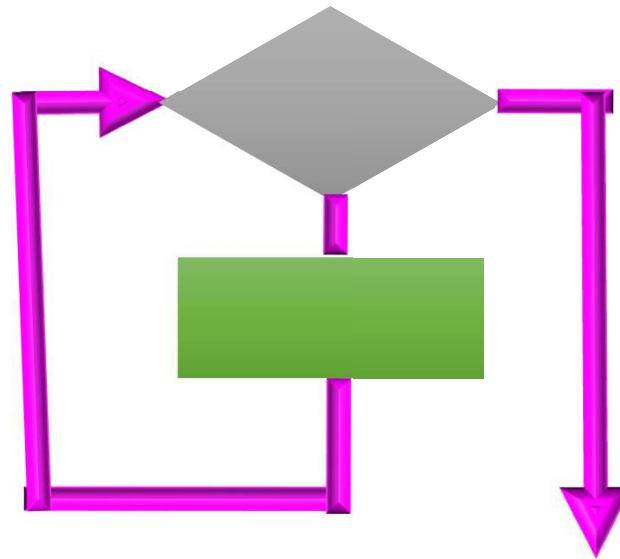


3. ITERATION OR LOOPING



ITERATION

3. ITERATION OR LOOPING



What is loop or iteration?

Loops can execute a block of code number of times until a certain condition is met.

OR

The iteration statement allows instructions to be executed until a certain condition is to be fulfilled.

The iteration statements are also called as loops or Looping statements.



3. ITERATION OR LOOPING

**Python provides two kinds of loops
& they are,**

for loop

while loop



for LOOP

for LOOP



Python's for-loop syntax is a more convenient alternative to a while loop when iterating through a series of elements. The for-loop syntax can be used on any type of iterable structure, such as a list, tuple str, set, dict, or file
Syntax or general format of for loop is,

```
for element in iterable:  
    body
```

for LOOP



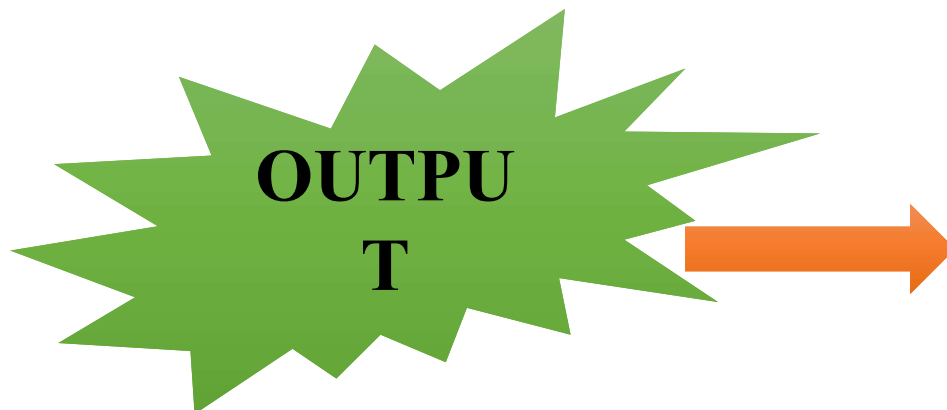
Python's for-loop syntax is a more convenient alternative to a while loop when iterating through a series of elements. The for-loop syntax can be used on any type of iterable structure, such as a list, tuple str, set, dict, or file
Syntax or general format of for loop is,

```
for element in iterable:  
    body
```

for LOOP

```
*Python 3.4.0: for_natno.py - C:/Python34...  
File Edit Format Run Options Windows Help  
def for_loop_example():  
    numbers=[765,23,56,89,14,78]  
    for i in numbers:  
        print(i)  
for_loop_example()  
Ln: 8 Col: 0
```

Till the list exhaust for loop will continue to execute.



```
Python 3.4.0 Shell  
File Edit Shell Debug Options  
Windows Help  
>>>  
765  
23  
56  
89  
14  
78  
>>>  
Ln: 8 Col: 0
```



for LOOP

range KEYWORD

for LOOP - range KEYWORD



The `range()` function returns a sequence of numbers, starting from 0 by default, and increments by 1 (by default), and ends at a specified number.

`range(start, stop, step)`

```
for n in range(3,6):  
    print(n)
```



```
x = range(3, 6)  
for n in x:  
    print(n)
```

for LOOP - range KEYWORD

#Generating series of numbers

```
Python 3.4.0: format.py - C:/Python34/forn...  
File Edit Format Run Options Windows Help  
def for_loop_example():  
    n=int(input("Enter number: "))  
    for i in range(0,n):  
        print(i)  
for_loop_example()  
Ln: 2 Col: 27
```

OUTPUT

```
Python 3.4.0 S...  
File Edit Shell Debug Options  
Windows Help  
Enter number: 4  
0  
1  
2  
3  
>>>  
Ln: 16 Col: 4
```

for LOOP - range KEYWORD

#Generating even numbers

```
*Python 3.4.0: format.py - C:/Python34/for...  
File Edit Format Run Options Windows Help  
def for_loop_example():  
    n=int(input("Enter number: "))  
    for i in range(0,n,2):  
        print(i)  
for_loop_example()  
Ln: 6 Col: 0
```

OUTPUT

```
Python 3.4.0 ...  
File Edit Shell Debug Options  
Windows Help  
Enter number: 6  
0  
2  
4  
>>>  
Ln: 12 Col: 4
```



for LOOP – len() FUNCTION

for LOOP - range KEYWORD



print string character by character

```
*Python 3.4.0: forstr.py - C:/Python34/fo...  
File Edit Format Run Options Windows Help  
#printing string char by char  
def for_loop_example():  
    name=input("Enter string: ")  
    for i in range(0,len(name)):  
        print(name[i])  
for_loop_example()  
Ln: 10 Co
```

```
Python 3.4.0 Shell  
File Edit Shell Debug Options Windows  
Help  
Enter string: Sainik  
S  
a  
i  
n  
i  
k  
>>> |  
Ln: 21 Col: 4
```

OUTPUT



else statement in loop

else statement in loop



else can be used in for and while loops the else body will be executed as and when the loop's conditional expression evaluates to false

OUTPUT

```
Python 3.4.0: breakinloop.py - C:/Python34/breakinloop.py
File Edit Format Run Options Windows Help

def break_in_loop():
    for i in range(0,5):
        print (i)
    else:
        print("Finally! Else of for executed")
break_in_loop()

Ln: 7 Col: 0
```

```
Python 3.4.0 Shell
File Edit Shell Debug Options Windows Help

=====
>>>
0
1
2
3
4
Finally! Else of for executed
>>>

Ln: 11 Col: 4
```