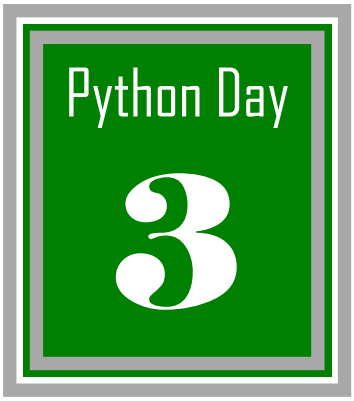
King Khalid University
College of Computer Science



Programming in Python
(Module No.1)

Decision and Control Structure

Prepared by

Eng. Abdelmoty Mahmoud Mohamed Ahmed

Lecturer in Computer engineering department, College of Computer Science, King Khalid University

M.Sc. in Computers and systems engineering department

Faculty of Engineering in Cairo

Al-Azhar University

E-mail: amoate@kku.edu.sa
abd2005moty@yahoo.com
Mobile: 00966504538402

2018-2019

What are Selections?

A program can decide which statements to execute based on a condition. In this lab we will construct program that allow statements to be optionally executed, depending on the selection condition. Selections are statements that decide whether specific statements to be executed or not.

Boolean Data Types, and Expressions

In Python, you can declare a variable of Boolean type. A variable that holds a Boolean value is known as a Boolean variable. The Boolean data type is used to represent Boolean values. A Boolean variable can hold one of the two values: True or False.

We can directly declare a variable and give it a Boolean value as the following example shows:

```
b1 = True
b2 = False
```

True and False are literals. They are reserved words and cannot be used as identifiers in a program.

Also, we can declare a variable and give it the value using Boolean expressions. Boolean expression is an expression that computes to True or false. The following example computes the value of the Boolean expression and stores the value in the variable b:

```
radius = 1
b = radius > 0
```

b now has the Boolean value True.

Python provides six comparison operators, also known as relational operators, shown in the table below, which can be used to compare two values and return a Boolean value.

Python Operator	*Mathematics Symbol*	*Name*	*Example* **(radius is 5)**	*Result*
`<`	<	less than	`radius < 0`	`False`
`<=`	≤	less than or equal to	`radius <= 0`	`False`
`>`	>	greater than	`radius > 0`	`True`
`>=`	≥	greater than or equal to	`radius >= 0`	`True`
`==`	=	equal to	`radius == 0`	`False`
`!=`	≠	not equal to	`radius != 0`	`True`

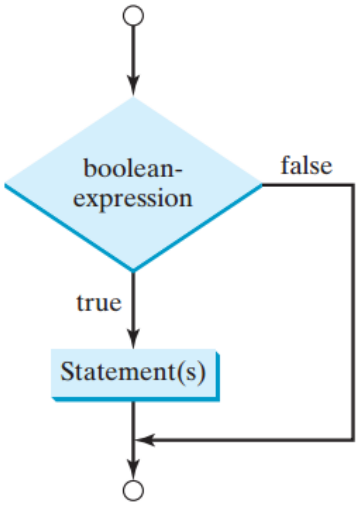
if Statement

A one-way if statement executes the statements if the condition is true. If the condition is false nothing will be executed. The syntax for a one-way if statement is:

```
if boolean-expression:
    statement(s) # Note that the statement(s) must be indented
```

The statement(s) must be indented at least one space to the right of the if keyword and each statement must be indented using the same number of spaces.

The following flowchart illustrates how Python executes the syntax of an if statement.



The following program computes the area of a circle only if the radius is positive.

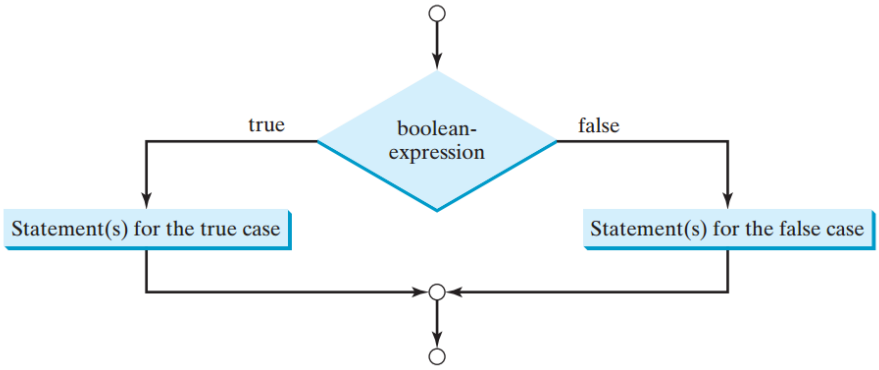
```
radius = 10
if radius >= 0:
    area = radius * radius * 3.14159
    print("The area for the circle is", area)
```

Two-Way if-else Statements

A one-way if statement takes an action if the specified condition is True. If the condition is False, nothing is done. But what if you want to take one or more alternative actions when the condition is False? You can use a two-way if-else statement. A two-way if-else statement decides which statements to execute based on whether the condition is true or false.

