



AKTU B.Tech II-Year

ONE SHOT Revision

Common to All Branches



Python Programming

Unit-2 By Pragya Ma'am

Python Program Flow Control Conditional blocks

PYTHON PROGRAMMING

AKTU UNIT-2 Syllabus

Python Program Flow Control Conditional blocks: if, else and else if, Simple for loops in python, For loop using ranges, string, list and dictionaries. Use of while loops in python, Loop manipulation using pass, continue, break and else. Programming using Python conditional and loop blocks.

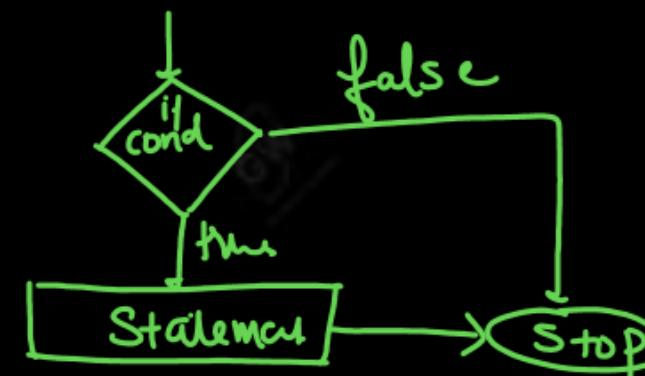
➤ A conditional statement as the name suggests itself, is used to handle conditions in your program.

➤ if statement

➤ Syntax

if expression:

statement



main.py

```

1 num = int(input("enter the number:"))
2 if num%2 == 0:
3     print("The Given number is an even number")
4
  
```

enter the number:6
The Given number is an even number

➤ If-else statement

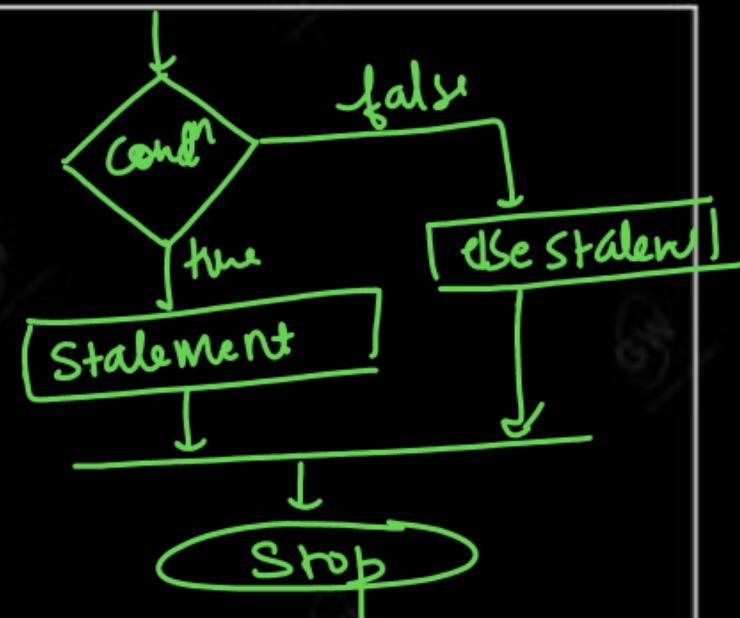
➤ Syntax

if condition:

#block of statements

else:

#another block of statements (else-block)
(Even Odd)



main.py

```

1 num = int(input("enter the number:"))
2 if num%2 == 0:
3     print("The Given number is an even number")
4 else:
5     print("the given number is not even")
  
```

enter the number:7
the given number is not even

Conditional statement

➤ elif statement

if expression 1:

block of statements

elif expression 2:

