for more updates visit: www.python4csip.com

Loop – repetition of task

VINOD KUMAR VERMA, PGT(CS), KV OEF KANPUR &
SACHIN BHARDWAJ, PGT(CS), KV NO.1 TEZPUR

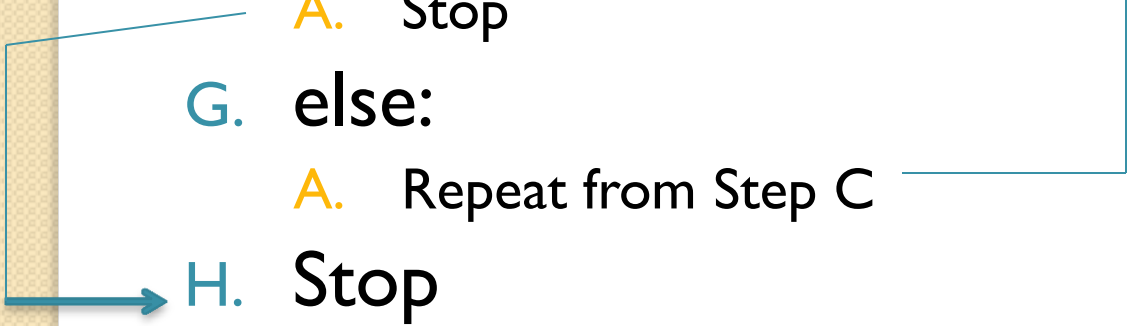
What is Loop?

- As we have already studied there are many situation where we need to repeat same set of tasks again and again. Like while writing table of any number we multiply same number from 1 to 10, multiplying number from n to 1 to find the factorial, In real life, to prepare parathas, suppose to prepare your bag with all items, daily coming to school again and again, etc. Please explore more real life conditions where repetition of task was done by you.

for more updates visit: www.python4csip.com

Pseudocode Reference

A. start
B. Input number(n) to print table
C. Let count=1
D. Display n * I
E. Add 1 to count
F. if count==10:
    A. Stop
G. else:
    A. Repeat from Step C
H. Stop



for more updates visit: www.python4csip.com

Python Loop Statements

- To carry out repetition of statements Python provide 2 loop statements
  - Conditional loop (while)
  - Counting loop (for)

for more updates visit: www.python4csip.com

range() function

- Before we proceed to for loop let us understand **range()** function which we will use in for loop to repeat the statement to **n number** of times.
- Syntax:
  - range(lower_limit, upper_limit)
- The range function generate set of values from **lower_limit** to **upper_limit-1**

range() function

- For e.g.
- **range(1,10)** will generate set of values from **1-9**
- range(0,7) will generate [0-6]
- Default step value will be +1 i.e.
- range(1,10) means (1,2,3,4,5,6,7,8,9)

range() function

- To change the step value we can use third parameter in range() which is step value
- For e.g.
- **range(1,10,2)** now this will generate **value [1,3,5,7,9]**
- Step value can be in –ve also to generate set of numbers in reverse order.
- range(10,0) will generate number as [10,9,8,7,6,5,4,3,2,1]

for more updates visit: www.python4csip.com

range() function

- To create list from 0 we can use
- **range(10)** it will generate **[0,1,2,3,4,5,6,7,8,9]**

Operators **in** and **not in**

- The operator ***in*** and ***not in*** is used in ***for*** loop to check whether the value is in the range / list or not
- For e.g.

```
>>> 5 in [1,2,3,4,5]
```

True

```
>>> 5 in [1,2,3,4]
```

False

```
>>>'a' in 'apple'
True
>>>'national' in 'international'
True
```

for more updates visit: www.python4csip.com

Program to check whether any word is a part of sentence or not

```
line = input("Enter any statement")
word = input("Enter any word")
if word in line:
    print("Yes ", word, "is a part of ",line)
else:
    print("No", word ," is not part of ",line)
```

for loop

- for loop in python is used to create a loop to process items of any sequence like List, Tuple, Dictionary, String
- It can also be used to create loop of fixed number of steps like 5 times, 10 times, n times etc using **range()** function.