The syntax for a two-way if-else statement:

```
if boolean-expression:
    statement(s)-for-the-true-case
else:
    statement(s)-for-the-false-case
```



The following program computes the area of a circle if the radius is positive, if the radius is negative it prints "Negative input".

```
radius = 10
if radius >= 0:
    area = radius * radius * 3.14159
    print("The area for the circle is", area)
else:
    print("Negative input")
```

Multi-Way if-elif-else Statements

The if-elif-else statement allows you to check multiple expressions for True and execute a block of code as soon as one of the conditions evaluates to True.

The syntax for multi-way if-elif-else statement:

```
if boolean-expression1:
    statements(1)
elif boolean-expression2:
    statements(2)
elif boolean-expression3:
    statements(3)
else :
    statements(4)
```

The following program checks whether an integer is zero, positive or negative:

```
a = 24
if a == 0:
    print "a is zero"
elif a < 0:
    print "a is negative"
else:
    print "a is positive"
```

Logical Operators

The logical operators not, and, and or can be used to create a composite condition. Sometimes, a combination of several conditions determines whether a statement is executed. You can use logical operators to combine these conditions to form a compound expression. The following table shows the logical operators in Python:

Operator	Description
and	If **both** the operands are **true** then condition becomes **true**
or	If **any** of the two operands is **true** then condition becomes **true**
not	Used to **reverse** the logical state

The following table shows the possible states:

e1	e2	e1 and e2	e1 or e2	not e1
True	True	True	True	False
True	False	False	True	False
False	True	False	True	True
False	False	False	False	True

Ex: Write a Python program that prompts the user to enter a number and checks whether the number is in the period [0,10] or not. Then print the result.

Solution:

```
value = input("Enter an integer value:")
if value >= 0 and value <= 10:
    print("In range")
else:
    print "Out of range"
print("Done")
```

Conditional Expressions

The following statement assigns 1 to y if x is greater than 0, and -1 to y if x is less than or equal to 0.

```
if x > 0:
    y = 1
else:
    y = -1
```

Alternatively, we can use a conditional expression to achieve the same result.

```
y = 1 if x > 0 else -1
```

The syntax for conditional expressions is as follows:

```
expression1 if boolean-expression else expression2
```

Ex: Write a Python program that computes the absolute value of a number entered by the user.

First solution using if-else statement:

```
n = input("Enter a number: ")
if n < 0:
    absValue = -n
else:
    absValue = n
print (absValue)
```

Other solution using conditional expression:

```
n = input("Enter a number: ")
absValue = -n if n < 0 else n
print (absValue)
```

Lab Work

Ex1: Write a Python Program that checks whether a number (entered by user) is even or odd.

Solution:

```
n = input("Enter a number: ")
if n % 2 == 0:
    print("Even number")
else:
    print("Odd number")
```

Other solution:

```
n = input("Enter a number: ")
print("Even number" if n % 2 == 0 else "Odd number")
```

Ex2: Write a program that requests from user to enter an integer and checks whether the number is divisible by both 5 and 6, or neither of them, or just one of them.

```
num = input("Enter number: ")
if num % 5 == 0 and num % 6 == 0:
    print(num, "is divisible by both 5 and 6")
elif num % 5 == 0 or num % 6 == 0:
    print(num, "is divisible by 5 or 6, but not both")
else:
    print(num, "is not divisible by either 5 or 6")
```

Ex3: Write a Python program that reads the month of birth from the user and prints the name for that month.

Solution:

```
month = input("Enter your month of birth: ")
if month == 1:
    print ("January")
elif month == 2:
    print("February")
elif month == 3:
    print("March")
elif month == 4:
    print("April")
elif month == 5:
    print("May")
elif month == 6:
    print("June")
elif month == 7:
    print("July")
elif month == 8:
    print("August")
elif month == 9:
    print("September")
elif month == 10:
    print("October")
elif month == 11:
    print("November")
elif month == 12:
    print("December")
else:
    print("Invalid input")
```

Homework

1. What is the output of the following code in (a) and (b) if number is 30, then if number is 35.

   a)

```
if number % 2 == 0:
    print(number, "is even.")
print(number, "is odd.")
```

   b)

```
if number % 2 == 0:
    print(number, "is even.")
else:
    print(number, "is odd.")
```

2. Rewrite the following if statement using a conditional expression:

```
if age >= 16:
    ticketPrice = 20
else:
    ticketPrice = 10
```

3. Rewrite the following conditional expressions using if-else statements:
   1. `score = 3 * scale if x > 10 else 4 * scale`
   2. `print(i if number % 3 == 0 else j)`

4. Write a Python program that reads username and password from the user and prints "successful login" if the username and password are correct, otherwise, it prints "username or password is wrong".