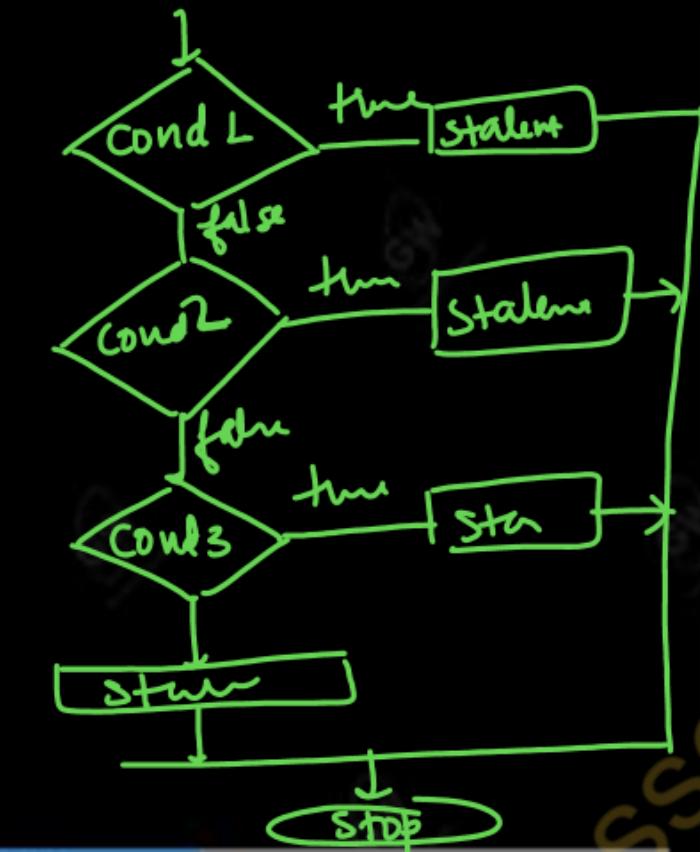
block of statements

elif expression 3:

block of statement

else:

#statement



```
main.py
1 n=2
2 if n>3:
3     print("hi")
4 elif n==2:
5     print ("hello")
```

main.py

```
1 a=int(input("enter the first string\n"))
2 b=int(input("enter the second string\n"))
3 c=int(input("enter the third string\n"))
4 if a>b and a>c:
5     print(" a is largest")
6 elif b>a and b>c:
7     print(" b is largest")
8 else:
9     print(" c is largest")
```

Largest Number among three.

▼ ⌂ ⌂ ⌂ ⌂

enter the first string

5

enter the second string

6

enter the third string

7

c is largest

> Nested if else

if condition 1:

 if condition2:

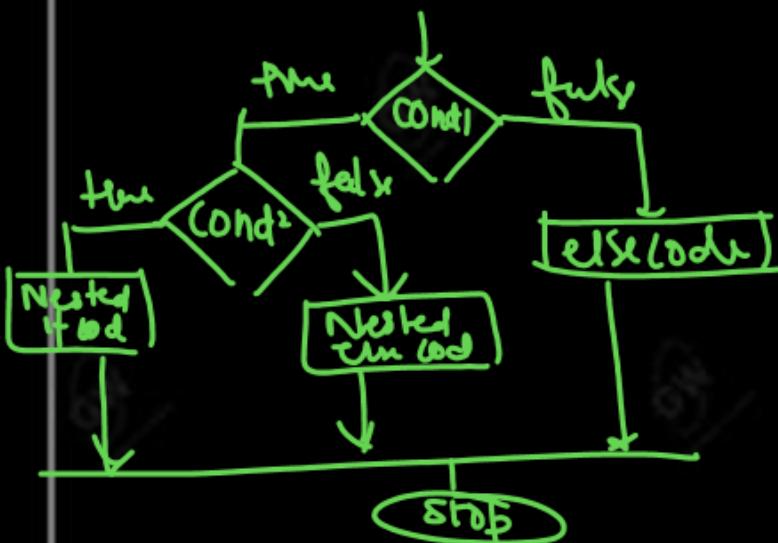
 #nested if code

 else:

 #nested else code

else:

 #else code



```
1 a=int(input("enter the value of a\n"))
2 b=int(input("enter the value of b\n"))
3 c=int(input("enter the value of c\n"))
4 if a>b:
5   if a>c:
6     print(a, "is the largest")
7   else:
8     print(c,"is the largest")
9
10 else:
11   if b>c:
12     print(b,"is the largest")
13   else:
14     print(c, "is the largest")
15
```

write a program to check whether a year is leap year or not

main.py

```
1 x=int(input("enter the year\n"))
2 if x%400==0:
3     print(x," is leap year")
4 elif x%100==0:
5     print(x,"is not a leap year")
6 elif x%4==0:
7     print(x,"is leap year")
8 else:
9     print(x,"is not leap year")
```



```
enter the year
1600
1600 is leap year
```

