```
———— *
——— *  *
— *  *  *
-*  *  *  *
*  *  *  *  *
```

```
space=4
for i in range(1,6):
  for k in range(space):
      print(" ",end="")
  for j in range(1,i+1):
     print("*",end=" ")
  space=space-1
  print()
```

```
*  *  *  *  *
  *  *  *  *
    *  *  *
      *  *
        *
```

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

```
        1
      1   1
    1  2  1
  1  2  3  1
1 2  3 4  1
```

```
space=4
for i in range(1,6):
  for k in range(space):
    print(" ",end='')
  for j in range(1,i+1):
     if j==1 or j==i:
        print("1",end=" ")
     else:
        print(j,end=" ")
  print()
  space=space-1
```

```
1  2  3 4  1
  1 2 3 1
    1 2 1
      1 1
       1
```

```
space=0
for i in range(5,0,-1):
   for k in range(space):
     print(" ",end='')
   for j in range(1,i+1):
     if j==1 or j==i:
        print("1",end=' ')
     else:
        print(j,end=' ')
    space=space+1
    print()
```

5					5
4				5	4
3			5	4	3
2		5	4	3	2
1	5	4	3	2	1

```
space=4
for i in range(5,0,-1):
  for k in range(space):
    print(" ",end=' ')
  for j in range(5,0,-1):
      if j>=i:
         print(j,end=' ')
  space=space-1
  print()
```

Nested While

While loop inside while loop is called nested while
While loop is used to repeat block of statements until given condition
Defining while as a statement within while is called nested while

Syntax:

```
while <condition>:
   while <condition>:
      statement-1
      statement-2
```

Example	**Output**

# Write a program to generate tables from 1 to 10 	
# nested while 	
 	
num=1
while num<=10: # Outer Looping
   i=1
   while i<=10: # Inner Looping
      print(f'{num}x{i}={num*i}')
      i=i+1
 	
   num=num+1
	1x1=1
	1x2=2
	1x3=3
	1x4=4
	1x5=5
	1x6=6
	1x7=7
	1x8=8
	1x9=9
	1x10=10
	2x1=2
	2x2=4
	2x3=6
	…

# Write a program to generate armstrong numbers from 100 to 999	**Output** 153 370 371

```
num=100
while num<=999:
   num1=num
   s=0
   while num1>0:
      d=num1%10
      s=s+(d**3)
      num1=num1//10
   if s==num:
      print(num)
   num=num+1
```

```
407
```

```
# Write a program to generate armstrong numbers from
# 1000 to 9999

num=1000
while num<=9999:
   s=0
   num1=num
   while num1>0:
      d=num1%10
      s=s+(d**4)
      num1=num1//10
   if s==num:
      print(num)
   num=num+1
```

Output

```
1634
8208
9474
```

```
# Write a program to generate factorials of
# numbers from 1-5

num=1
```

```
1--1
2--2
3--6
4--24
5—120
```

```
while num<=5: # generating numbers (1-5)
   fact=1
   i=1
   while i<=num:
      fact=fact*i
      i=i+1
   print(f'{num}--{fact}')
   num=num+1
```

```
num=5
fact=1
i=1
while i<=num:
   fact=fact*i
   print(f'{i}-->{fact}')
   i=i+1
```

Output

```
1-->1
2-->2
3-->6
4-->24
5-->120
```

Branching statements

Python support the following branching statements

1. break
2. continue
3. Return (Functions)

Branching statements are used to control the execution of looping statements (while loop, for loop)

break

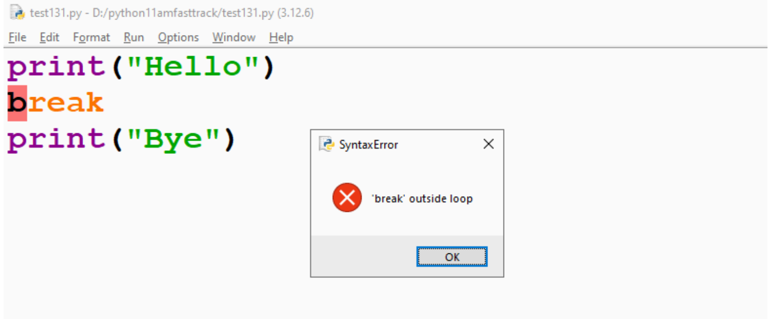
“break” is keyword which represents branching statement
This keyword is used inside while loop or for loop

This keyword is sued to terminate execution of while loop or for loop in between.

Syntax:

```
for variable-name in iterable/collection:
      statement-1
      statement-1
      break
```

```
while condition:
     statement-1
     statement-2
     break
```



test131.py - D:/python11amfasttrack/test131.py (3.12.6)
File Edit Format Run Options Window Help
SyntaxError
OK

```
print("Hello")
break
print("Bye")
```

'break' outside loop

Example:

```
for i in range(1,11):
   print("Hello")
   break
```

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

Output

```
Hello
While
```

Example:

```
# Login

while True:
    uname=input("UserName :")
    pwd=input("Password :")
    if uname=="naresh" and pwd=="n123":
        print("Welcome to My Application")
        break
    else:
        print("Invalid username or password")
```

Output

```
UserName :abc
Password :xyz
Invalid username or password
UserName :naresh
Password :n321
Invalid username or password
UserName :naresh
Password :n123
Welcome to My Application
```

Example:

```
# Write a program to find input number is prime or not

num=int(input("Enter any number "))
c=0
for i in range(1,num+1):
    if num%i==0:
        c=c+1
    if c>2:
        break

if c==2:
    print(f'{num} prime')
else:
    print(f'{num} not a prime')
```

Output

```
Enter any number 8
8 not a prime
```

continue

“continue” is keyword which represents branching statement
This statement also used inside while loop or for loop
Continue keyword move the execution control to the beginning the looping statement (while or for) (OR) continue the loop

Syntax:

```
while <condition>:
    Statement-1
     Statement-2
    Continue
```

Syntax:

```
for variable-name in iterable:
    Statement-1
    Statement-2
    Continue
```

Example:

```
for i in range(1,6):
   print("Hello")
   continue
   print("Bye")
```

Output

```
Hello
Hello
Hello
Hello
Hello
```