   Here is a sample run:

```
Enter username: "john"
Enter password: 12345
successful login

Process finished with exit code 0
```

```
Enter username: "john"
Enter password: 0000
username or password is wrong

Process finished with exit code 0
```

5. Write a Python program that uses an if statement to find the smallest of three given integers. (The user should enter the three numbers).

Good Luck

Exercises:

1) Write a Python program that requests three integer values from the user. It then prints one of two things: if any of the values entered are duplicates, it prints "DUPLICATES"; otherwise, it prints "ALL UNIQUE".

2) Write a Python program that use an if statement to find the smallest of three given integers without using the min function

3) Write a Python program that sorts three integers. The integers are entered from user using input function.

   For example if the user enters:

   7, 3, 5 then the output should be 3<=5 <=7.

   6, 2, 6 then the output should be 2<=6 <=6

Exercises:

1) Write a Python program that prompts the user to enter an integer for today's day of the week (Saturday is 0, Sunday is 1 ,. . . and Friday is 6). Also prompt the user to enter the number of days after today for a future day and display the future day of the week. Here is a sample run:

```
C:\Python27\python.exe C:/Users/ibraheem/Pycharm
Enter today's day number:1
Enter the number of days after today:3
Today is Sunday and the future day is Wednesday
```

```
C:\Python27\python.exe C:/Users/ibraheem/PycharmPr
Enter today's day number:4
Enter the number of days after today:15
Today is Wednesday and the future day is Thursday
```

Solution

```
x = input("Enter today's day number:")
y = input("Enter the number of days after today:")
days = ("Saturday", "Sunday", "Monday", "Tuesday", "Wednesday", "Thursday",
"Friday")

print "Today is %s and the future day is %s" % (days[x], days[(x + y) % 7])
```

.

2) Write a Python program that requests three integer values from the user. It then prints one of two things: if any of the values entered are duplicates, it prints "*DUPLICATES*"; otherwise, it prints "*ALL UNIQUE*".

Solution

```
x = input("First number : ")
y = input("Second number : ")
z = input("Third number : ")

if x == y or x == z or y == z:
    print "DUPLICATES"
else:
    print "ALL UNIQUE"
```

3) Write a Python program that use an if statement to find the largest of three given integers without using the max() function.

For example if user enters: 25, 70, and 16 , then output should be: 70

Solution

```
x = input("First number : ")
y = input("Second number : ")
z = input("Third number : ")

if x > y and x > z:
    max = x
elif y > x and y > z:
    max = y
else:
    max = z
print "The largest  is ", max
```

4) Write a Python program that sorts three integers. The integers are entered from user using input function.

For example if the user enters:

7, 3, 5 then the output should be 3 <=5 <=7.
6, 2, 6 then the output should be 2<= 6 <=6

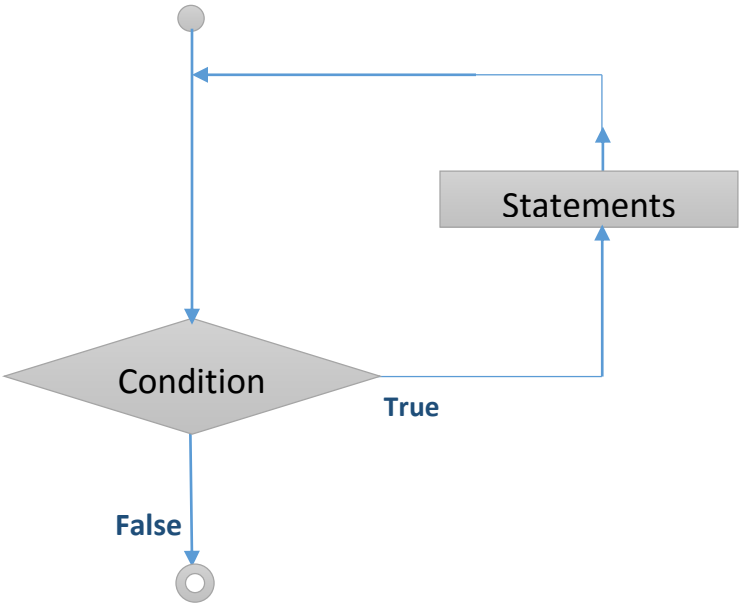
Solution

```
x = input("first number : ")
y = input("second number : ")
z = input("third number : ")

if x > y and x > z:
    max = x
    if y > z:
        mid = y
        min = z
    else:
        mid = z
        min = y
elif y > x and y > z:
    max = y
    if x > z:
        mid = x
        min = z
    else:
        mid = z
        min = x
else:
    max = z
    if y > x:
        mid = y
        min = x
    else:
        mid = x
        min = y
print min, "<= ", mid, " <=", max
```

Python Day No.3 Notes

A loop statement allows us to execute a statement or group of statements multiple times.