➤ Write program to make calculator

main.py

```
1 x=float(input("enter the first number\n"))
2 y=float(input("enter the second number\n"))
3 z=input("enter the operator{+, -, *, /, //}\n")
4 if z=="+":
5     print("the sum of two number", x+y)
6 elif z=="-":
7     print("subtraction of x-y number",x-y)
8 elif z=="*":
9     print("multiplication of two number",x*y)
10 elif z=="/":
11     print("division of x/y",x/y)
12 elif z=="//":
13     print("floor division of x//y",x//y)
14 else:
15     print("enter the wrong choice")
16
17
```

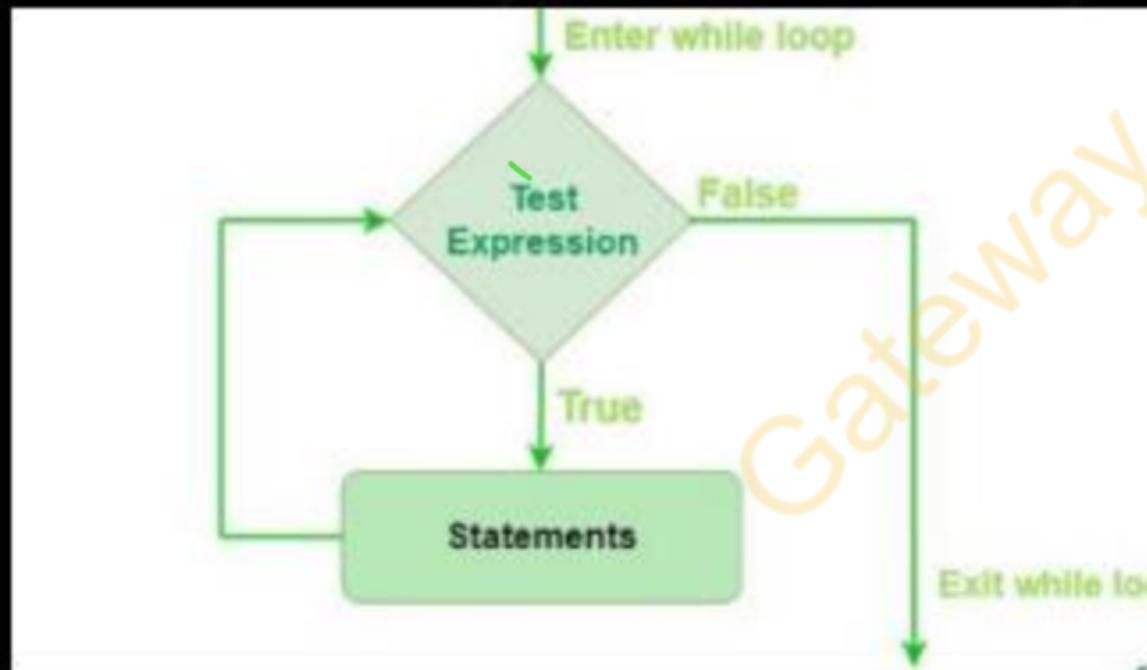
- Loops is used to execute a block of statements repeatedly until a given condition is satisfied.

1 while loop

Syntax

while expression:

statement(s)

