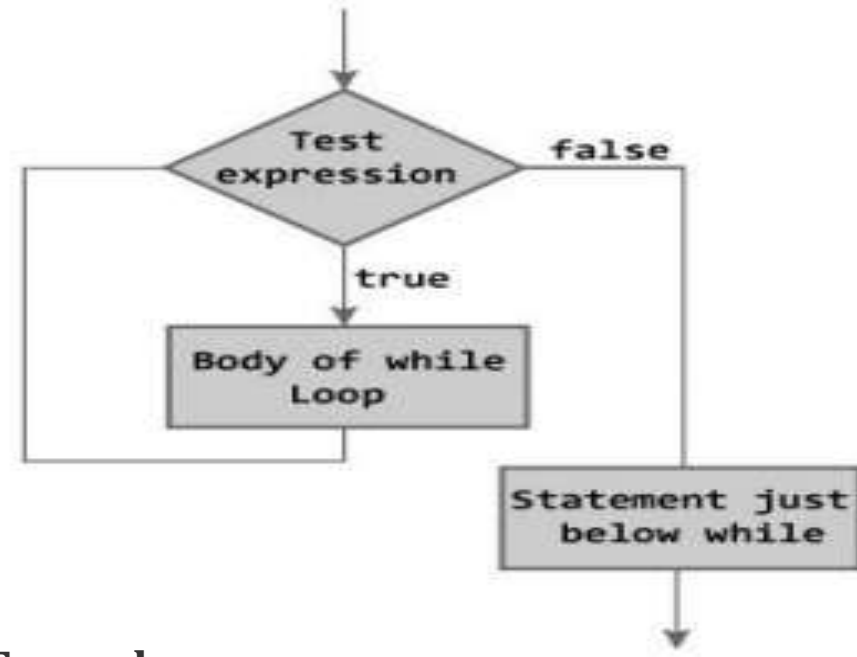
In Python, the body of the while loop is determined through indentation.

- The statements inside the while starts with indentation and the first unindented line marks the end.

Syntax:

```
initial value
while(condition):
    body of while loop
    increment
```

Flowchart:



Examples:

1. program to find sum of n numbers:
2. program to find factorial of a number
3. program to find sum of digits of a number:
4. Program to Reverse the given number:
5. Program to find number is Armstrong number or not
6. Program to check the number is palindrome or not

Sum of n numbers:

```
n=eval(input("enter n"))
```

```
i=1
```

```
sum=0
```

```
while(i<=n):
```

```
sum=sum+i
```

```
i=i+1
```

```
print(sum)
```

output

```
enter n
```

```
10
```

```
55
```

Factorial of a numbers:

```
n=eval(input("enter n"))  
i=1  
fact=1  
while(i<=n):  
fact=fact*i  
i=i+1  
print(fact)
```

output

```
enter n  
5  
120
```

For loop:

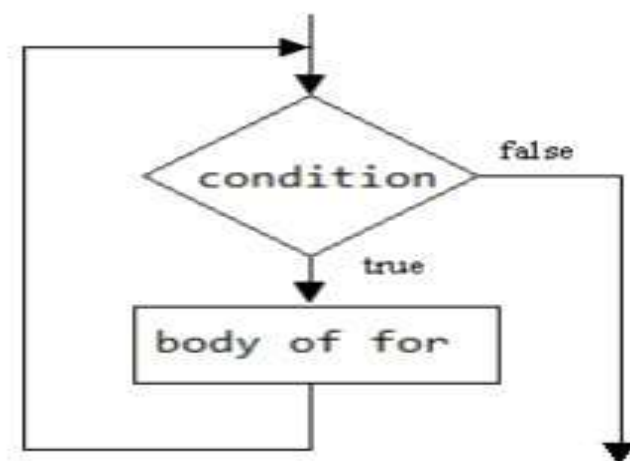
for in range:

- We can generate a sequence of numbers using range() function. range(10) will generate numbers from 0 to 9 (10 numbers).
- In range function have to define the start, stop and step size as range(start,stop,step size). step size defaults to 1 if not provided.