The following diagram illustrates a loop statement.



Types of loops:

1. While loop
2. For loop

➢ While loop :

Repeats a statement or group of statements while a given condition is True. It tests the condition before executing the loop body.

Syntax of while loop
`while Boolean Expression:` `  statement(s)`

Note:

All the statements **indented** by the same number of character spaces after a programming construct are considered to be part of a single block of code.

Example 1

```
count = 0
while count < 5:
   print 'The count is:', count
   count = count + 1

print "Done"
```

while block

```
C:\Python27\python.exe (
The count is: 0
The count is: 1
The count is: 2
The count is: 3
The count is: 4
Done
```

When the condition is tested and the result is false, the loop body will be skipped and the first statement after the while loop will be executed.

Using else Statement Loops:

If the else statement is used with a while loop, the else statement is executed when the condition becomes false.

Example 2

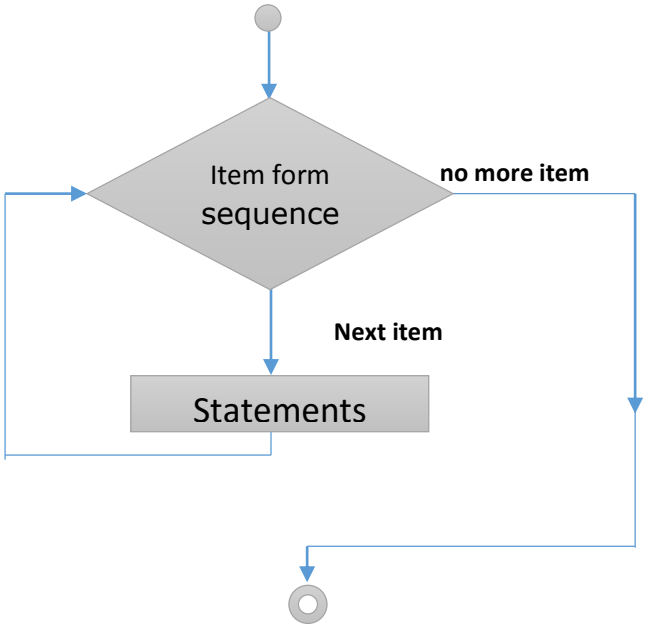
```
count = 0
while count < 5:
   print 'The count is:', count
   count = count + 1

else:
    print "Finish..."
```

➢ for loop:

It has the ability to iterate over the items of any **sequence**, such as a **list**, **tuple** or a **string**.

Syntax of for loop

```
for item in sequence:
    statements(s)
```



Example 3

```
items=["Orange","Apple","Banana"]
for item in items:
    print item
```

Output

```
C:\Python27\python.exe C:/Users/ib
Orange
Apple
Banana

Process finished with exit code 0
```

Example 4

```
str= "Python"
for char in str:
    print char
```

Output

```
C:\Python27\python.exe C:/Users/ib
P
y
t
h
o
n

Process finished with exit code 0
```

➢ Range function:

The **range()** function has two sets of parameters:

range(stop):

Generate integer numbers, starting from zero to less than stop number.

eg. range (3) == [0, 1, 2]

range([start], stop[, step])

- start: Starting number of the sequence.
- stop: Generate numbers up to, but not including this number.
- step: Difference between each number in the sequence.

Example 5: show how range can be used

```
>>> range(10)
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
>>> range(1, 10)
[1, 2, 3, 4, 5, 6, 7, 8, 9]
>>> range(1, 10, 2)
[1, 3, 5, 7, 9]
>>> range(10, 0, -1)
[10, 9, 8, 7, 6, 5, 4, 3, 2, 1]
>>> range(10, 0, -2)
[10, 8, 6, 4, 2]
>>> range(-5, 5)
[-5, -4, -3, -2, -1, 0, 1, 2, 3, 4]
>>> range(1, 2)
[1]
>>> range(1, 1)
[]
>>> range(1, -1, -1)
[1, 0]
>>> range(0)
[]
```

➢ Iterating by Sequence Index:

