SNS College of Engineering
Department of Information Technology
Iterations

M.Malarmathi

Ap/IT

Indefinite Loops(while loop)

- While loops are called "indefinite loops" because they keep going until a logical condition becomes False

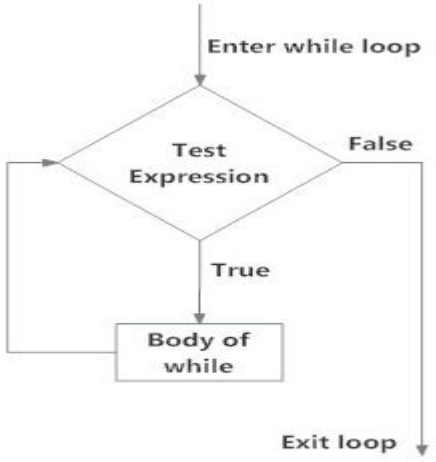
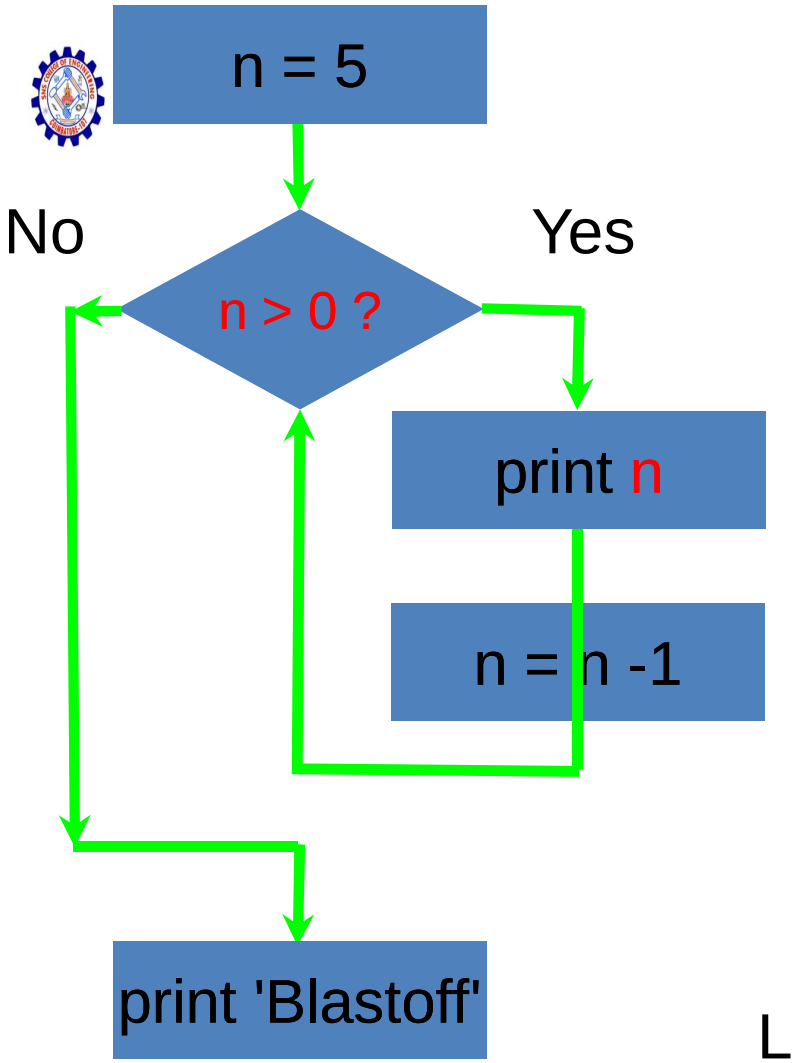


Fig: operation of while loop



Repeated Steps

Program:

```
n = 5
while n > 0 :
    print n
    n = n – 1
print 'Blastoff!'
print n
```

Output:

```
5
4
3
2
1
Blastoff!
0
```

Loops (repeated steps) have iteration variables that change each time through a loop. Often these iteration variables go through a sequence of numbers.