Syntax:

```
for i in range(start,stop,steps):  
    body of for loop
```

Flowchart:



For in sequence

The for loop in Python is used to iterate over a sequence (list, tuple, string). Iterating over a sequence is called traversal. Loop continues until we reach the last element in the sequence. The body of for loop is separated from the rest of the code using indentation.

```
for i in sequence:  
    print(i)
```

Sequence can be a list, strings or tuples

s.no	sequences	example	output
1.	For loop in string	for i in "Ramu": print(i)	R A M U
2.	For loop in list	for i in [2,3,5,6,9]: print(i)	2 3 5 6 9
3.	For loop in tuple	for i in (2,3,1): print(i)	2 3 1



Examples:

1. print nos divisible by 5 not by 10:
2. Program to print fibonacci series.
3. Program to find factors of a given number
4. check the given number is perfect number or not
5. check the no is prime or not
6. Print first n prime numbers
7. Program to print prime numbers in range

print nos divisible by 5 not by 10

```
n=eval(input("enter a"))  
for i in range(1,n,1):  
if(i%5==0 and i%10!=0):  
print(i)
```

output

```
enter a:30  
5  
15  
25
```



Fibonacci series

a=0

b=1

```
n=eval(input("Enter the number of terms: "))
```

```
print("Fibonacci Series: ")
```

```
print(a,b)
```

```
for i in range(1,n,1):
```

```
    c=a+b
```

```
    print(c)
```

```
    a=b
```

```
    b=c
```

output

Enter the number of terms: 6

Fibonacci Series:

0 1

1

2

3

5

8

Loop Control Structures


BREAK

- ✓ Break statements can alter the flow of a loop.
- ✓ It terminates the current
- ✓ loop and executes the remaining statement outside the loop.
- ✓ If the loop has else statement, that will also gets terminated and come out of the loop completely.

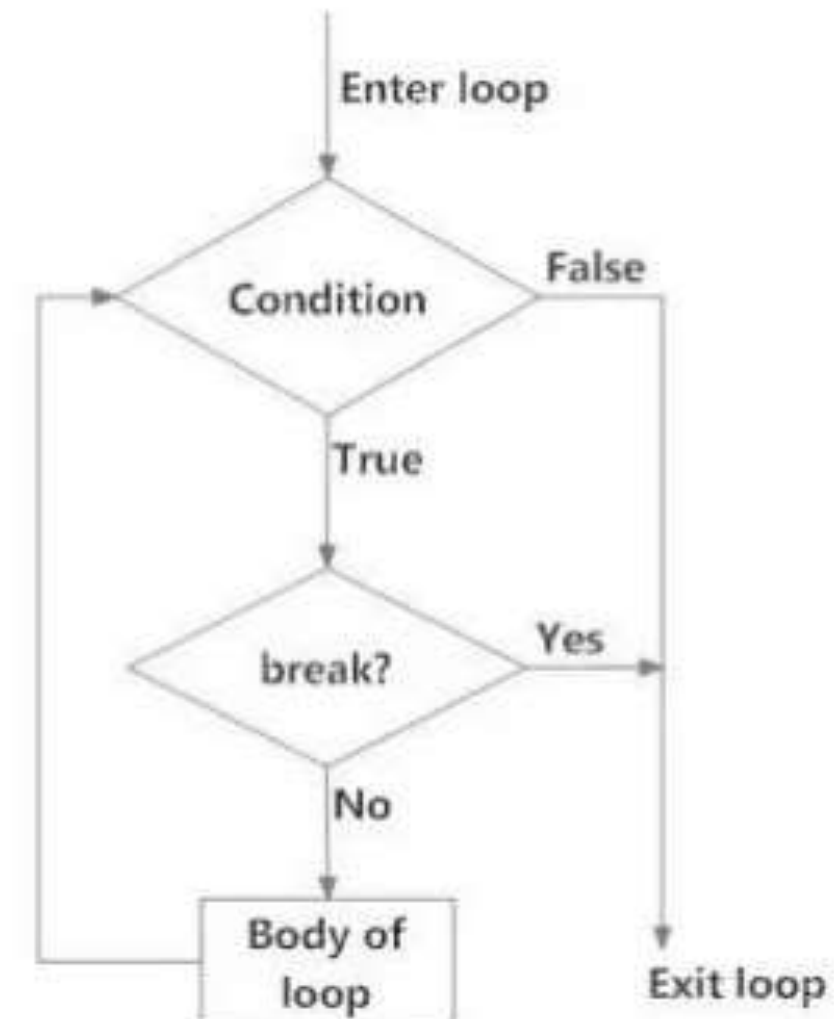
Syntax:

Break

```
while (test Expression):  
    // codes  
    if (condition for break):  
        break  
    // codes
```