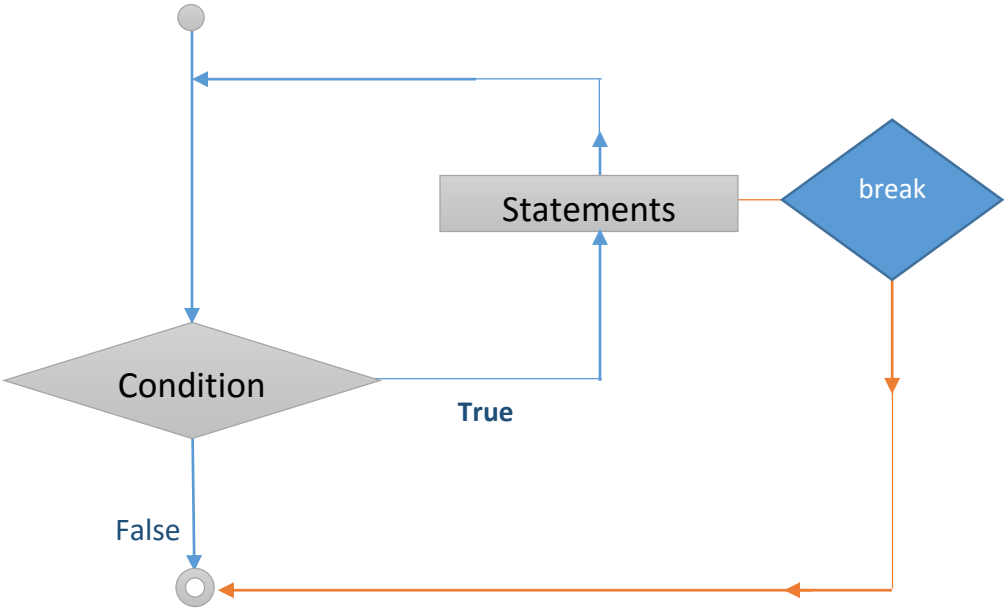
An alternative way of iterating through each item is by index into the sequence itself. We can generate sequence index using range function:

Example 6: using index

```
items=["Orange","Apple","Banana"]
for i in range(len(items)):
    print items[i]
```

➢ break statement:

break statement quits the loop(while , for) without executing the rest of the statements in the loop.



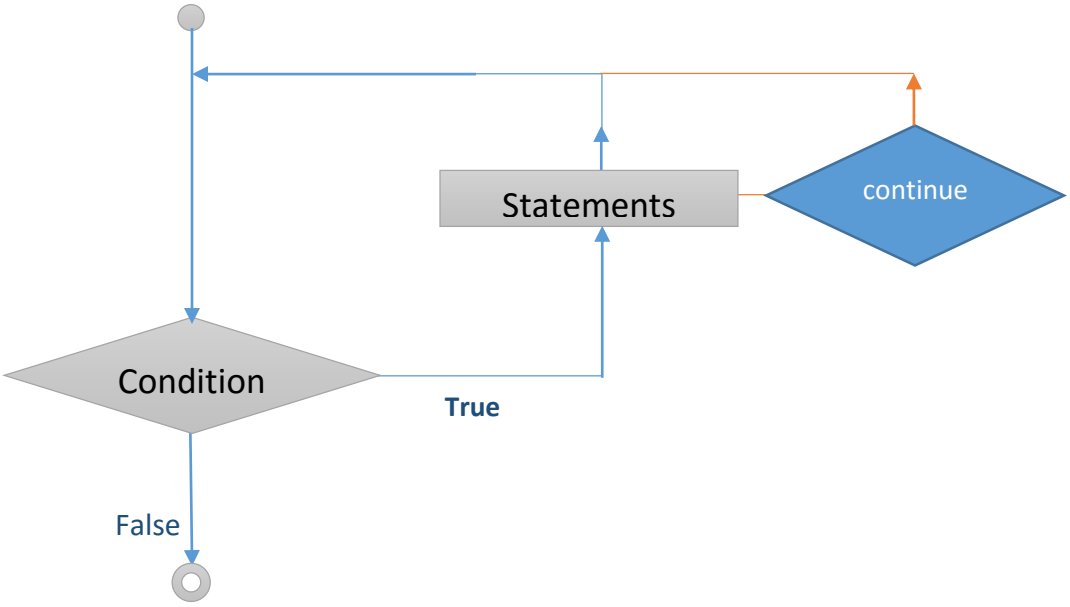
Example 7: break statement

```
num = 10
while num > 0:

    num = num -1
    if num == 5:
        break
    print 'Current number value :', num
```

```
C:\Python27\python.exe C:/Users/ibra
Current number value : 9
Current number value : 8
Current number value : 7
Current number value : 6

Process finished with exit code 0
```

➢ continue statement:

The **continue** statement skip all the remaining statements in the current iteration of the loop and moves the control back to the top of the loop.



Example 8: Use continue statement

```
num = 10
while num > 0:
    num = num -1
    if num == 5:
        continue

    print 'Current number value :', num
```

```
C:\Python27\python.exe C:/Users/i
Current number value : 9
Current number value : 8
Current number value : 7
Current number value : 6
Current number value : 4
Current number value : 3
Current number value : 2
Current number value : 1
Current number value : 0
```

Note:

- The **while** and **for** loop statements use the **else** statement is executed when the condition becomes false.
- The **continue** and **break** statements can be used in both while and for loops.