Example – for loop with List

```
School=["Principal","PGT","TGT","PRT"]
for sm in School:
        print(sm)
```

Example – for loop with Tuple

```
Code=(10,20,30,40,50,60)
for cd in Code:
        print(cd)
```

Let us understand how for loop works

```
Code=(10,20,30,40,50,60)
for cd in Code:
        print(cd)
```

First it will create Tuple 'Code'

```
Code=(10,20,30,40,50,60)
for cd in Code:
        print(cd)
```

Then for loop starts, it will pick values one by one from **Code** and assign it to **cd**

```
Code=(10,20,30,40,50,60)
for cd in Code:
        print(cd)
```

Then for loop starts, it will pick values one by one from **Code** and assign it to **cd**

```
Code=(10,20,30,40,50,60)
for cd in Code:
    print(cd)
```

cd = 10

Then for loop starts, it will pick values one by one from **Code** and assign it to **cd**

```
Code=(10,20,30,40,50,60)
for cd in Code:
    print(cd)
```

cd = 10

OUTPUT

```
10
```

Then for loop starts, it will pick values one by one from **Code** and assign it to **cd**

```
Code=(10,20,30,40,50,60)
for cd in Code:
    print(cd)
```

cd = 20

OUTPUT

```
10
```

Then for loop starts, it will pick values one by one from **Code** and assign it to **cd**

```
Code=(10,20,30,40,50,60)
for cd in Code:
    print(cd)
```

cd = 20

OUTPUT

```
10
20

AND SO ON…
```

FOR LOOP WILL AUTOMATICALLY ENDS WHEN LAST ITEM OF LIST IS PROCESSED

for loop with string

```
for ch in 'Plan':
        print(ch)
```

The above loop product output

P

l

a

n

for more updates visit: www.python4csip.com

for with range()

Let us create a loop to print all the natural number from 1 to 100

```
for i in range(1,101):
        print(i,end='\t')
```

** here end='\t' will cause output to appear without changing line and give one tab space between next output.

for more updates visit: www.python4csip.com

Program to print table of any number

```
num = int(input("Enter any number "))
for i in range(1,11):
        print(num,'x',i,'=',num*i)
```

Program to find the sum of all number divisible by 7 between 1 to 100

```
sum=0
for i in range(1,101):
        if i % 7 == 0:
                sum+=i
print("total of number divisible by 7 between 1 to 100 is ",sum)
```

for more updates visit: www.python4csip.com

Lab work

1. WAP to enter any number and find its factorial
2. WAP to print the following fibonacci series

   0, 1, 1, 2, 3, 5, 8,……..n terms
3. WAP to enter 10 number and find the sum and average.
4. WAP to enter Lower_Limit, Upper_Limit and find the sum of all odds and evens number between the range separately

for more updates visit: www.python4csip.com

while loop

- While loop in python is conditional loop which repeat the instruction as long as condition remains true.
- It is entry-controlled loop i.e. it first check the condition and if it is true then allows to enter in loop.
- while loop contains various loop elements: **initialization**, **test condition**, **body of loop** and **update statement**