Flowchart



example

```
for i in "welcome":  
    if(i=="c"):  
        break  
    print(i)
```

Output

```
w  
e  
l
```

CONTINUE

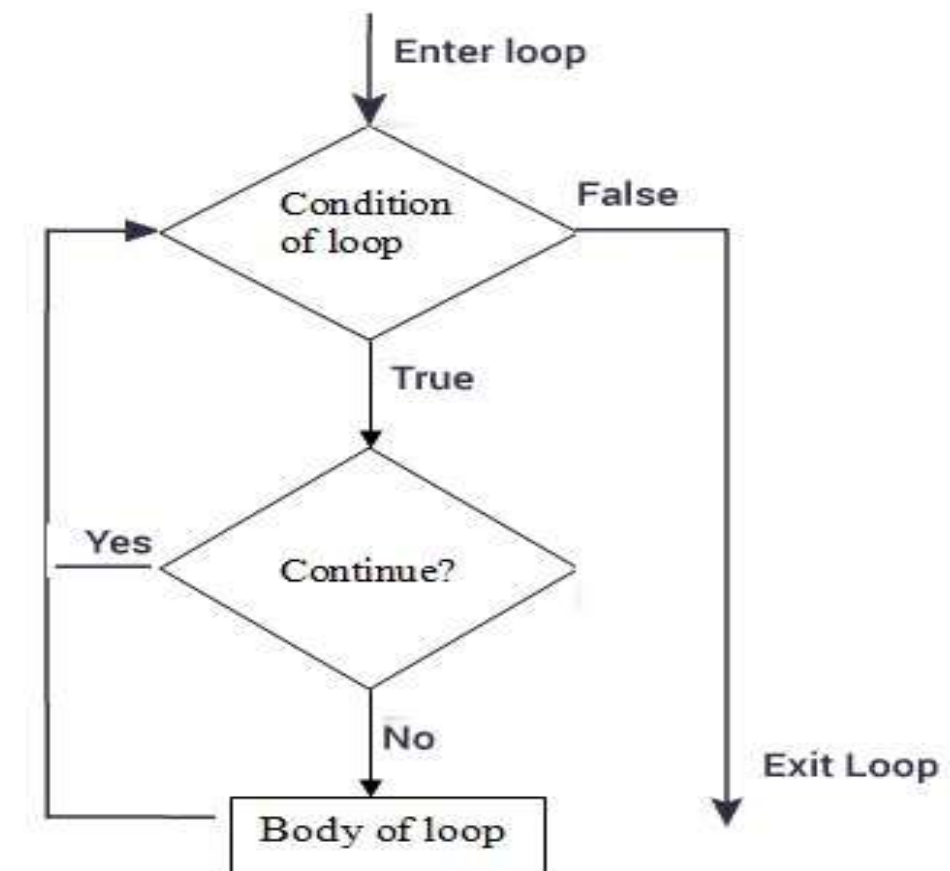
It terminates the current iteration and transfer the control to the next iteration in the loop.

Syntax:

Continue

Flowchart

```
while (test Expression):  
    // codes  
    if (condition for continue):  
        continue  
    // codes
```



**Example:**

```
for i in "welcome":  
    if(i=="c"):  
        continue  
    print(i)
```

Output

```
w  
e  
l  
o  
m  
e
```

PASS

- ❖ It is used when a statement is required syntactically but you don't want any code to execute.
- ❖ It is a null statement, nothing happens when it is executed.

Syntax:

```
pass  
break
```

Example

```
for i in "welcome":  
    if (i == "c"):  
        pass  
    print(i)
```

Output

```
w  
e  
l  
c  
o  
m  
e
```



else in for loop:

- ❖ If else statement is used in for loop, the else statement is executed when the loop has reached the limit.
- ❖ The statements inside for loop and statements inside else will also execute.

example

```
for i in range(1,6):  
    print(i)  
else:  
    print("the number greater than 6")
```

output

```
1  
2  
3  
4  
5    the number greater than 6
```

else in while loop:

- ❖ If else statement is used within while loop , the else part will be executed when the condition become false.
- ❖ The statements inside for loop and statements inside else will also execute.

Program

```
i=1  
while(i<=5):  
    print(i)  
    i=i+1  
else:  
    print("the number greater than 5")
```

output

```
1  
2  
3  
4  
5  
the number greater than 5
```