➢ The Infinite Loop:

A loop becomes infinite loop if a condition never becomes False. You must use caution when using while loops because a loop maybe never ends. Such a loop is called an infinite loop.

Example 9

```
count = 0
while count < 5:
      print 'The count is:', count
```

In this example condition **always true**, so loop never ends, to solve this problem we need to increment count variable every iteration ,until make condition false.

➢ Nested loops:

We can use one loop inside another loop.

For the following matrix, print all numbers in it.

1	12	19
11	7	13
25	10	66

Example 10

```
matrix= [[1,12,19],[11,7,13],[25,10,66]]
for row in matrix:
     for col in row:
         print "%3d"%(col),
     print " "
```

```
C:\Python27\python.exe C:
  1  12  19
 11   7  13
 25  10  66
```

Lab Work:

Q 1) Write a program that sum of all the positive integers less than 100.

Solution

```
sum = 0
for i in range(1, 100):
    sum += i
print "Sum numbers (1 - 99)    =", sum
```

```
C:\Python27\python.exe C:/Users/ibr
Sum numbers (1 - 99)    = 4950

Process finished with exit code 0
```

Q 2) Write a program that sums the integer numbers entered by the user. The program that requests from user to enter an integer until 0 is entered. If so, the sum is displayed.

Solution

```
a= input("Enter numbers to sum, Zero number ends list :  ")
sum=0
while a != 0:
        sum +=a
        a = input("Enter number : ")
print "Sum numbers  = ", sum
```

```
C:\Python27\python.exe C:/Users/ibraheem/PycharmProj
Enter numbers to sum, Zero number ends list :  5
Enter number : 6
Enter number : 5
Enter number : 0
Sum numbers  =  16

Process finished with exit code 0
```

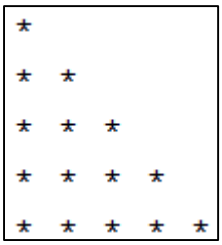
Q 3) Write a program that display sum the even numbers, odd numbers and sum all numbers from 0 to 100.

Solution

```
count =1
sumo=0
sume=0
suma=0
while( count <=100):

    suma +=count

    if count%2==0:
        sume += count
    else:
        sumo +=count

    count +=1

else :
    print "Sum even numbers =", sume
    print "Sum odd  numbers =" ,sumo
    print "Sum all  numbers =" , suma
```

```
C:\Python27\python.exe C:/Users/ibrahe
Sum even numbers = 2550
Sum odd  numbers = 2500
Sum all  numbers = 5050

Process finished with exit code 0
```

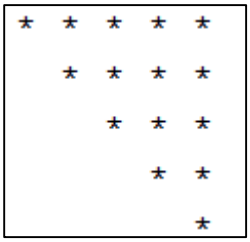
Q 4) Write code that will display the following shape using loop:



Solution

```
for i in range(5):
    for j in range(i+1):
        print "*",
    print ""
```

Q 5) Write a code that will display the following shape using loop:



Solution

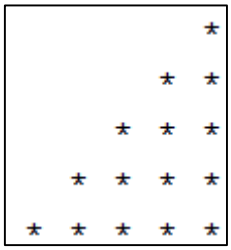
```
for i in range(5):
    for j in range(i):
        print " ",
    for j in range(5-i):
        print "*",
    print ""
```

Another way using **one** loop:

Solution

```
for i in range(5):
    print " "*i ,"*"*(5-i)
```

Q 6) Write a code that will display the following shape using (a while) loop:



Solution

```
row=1
while row<=5:
    star=0
    space=0
    while space < (5-row):
        print " ",
        space += 1
    while star < row:
        print "*",
        star += 1
    print ""
    row +=1
```

Try do it using one loop.

Exercises:

1. Write a program that use for statement to compute the following sums.

   a) 5 + 10 + 15 + . . . +50
   b) 1 +3 +7 +15 + 31 . . . + (2^{20} - 1)

2. Write a program that reads an unspecified number of integers, and determines how many positive and negative values have been read, and computes the total and average of the input values (not counting zeros) then display the average. Your program ends with the input 0.

3. Write a program to take a number from user and find the last index of the number in the list if a number **not** in a list display message "not found".
   for example let the list = [5,3,6,8,9,6,2,3,4].

```
Enter  a number  :3
last index : 7
```

```
Enter  a number  :11
last index :  not found
```

4. Write a program that represent the following matrix and find the maximum and minimum numbers in the matrix :

   For example in the following matrix:
   Max number is 98
   Min number is 8

51	8	19	25
98	16	13	22
25	10	66	13
20	12	64	58

Lab Work

Ex1: Write a program that prints the sum of all positive integers less than 50.