for more updates visit: www.python4csip.com

while loop elements

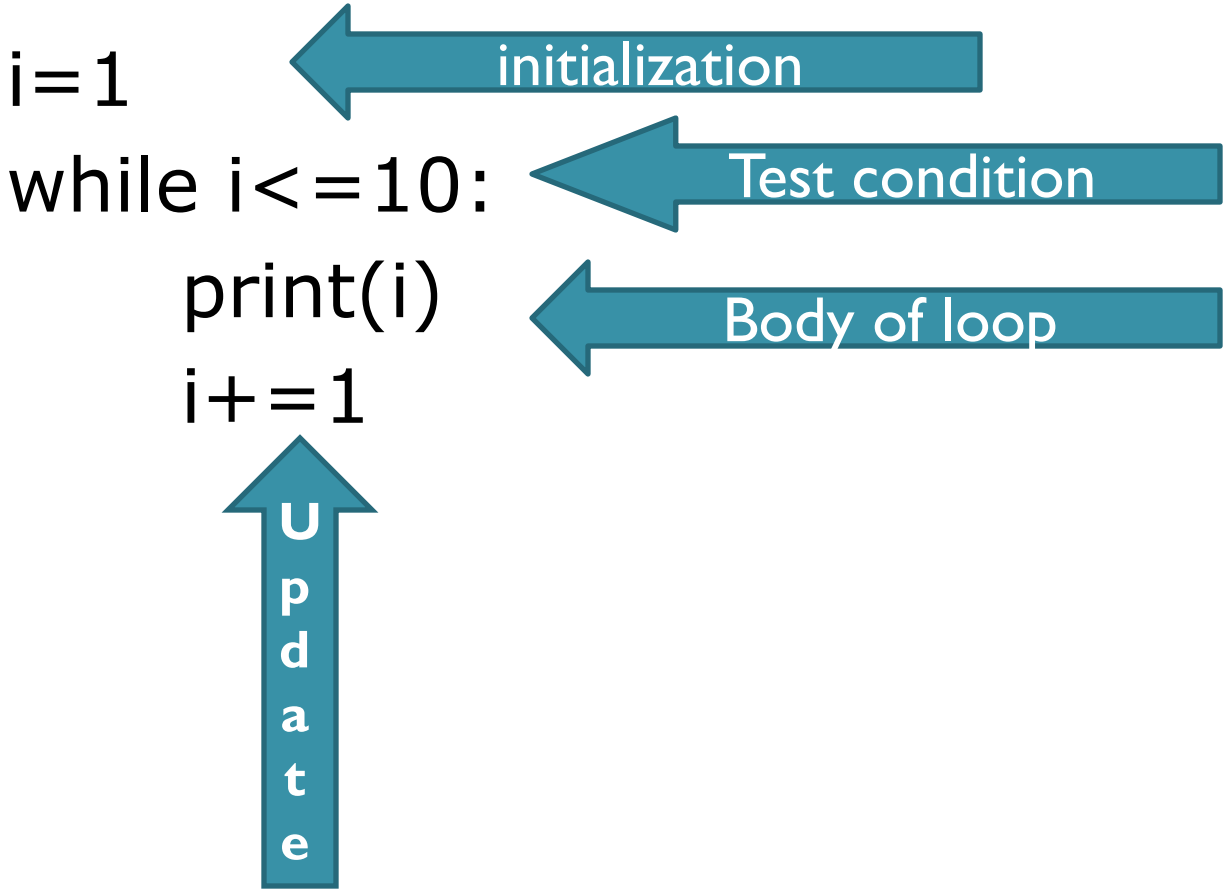
1. **Initialization :** it is used to give starting value in a loop variable from where to start the loop
2. **Test condition :** it is the condition or last value up to which loop will be executed.
3. **Body of loop :** it specifies the action/statement to repeat in the loop
4. **Update statement :** it is the increase or decrease in loop variable to reach the test condition.