Definite Loops

- Run the loop once for each of the items in a set using the Python for construct
- These loops are called "definite loops" because they execute an exact number of times
- definite loops iterate through the members of a set

A Simple Definite Loop(for loop)

```
for i in [5, 4, 3, 2, 1] :
    print i
print 'Blastoff!'
```

```
5
4
3
2
1
Blastoff!
```

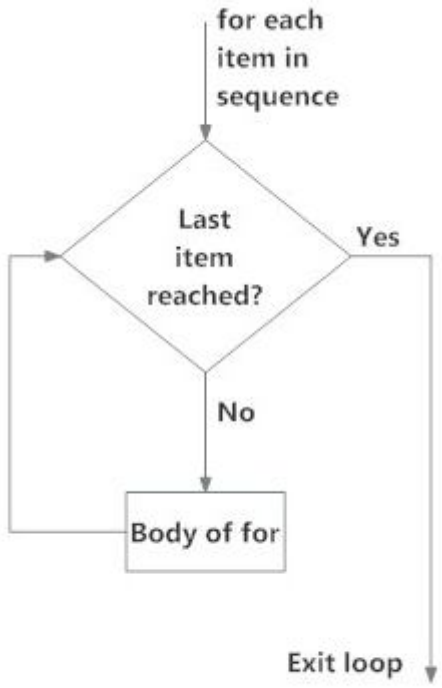


Fig: operation of for loop

Breaking Out of a Loop

- The break statement ends the current loop and jumps to the statement immediately following the loop
- It is like a loop test that can happen anywhere in the body of the loop

```
while True:
    line = raw_input('  ')
    if line == 'done' :
        break
    print line
print 'Done!'
```

```
> hello there
hello there
> finished
finished
> done
Done!
```

Breaking Out of a Loop

- The break statement ends the current loop and jumps to the statement immediately following the loop
- It is like a loop test that can happen anywhere in the body of the loop

```
while True:
    line = raw_input('> ')
    if line == 'done' :
        break
    print line
print 'Done!'
```

```
> hello there
hello there
> finished
Finished
> done
Done!
```

Example

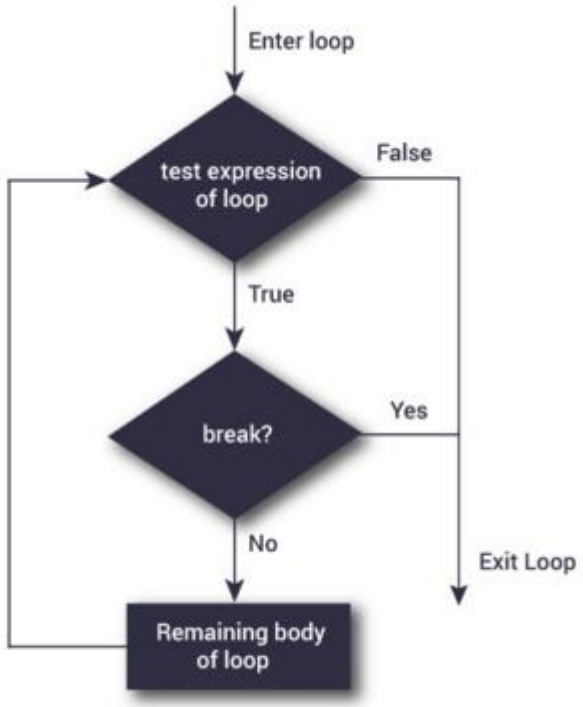
```
for letter in 'Python':     # First Example
  if letter == 'h':
break
 print 'Current Letter :', letter
```

OUTPUT:

Current Letter : P

Current Letter : y

Current Letter : t



Example

```
# Second Example
 var = 10
while var > 0:
      print 'Current variable value :', var
 var = var -1
  if var == 5:
    break
 print "Good bye!“
```

OUTPUT:

```
Current variable value : 10
Current variable value : 9
Current variable value : 8
Current variable value : 7
Current variable value : 6
Good bye!
```