Solution:

```
sum = 0
for i in range(1, 50):
    sum += i
print("Sum numbers (1 to 49) =", sum)
```

Ex2: Write a program that sums the integer numbers entered by the user. The program requests from user to enter an integer until 0 is entered, if so, the sum is displayed.

Solution:

```
a = input("Enter numbers to sum, Zero number ends list : ")
sum = 0
while a != 0:
    sum += a
    a = input("Enter number : ")
print("Sum numbers = ", sum)
```

Ex3: Write a program that asks the user to type 10 integers, then prints the smallest value of the entered integers.

Solution:

```
a = input("Enter a number : ")
min = a
for i in range(9):
    a = input("Enter a number : ")
    if a < min:
        min = a
print("The minimum number = ", min)
```

Ex4: Write a Python program that displays the sum of even numbers form 0 – 100, and the sum of odd numbers from 0 – 100, and the sum of all numbers from 0 – 100.

Solution:

```
count = 0
sumOdd = 0
sumEven = 0
sumAll = 0
while count <= 100:
    sumAll += count
    if count % 2 == 0:
        sumEven += count
    else:
        sumOdd += count
    count += 1

print "Sum even numbers =", sumEven
print "Sum odd numbers =", sumOdd
print "Sum all numbers =", sumAll
```

Homework

1. Write a program that computes the factorial of an integer entered by user.
2. Write a program that use for statement to compute the following sums and products.
    - 1 + 2 + 3 + ... + 100
    - 5 + 10 + 15 + ... + 50
    - 1 + 3 + 7 + 15 + 31 + ... + $(2^{20} - 1)$
    - 1 * 2 * 4 * 8 * ... * 2^{20}
3. Write a program that reads an unspecified number of integers, determines how many positive and negative values have been read, and computes the average of the input values. Your program ends with the input 0. Print the average in float format.

Good Luck

Exercises:

1. Write a program that use for statement to compute the following sums.
   a) 5 + 10 + 15 + . . . +50

Solution

```
sum = 0
for i in range(11):
    sum += i * 5

print sum
```

   b) 1 +3 +7 +15 + 31 . . . + $(2^{20} - 1)$

Solution

```
sum = 0
for i in range(1, 21):
    sum += 2 ** i - 1

print sum
```

2. Write a program that reads an unspecified number of integers, and determines how many positive and negative values have been read, and computes the total and average of the input values (not counting zeros) then display the average. Your program ends with the input 0.

Solution

```
cpos = 0
cneg = 0
sumNums = 0
while True:
    x = input()
    if x == 0:
        break
    elif x > 0:
        cpos += 1
    else:
        cneg += 1

    sumNums += x

print  "Count the positive numbers   = ", cpos
print  "Count the negative  numbers  = ", cneg
print  "Total sum  =  ", sumNums
print  "Average   = ", sumNums /(cneg+cpos)
```

3. Write a program to take a number from user and find the last index of the number in the list if a number **not** in a list display message “not found”.
   for example let the list = [5,3,6,8,9,6,2,3,4].

```
Enter  a number  :3
last index : 7
```

```
Enter  a number  :11
last index :  not found
```

Solution

```
list=[5,3,6,8,9,6,2,3,4]
x=input("Enter  a number  :")
index=-1;
i=0
while i < len(list):
      if list[i] == x:
          index=i
      i+=1

print  "last index :" ,index if index != -1  else " not found"
```

Another solution

```
list = [5, 3, 6, 8, 9, 6, 2, 3, 4]
x = input("Enter  a number  :")
index = len(list) - 1
i = 0
for item in list[::-1]:
    if item == x:
        print  "last index :", index
        break
    index -= 1
else:

    print  "last index :not found"
```

4. Write a program that represent the following matrix and find the maximum and minimum numbers in the matrix :

   For example in the following matrix:
   Max number is 98
   Min number is 8

51	8	19	25
98	16	13	22
25	10	66	13
20	12	64	58

Solution

```
matrix = [
    [51, 8, 19, 25],
    [98, 16, 13, 22],
    [25, 10, 66, 13],
    [20, 12, 64, 58]
]

max = matrix[0][0]
min = matrix[0][0]
for row in matrix:
    for col in row:
        if col > max:
            max = col
        if col < min:
            min = col

print " Max number is ", max
print " Min number is ", min
```

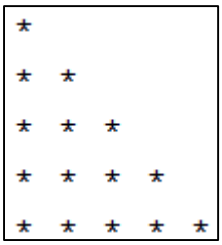
Q 3) Write a program that display sum the even numbers, odd numbers and sum all numbers from 0 to 100.