Example of simple while loop

```
i=1
while i<=10:
        print(i)
        i+=1
```

for more updates visit: www.python4csip.com

Example of simple while loop



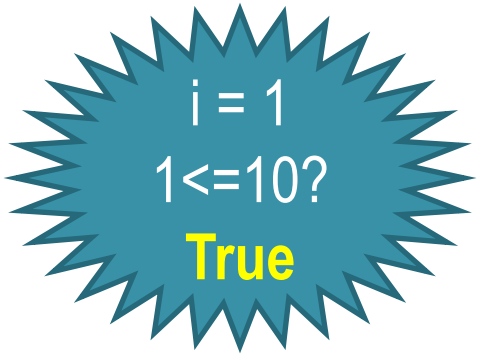
Let us see the flow of loop

i=1

```
while i<=10:
        print(i)
        i+=1
```

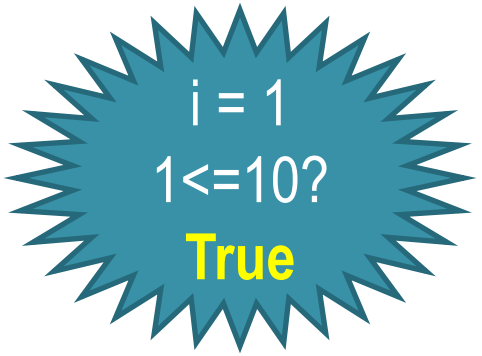
Let us see the flow of loop

```
i=1
while i<=10:
        print(i)
        i+=1
```



Let us see the flow of loop

```
i=1
while i<=10:
        print(i)
        i+=1
```

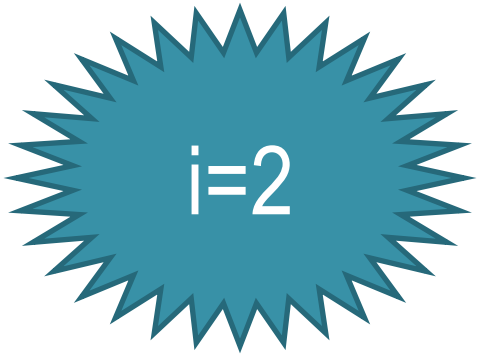


OUTPUT

1

Let us see the flow of loop

```
i=1
while i<=10:
        print(i)
        i+=1
```

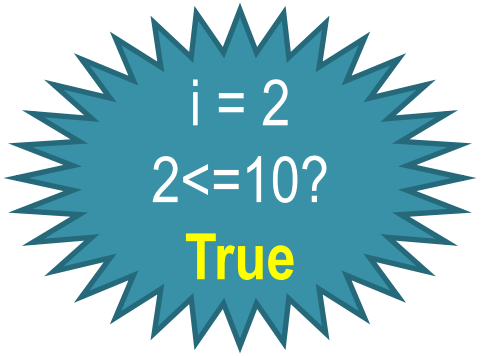


OUTPUT

1

Let us see the flow of loop

```
i=1
while i<=10:
        print(i)
        i+=1
```

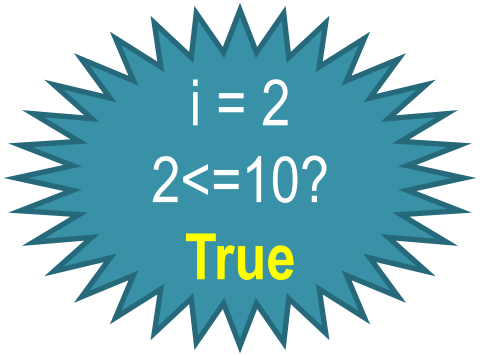


OUTPUT

1

Let us see the flow of loop

```
i=1
while i<=10:
        print(i)
        i+=1
```

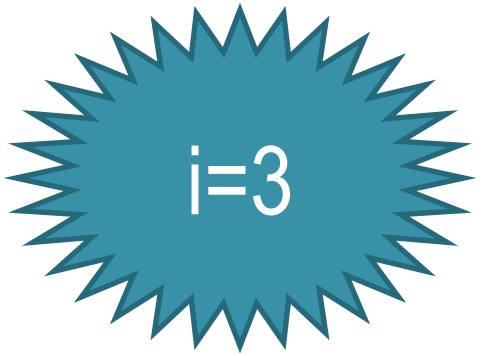


OUTPUT

1
2

Let us see the flow of loop

```
i=1
while i<=10:
        print(i)
        i+=1
```