Finishing an Iteration with continue

- The continue statement ends the current iteration and jumps to the top of the loop and starts the next iteration

```
while True:
    line = raw_input('> ')
    if line[0] == '#' :
        continue
    if line == 'done'
:       break
    print line
print 'Done!'
```

```
> hello there
hello there
> # don't print this
> print this!
print this!
> done
Done!
```

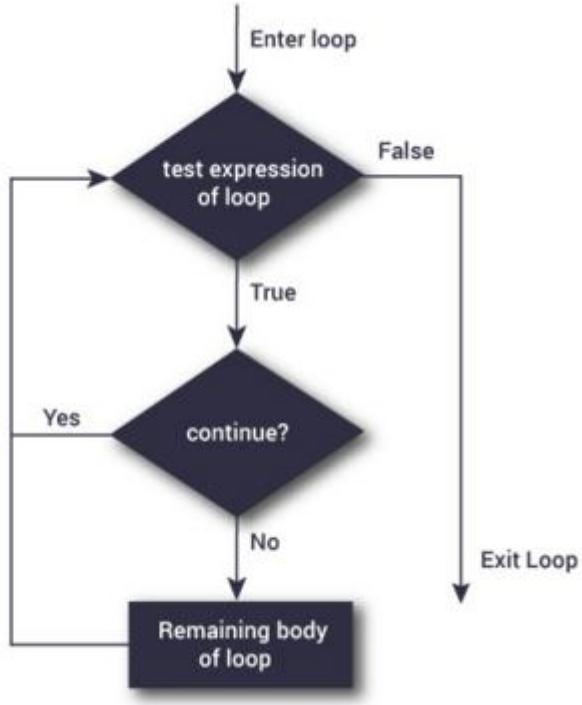
Finishing an Iteration with continue

- The continue statement ends the *current iteration* and jumps to the top of the loop and starts the next iteration

```
while True:
    line = raw_input('> ')
    if line[0] == '#' :
        continue
    if line == 'done' :
        break
    print line
print 'Done!'
```

```
> hello there
hello there
> # don't print this
> print this!
print this!
> done
Done!
```

```
while True:
   line = raw_input('> ')
   if line[0] == '#' :
        continue
    if line == 'done' :
        break
    print line
print 'Done!'
```



Example :Continue

```
for letter in 'Python':
if letter == 'h':
  continue
print 'Current Letter :', letter
```

OUTPUT:

```
Current Letter : P
Current Letter : y
Current Letter : t
Current Letter : o
Current Letter : n
```

Example

```
for x in range(7):
 if (x == 3 or x==6):
continue
print(x)
```

Example

```
var = 10
while var > 0:
        var = var -1
        if var == 5:
                continue
        print 'Current variable value :', var
print "Good bye!“
```

OUTPUT

```
Current variable value : 10
Current variable value : 9
Current variable value : 8
Current variable value : 7
Current variable value : 6
Current variable value : 4
Current variable value : 3
Current variable value : 2
Current variable value : 1
Good bye!
```

Pass

- The **pass** statement is a *null* operation
- nothing happens when it executes
- It is used when a statement is required syntactically but you do not want any command or code to execute.

Example

```
for letter in 'Python':
    if letter == 'h':
            pass
            print 'This is pass block‘
    print 'Current Letter :', letter
print "Good bye!“
```

OUTPUT:

```
Current Letter : P
Current Letter : y
Current Letter : t
This is pass block
Current Letter : h
Current Letter : o
Current Letter : n
Good bye!
```

Difference between various iterations

Pass	*Continue*	*Break*
Statement simply means 'do nothing'	continue with the loop	Breaks the loop
when the python interpreter encounters the pass statement, it simply continues with its execution	resume execution at the top of the loop or goes to next iteration	When a break statement is encountered, it terminates the block and gets the control out of the loop

While	For
Indefinite Loops	Definite Loop
the exit condition will be evaluated again, and execution resumes from the top	the item being iterated over will move to its next element

Thank you