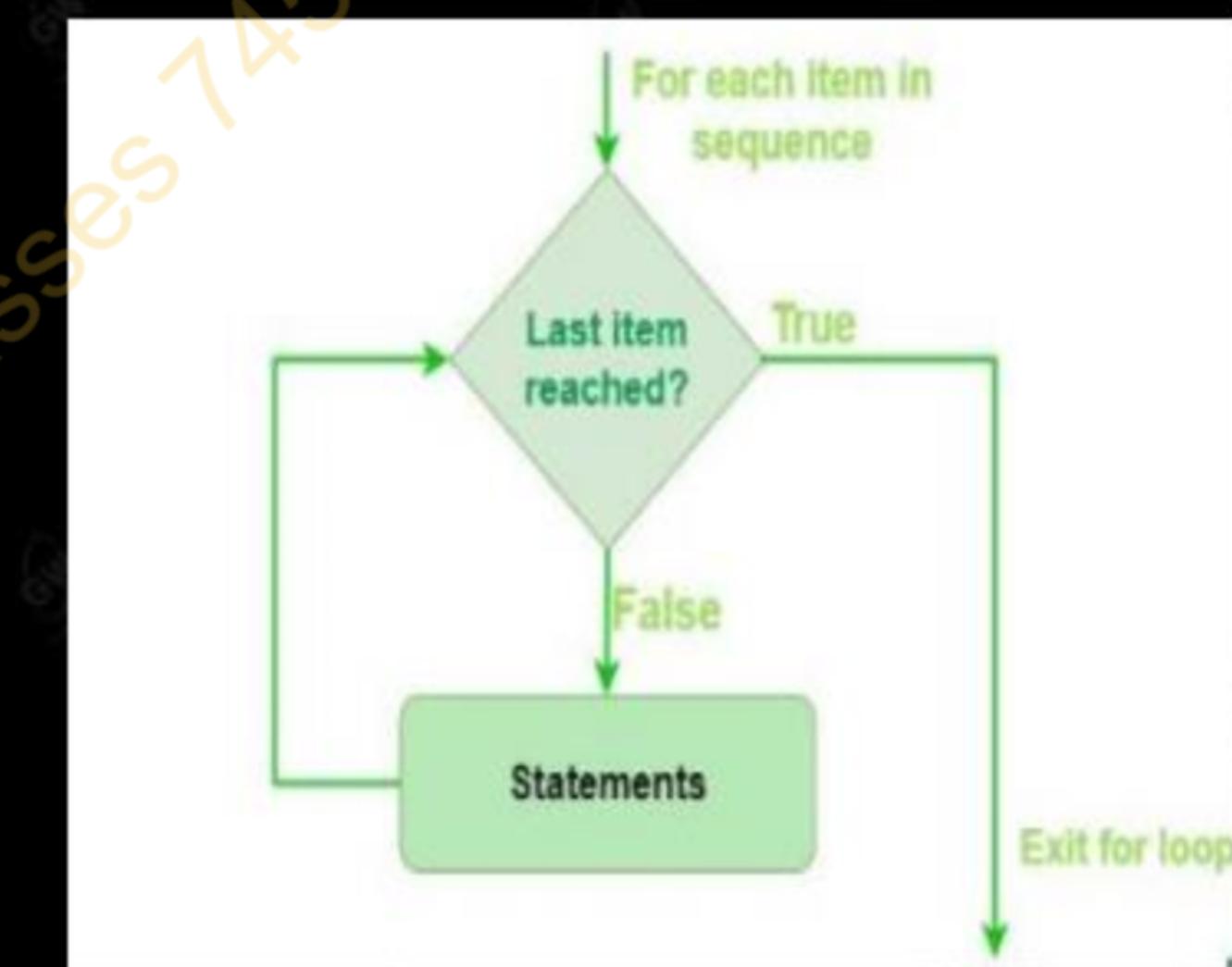


2 For loop

Syntax

for var in iterable:

statements



- The Python `range()` function returns a sequence of numbers, in a given range.

- range (stop)

```
1 for i in range(6):
2     print(i, end=" ")
3
```

0 1 2 3 4 5

- range (start, stop)

```
1
2
3 for i in range(5, 20):
4     print(i, end=" ")
5
```

5 6 7 8 9 10 11 12 13 14 15 16 17 18 19

range (start, stop, step)

```
1 for i in range(0, 30, 4):
2     print(i, end=" ")
3
```

0 4 8 12 16 20 24 28

Write a for loop that prints number from 0 to 57 using range function

for i in range(58)
print(i, end=" ")

```
1 for i in range(58):
2     print(i)
3
```

write a program to check whether a number is Armstrong or not

```
main.py
1 n=int(input("enter the number"))
2 sum = 0
3 num=n
4 while n> 0:
5     digit = n% 10
6     sum += digit ** 3
7     n /= 10
8 if num==sum:
9     print("armstrong")
10 else:
11     print("not armstrong")
12
```

enter the number153
armstrong

➤ write a program to find out whether a number is palindrome or not

```
main.py
1 n=int(input("enter n digit number\n"))
2 num=n
3 rev=0
4 while n>0:
5     d=n%10
6     rev=rev*10+d
7     n=n//10
8 print("reverse",rev)
9 if rev== num:
10    print("palindrome")
11 else:
12    print("not palindrome")
```

enter n digit number
121
reverse 121
palindrome

GW Write a program to find out the fibonnaci series upto n term

main.py

```
1 nterms = int(input("How many terms? "))
2 n1, n2 = 0, 1
3 count = 0
4 if nterms <= 0:
5     print("Please enter a positive integer")
6 elif nterms == 1:
7     print(n1)
8 else:
9     print("Fibonacci sequence:")
10    print(n1,n2)
11    for i in range(3,nterms+1):
12        nth = n1 + n2
13        print(nth)
14        n1 = n2
15        n2 = nth
```