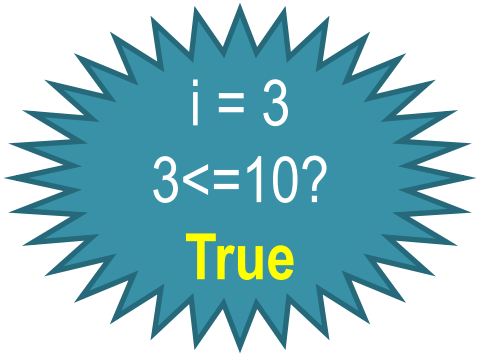


OUTPUT

1
2

Let us see the flow of loop

```
i=1
while i<=10:
        print(i)
        i+=1
```

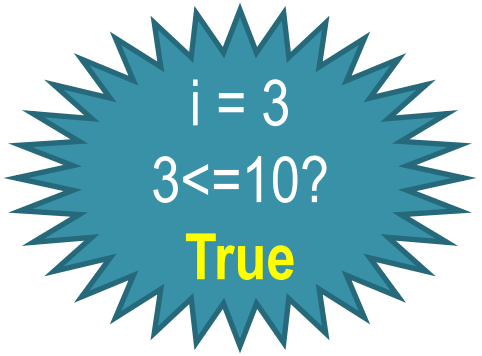


OUTPUT

1
2

Let us see the flow of loop

```
i=1
while i<=10:
        print(i)
        i+=1
```



OUTPUT

```
1
2
3
```

Let us see the flow of loop

```
i=1
while i<=10:
        print(i)
        i+=1
```

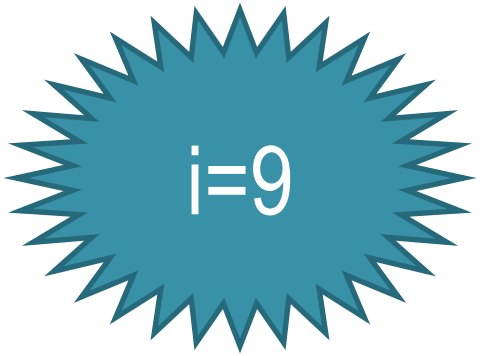
OUTPUT

1
2
3

In this way loop will execute for the values 4 to 10, let us see from the value 9

Let us see the flow of loop

```
i=1
while i<=10:
        print(i)
        i+=1
```



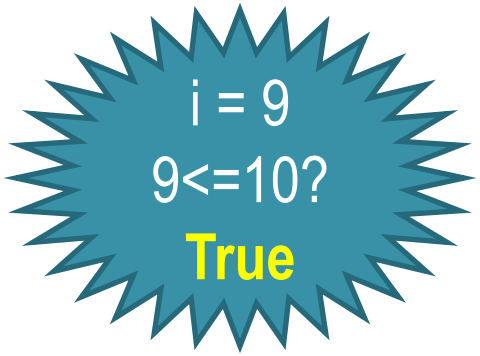
OUTPUT

1
2
3
4
5
6
7
8

In this way loop will execute for the values 4 to 10, let us see from the value 9

Let us see the flow of loop

```
i=1
while i<=10:
        print(i)
        i+=1
```

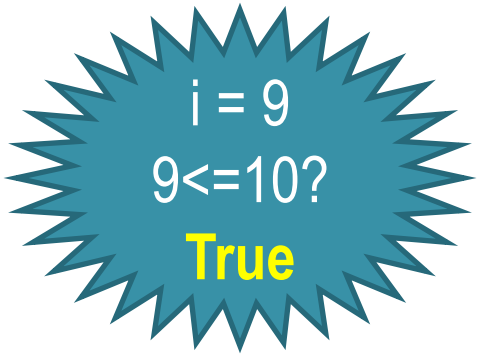


OUTPUT

1
2
3
4
5
6
7
8

Let us see the flow of loop

```
i=1
while i<=10:
        print(i)
        i+=1
```