Solution

```
count =1
sumo=0
sume=0
suma=0
while( count <=100):

    suma +=count

    if count%2==0:
        sume += count
    else:
        sumo +=count

    count +=1

else :
    print "Sum even numbers =", sume
    print "Sum odd  numbers =" ,sumo
    print "Sum all  numbers =" , suma
```

```
C:\Python27\python.exe C:/Users/ibrahe
Sum even numbers = 2550
Sum odd  numbers = 2500
Sum all  numbers = 5050

Process finished with exit code 0
```

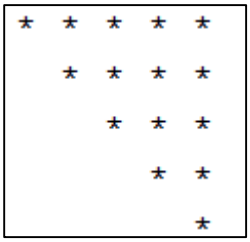
Q 4) Write code that will display the following shape using loop:



Solution

```
for i in range(5):
    for j in range(i+1):
        print "*",
    print ""
```

Q 5) Write a code that will display the following shape using loop:



Solution

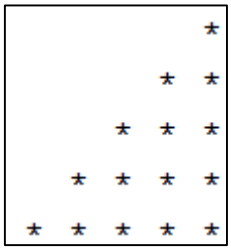
```
for i in range(5):
    for j in range(i):
        print " ",
    for j in range(5-i):
        print "*",
    print ""
```

Another way using **one** loop:

Solution

```
for i in range(5):
    print " "*i ,"*"*(5-i)
```

Q 6) Write a code that will display the following shape using (a while) loop:



Solution

```
row=1
while row<=5:
    star=0
    space=0
    while space < (5-row):
        print " ",
        space += 1
    while star < row:
        print "*",
        star += 1
    print ""
    row +=1
```

Try do it using one loop.

Exercises:

1. Write a program that use for statement to compute the following sums.

   a) 5 + 10 + 15 + . . . +50
   b) 1 +3 +7 +15 + 31 . . . + (2^{20} - 1)

2. Write a program that reads an unspecified number of integers, and determines how many positive and negative values have been read, and computes the total and average of the input values (not counting zeros) then display the average. Your program ends with the input 0.

3. Write a program to take a number from user and find the last index of the number in the list if a number **not** in a list display message "not found".
   for example let the list = [5,3,6,8,9,6,2,3,4].

```
Enter  a number  :3
last index : 7
```

```
Enter  a number  :11
last index :  not found
```

4. Write a program that represent the following matrix and find the maximum and minimum numbers in the matrix :

   For example in the following matrix:
   Max number is 98
   Min number is 8

51	8	19	25
98	16	13	22
25	10	66	13
20	12	64	58

Exercises:

1. Write a program that use for statement to compute the following sums.
   a) 5 + 10 + 15 + . . . +50

Solution

```
sum = 0
for i in range(11):
    sum += i * 5

print sum
```

   b) 1 +3 +7 +15 + 31 . . . + $(2^{20} - 1)$

Solution

```
sum = 0
for i in range(1, 21):
    sum += 2 ** i - 1

print sum
```

2. Write a program that reads an unspecified number of integers, and determines how many positive and negative values have been read, and computes the total and average of the input values (not counting zeros) then display the average. Your program ends with the input 0.

Solution

```
cpos = 0
cneg = 0
sumNums = 0
while True:
    x = input()
    if x == 0:
        break
    elif x > 0:
        cpos += 1
    else:
        cneg += 1

    sumNums += x

print  "Count the positive numbers   = ", cpos
print  "Count the negative  numbers  = ", cneg
print  "Total sum  =  ", sumNums
print  "Average   = ", sumNums /(cneg+cpos)
```

3. Write a program to take a number from user and find the last index of the number in the list if a number **not** in a list display message “not found”.
   for example let the list = [5,3,6,8,9,6,2,3,4].

```
Enter  a number  :3
last index : 7
```

```
Enter  a number  :11
last index :  not found
```

Solution

```
list=[5,3,6,8,9,6,2,3,4]
x=input("Enter  a number  :")
index=-1;
i=0
while i < len(list):
      if list[i] == x:
          index=i
      i+=1

print  "last index :" ,index if index != -1  else " not found"
```

Another solution

```
list = [5, 3, 6, 8, 9, 6, 2, 3, 4]
x = input("Enter  a number  :")
index = len(list) - 1
i = 0
for item in list[::-1]:
    if item == x:
        print  "last index :", index
        break
    index -= 1
else:

    print  "last index :not found"
```

4. Write a program that represent the following matrix and find the maximum and minimum numbers in the matrix :

   For example in the following matrix:
   Max number is 98
   Min number is 8

51	8	19	25
98	16	13	22
25	10	66	13
20	12	64	58

Solution

```
matrix = [
    [51, 8, 19, 25],
    [98, 16, 13, 22],
    [25, 10, 66, 13],
    [20, 12, 64, 58]
]

max = matrix[0][0]
min = matrix[0][0]
for row in matrix:
    for col in row:
        if col > max:
            max = col
        if col < min:
            min = col

print " Max number is ", max
print " Min number is ", min
```

Python Day No.4 Notes