How many terms? 3

Fibonacci sequence:

0

1

1

0,1 1 2,3 5,8

GW Write a program to find out the factorial of a number

main.py

```
1 num=int(input("enter the number\n"))
2 factorial =1
3 if num < 0:
4     print("Sorry, factorial does not exist for negative numbers")
5 elif num == 0:
6     print("The factorial of 0 is 1")
7 else:
8     for i in range(1,num + 1):
9         factorial = factorial*i
10    print("The factorial of",num,"is",factorial)
```

v , P ⚙ S

input

enter the number

4

The factorial of 4 is 24

num factorial
X 2, 3, 4, 5, 6, 7, 8, 9

Write a program to check whether a number is prime or not

```
main.py
1 num = int(input("Enter a number: "))
2 flag = False
3 if num == 1:
4     print(num, "is not a prime number")
5 elif num>1:
6     for i in range(2, num):
7         if (num % i) == 0:
8             flag = True
9             break
10    if flag:
11        print(num, "is not a prime number")
12    else:
13        print(num, "is a prime number")
14 else:
15    print("enter positive number")
```

input
Enter a number: 7
7 is a prime number

Print the following pattern

```
main.py
1 rows=int(input("enter the number of rows"))
2 for i in range(rows):
3     for j in range(i+1):
4         print("*",end="")
5     print("\n")
```

enter the number of rows5
*
**

Write a program to print the reverse of a number

```
main.py
1 n=int(input("enter n digit number\n"))
2 num=n
3 rev=0
4 while n>0:
5     d=n%10
6     rev=rev*10+d
7     n=n//10
8 print("reverse",rev)
9
```

```
enter n digit number
1214
reverse 4121
```

What will be the output

```
1 a=0
2 b=2
3 c=3
4 x= a |n| c
5 print(c)
```

```
3 7455 96184
```

What will be the Output

```
1 i=0
2 while i<3:
3     i+=1
4 else:
5     print(0)
```

Print(0) of else is
not at collect
Indendation

```
1 i=0
2 while i<3:
3     print(i)
4     i+=1
5 else:
6     print(0)
```

```
0
1
2
0
```

- 1 start
 - 2 enter the lower range upper range
 - 3 Take one number one by one from the range
 - 4 check number is greater than 1 if yes then go to set 5
 - 5 check whether that number is divisible by (2 to less than that number)
 - 6 if yes then break check for another number else print that number
- Repeat the process for all the number
- 7 stop

Write a program to print the prime number in range

```
main.py
1 lower = int(input("enter the lower range"))
2 upper = int(input ("enter the upper range"))
3 print("Prime numbers between", lower, "and", upper, "are:")
4 for num in range(lower, upper + 1):
5     if num > 1:
6         for i in range(2, num):
7             if (num % i) == 0:
8                 break
9     else:
10        print(num)

enter the lower range2
enter the upper range7
Prime numbers between 2 and 7 are:
2
3
5
7
```



Print the following pattern

```
1 rows=int(input("enter the number of rows"))
2 for i in range(rows,0,-1):
3     for j in range(0,i):
4         print("*",end="")
5     print("\n")
```

v , F S input

enter the number of rows5

**

*

```
1 rows = int(input("Enter number of rows: "))
2
3 k = 0
4
5 for i in range(1, rows+1):
6     for space in range(1, (rows-i)+1):
7         print(end=" ")
8
9     while k!=(2*i-1):
10        print("* ", end="")
11        k += 1
12    k=0
13    print()
```

v , F S input

Enter number of rows: 4

*
* * *
* * * * *
* * * * *

the break statement is used to terminate the loop immediately when it is encountered.

The syntax of the break statement is:

break

```
1 for i in range(5):  
2     if i == 3:  
3         break  
4     print(i)
```



0

1

2

```
for val in sequence:  
    # code  
    if condition:  
        break  
    # code  
while condition:  
    # code  
    if condition:  
        break  
    # code
```

The continue statement is used to skip the current iteration of the loop and the control flow of the program goes to the next iteration.