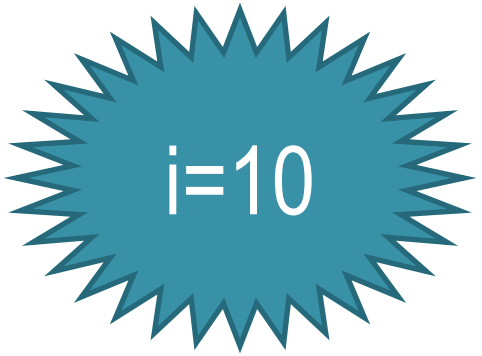


OUTPUT

1
2
3
4
5
6
7
8
9

Let us see the flow of loop

```
i=1
while i<=10:
        print(i)
        i+=1
```

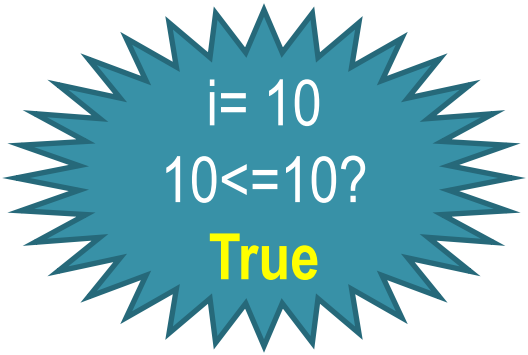


OUTPUT

```
1
2
3
4
5
6
7
8
9
```

Let us see the flow of loop

```
i=1
while i<=10:
        print(i)
        i+=1
```

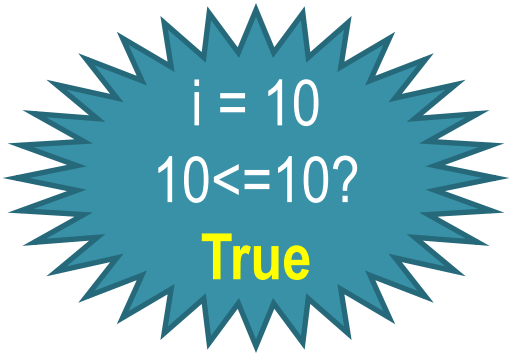


OUTPUT

1
2
3
4
5
6
7
8
9

Let us see the flow of loop

```
i=1
while i<=10:
        print(i)
        i+=1
```

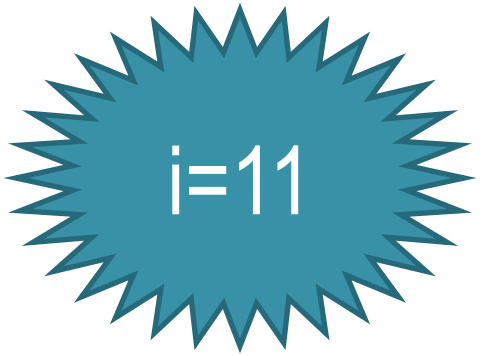


OUTPUT

1
2
3
4
5
6
7
8
9
10

Let us see the flow of loop

```
i=1
while i<=10:
        print(i)
        i+=1
```



OUTPUT

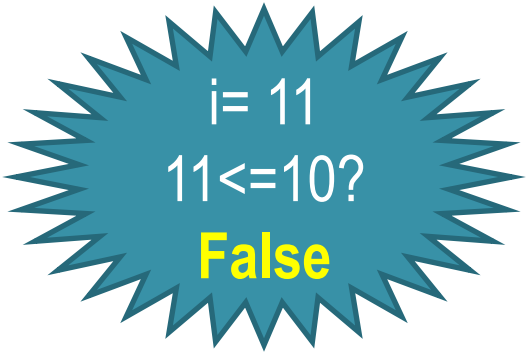
1
2
3
4
5
6
7
8
9
10

Let us see the flow of loop

```
i=1
while i<=10:
        print(i)
        i+=1
```

OUTPUT

1
2
3
4
5
6
7
8
9
10



Now the value of i is 11 and condition will return False so loop will terminate

for more updates visit: www.python4csip.com

Programs of while loop

- Convert all 'for' loop program with while loop
- WAP to enter any number and find its reverse
- WAP to enter any number and a digit and count how many times digit is in the number
- WAP to enter any number and check it is armstrong or not
- WAP to enter any number and check it is palindrome or not
- WAP to enter numbers as long as user wishes to enter and find the sum highest and lowest number entered by user

for more updates visit: www.python4csip.com

WAP to enter any number and find its reverse

```
num = int(input("Enter any number "))
rev = 0
num2=num
while num>0:
        rem = num % 10
        rev = rev * 10 + rem
        num //=10
print("Reverse of ",num2," is ",rev)
```

for more updates visit: www.python4csip.com

WAP to enter numbers as long as user wishes to enter and find the sum highest and lowest number entered by user

```
choice="y"
i=1
while choice=="y":
        num = int(input("Enter any number (>0)"))
        if i==1:
                low = num
                high = num
                i+=1
        else:
                if (high<num):
                        high=num
                if (num<low):
                        low = num
        choice=input("Enter continue or not (y/n)")
print("Highest value was ",high)
print("Lowest value was ",low)
```

for more updates visit: www.python4csip.com

Jump Statements – **break** & **continue**

break statement in python is used to **terminate the containing loop** for any given condition. Program resumes from the statement immediately after the loop

Continue statement in python is used to **skip the statements below continue statement** inside loop and **forces the loop to continue with next value**.

Example – break

```
for i in range(1,20):
        if i % 6 == 0:
                break
        print(i)
print("Loop Over")
```

The above code produces output

1
2
3
4
5
Loop Over

when the value of i reaches to 6 condition will becomes True and loop will end and message "Loop Over" will be printed

Example – continue

```
for i in range(1,20):
        if i % 6 == 0:
                continue
        print(i, end = ' ')
print("Loop Over")
```

The above code produces output

1 2 3 4 5 7 8 9 10 11 13 14 15 16 17 19

Loop Over

when the value of i becomes divisible from 6, condition will becomes True and loop will skip all the statement below continue and continues in loop with next value of iteration.

Loop .. else .. Statement

- Loop in python provides else clause with loop also which will execute when the loop terminates normally i.e. when the test condition fails in while loop or when last value is executed in for loop **but not when break terminates the loop**

for more updates visit: www.python4csip.com

Syntax of Python loop along with 'else' clause

for with "else"	while with "else"
for <var> in <seq>: statement 1 statement 2 **else:** statement(s)	while <test condition>: statement 1 statement 2 **else:** statement(s)

Example ("else" with while)

```
i=1
while i<=10:
        print(i)
        i+=1
else:
        print("Loop Over")
```

Output

```
1
2
3
4
5
6
7
8
9
10
Loop Over
```

Example ("else" with for)

```
names=["allahabad","lucknow","varanasi","kanpur","agra","ghaziabad"
,"mathura","meerut"]
city = input("Enter city to search: ")
for c in names:
        if c == city:
                print("City Found")
                break
else:
        print("Not found")
```