The syntax Continue;

```
1 for i in range(5):  
2     if i == 3:  
3         continue  
4     print(i)
```

0
1
2
3
4

for val in sequence:
 # code
 if condition:
 continue

code

while condition:
 # code
 if condition:
 continue

code

Continue example

main.py

```
1 num = 0
2 while num < 10:
3     num += 1
4     if (num % 2) == 0:
5         continue
6     print(num)
```

```
1 ✓
3 ✓
5 ✓
7 ✓
9 ✓
```

Break example

```
1 i = 1
2 while i <= 10:
3     print('6 *',(i),'=',6 * i)
4     if i >= 5:
5         break
6     i = i + 1
```

```
6 * 1 = 6
6 * 2 = 12
6 * 3 = 18
6 * 4 = 24
6 * 5 = 30
```

- python pass statement is a null statement. But the difference between pass and comment is that comment is ignored by the interpreter whereas pass is not ignored.

```
1 a = 10
2 b = 20
3 if(a<b):
4     pass
5 else:
6     print("b < a")
```

Program finished with exit code 0
Press ENTER to exit console.

What is pass statement in Python?

When the user does not know what code to write, So user simply places a pass at that line. Sometimes, the pass is used when the user doesn't want any code to execute. So users can simply place a pass where empty code is not allowed, like in loops, function definitions, class definitions, or in if statements. So using a pass statement user avoids this error.

```
1 li =['a', 'b', 'c', 'd']
2 for i in li:
3     if(i == 'a'):
4         pass
5     else:
6         print(i)
```

b
c
d

Write a program to check whether entered number is Fibonacci number or not

0, 1, 1, 2, 3, 5

```
main.py
1 import math
2 def isPerfectSquare(x):
3     s = int(math.sqrt(x))
4     return s*s == x
5 def isFibonacci(n):
6     return isPerfectSquare(5*n*n + 4) or isPerfectSquare(5*n*n - 4)
7 i=int(input("enter the number"))
8 if (isFibonacci(i) == True):
9     print(i,"is a Fibonacci Number")
10 else:
11     print(i,"is a not Fibonacci Number ")
12
```

```
enter the number3
3 is a Fibonacci Number
```

Write a program to check whether entered number is Fibonacci number or not in range

```
main.py
1 import math
2 def isPerfectSquare(x):
3     s = int(math.sqrt(x))
4     return s*s == x
5 def isFibonacci(n):
6     return isPerfectSquare(5*n*n + 4) or isPerfectSquare(5*n*n - 4)
7 a=int(input("enter the terms"))
8 for i in range(1,a):
9     if (isFibonacci(i) == True):
10         print( i,"is a Fibonacci Number")
11     else:
12         print( i,"is a not Fibonacci Number ")
13
```

enter the terms6

1 is a Fibonacci Number

2 is a Fibonacci Number

3 is a Fibonacci Number

4 is a not Fibonacci Number

5 is a Fibonacci Number

DIFFERENCE BETWEEN

for loop

For loop is used to iterate over a sequence of items.

For loops are designed for iterating over a sequence of items. Eg. list, tuple, etc.

For loop require a sequence to iterate over.

For loop is typically used for iterating over a fixed sequence of items

While loop

While loop is used to repeatedly execute a block of statements while a condition is true.

While loop is used when the number of iterations is not known in advance or when we want to repeat a block of code until a certain condition is met.

While the loop requires an initial condition that is tested at the beginning of the loop.

While loop is used for more complex control flow situations.

Write a program to convert time from 12 hour to 24 hour format

main.py

```
1 def convert24(str1):
2     if str1[-2:] == "AM" and str1[:2] == "12":
3         return "00" + str1[2:-2]
4     elif str1[-2:] == "AM":
5         return str1[:-2]
6     elif str1[-2:] == "PM" and str1[:2] == "12":
7         return str1[:-2]
8     else:
9         return str(int(str1[:2]) + 12) + str1[2:8]
10 print(convert24("08:05:45 PM"))
11
12
```

20:05:45

input

```
n = 5
# upper triangle
for i in range(n):
    for j in range(i + 1):
        print('*', end="")
    print()
# Lower triangles
for i in range(n):
    for j in range(n - i - 1):
        print('*', end="")
    print()
```

Thank You

Gateway Classes 7455 9612 84