Output

Enter city to search : varanasi
City Found
Enter city to search : unnao
Not found

for more updates visit: www.python4csip.com

Example- program to enter numbers repeatedly and print their sum. The program ends when the user says no more to enter(normal termination) or program aborts when the number entered is less than zero

```
count = sum = 0
ans = 'y'
while ans=='y':
        num = int(input("Enter number :"))
        if num < 0:
                print("You entered number below zero,
Aborting...")
                break
        sum += num
        count+=1
        ans=input("Continue(y/n)")
else:
        print("You entered ", count, " numbers so far")
print("Sum of entered number is :",sum)
```

for more updates visit: www.python4csip.com

Example- program to enter numbers repeatedly and print their sum. The program ends when the user says no more to enter(normal termination) or program aborts when the number entered is less than zero - **OUTPUT**

```
Enter number :3
Continue(y/n)y
Enter number :6
Continue(y/n)y
Enter number :8
Continue(y/n)y
Enter number :-10
You entered number below zero, Aborting...
Sum of entered number is : 17
```

```
Enter number :10
Continue(y/n)y
Enter number :20
Continue(y/n)y
Enter number :50
Continue(y/n)n
You entered  3  numbers so far
Sum of entered number is : 80
```

for more updates visit: www.python4csip.com

Example- program to input number and tests if it is a prime number or not

```
num = int(input("Enter any number :"))
lim = num//2 + 1
for i in range(2,lim):
        rem = num % i
        if rem == 0:
                print(num," is not a prime number ")
                break
else:
        print(num," is a prime number ")
```

for more updates visit: www.python4csip.com

Example- program to input number and tests if it is a prime number or not - OUTPUT

Enter any number :23
23 is a prime number

Enter any number :36
23 is not a prime number

Generating Random numbers

- Python allows us to generate random number between any range
- To generate random number we need to import **random** library
- Syntax: (to generate random number)
  - random.randint(x,y)
  - it will generate random number between x to y
  - For e.g. : **random.randint(1,10)**
  - **Refer to code page no. 156**

Infinite loop and 'break'

- We can also execute infinite loop purposely by specifying an expression which always remains True.
- However in loop we can mention any condition to exit or terminate loop

```
a=2
while True:
        print(a)
        a*=2
        if a>100:
                break
```

for more updates visit: www.python4csip.com

Nested Loop (loop within loop)

- A loop may contain another loop in its body. This form of loop is called nested loop.
- In Nested loop, the inner loop will execute for each value of outer loop.
- For e.g. if the outer loop is of 4 times and inner loop is of 5 times then statement will execute 4 x 5 = 20 times

Syntax of Nested Loop

Nested Loop – FOR	Nested Loop – WHILE

for i in sequence:
  for j in sequence:
  Statement1
  Statement2
	while condition:
	  while condition:
	  Statement1
	  Statement2

Example – to print table of 2 to 10

```
for i in range(2,11):
        print("\nTable of ",i)
        for j in range(1,11):
                print(i,'x',j,'=',i*j)
```

```
Table of  2
2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8
2 x 5 = 10
2 x 6 = 12
2 x 7 = 14
2 x 8 = 16
2 x 9 = 18
2 x 10 = 20

Table of  3
3 x 1 = 3
3 x 2 = 6
3 x 3 = 9
3 x 4 = 12
3 x 5 = 15
3 x 6 = 18
3 x 7 = 21
3 x 8 = 24
3 x 9 = 27
3 x 10 = 30
```

4 to 8

```
Table of  9
9 x 1 = 9
9 x 2 = 18
9 x 3 = 27
9 x 4 = 36
9 x 5 = 45
9 x 6 = 54
9 x 7 = 63
9 x 8 = 72
9 x 9 = 81
9 x 10 = 90

Table of  10
10 x 1 = 10
10 x 2 = 20
10 x 3 = 30
10 x 4 = 40
10 x 5 = 50
10 x 6 = 60
10 x 7 = 70
10 x 8 = 80
10 x 9 = 90
10 x 10 = 100
```

Example – Program to print the following patterns

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

```
1
12
123
1234
12345
```

```
for i in range(5,0,-1):
        for j in range(i,0,-1):
                print(j,end=' ')
        print()
```

```
54321
4321
321
21
1
```

Write a program to print the following pyramid pattern

```
    1
   121
  12321
 1234321
123454321
```

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