


# Python Programming Fundamentals

## (Chapter 2)

### PYTHON CHARACTER SET

A set of valid sequence of characters recognized by Python is described as Python Character set. Python uses the traditional ASCII character set. The latest version recognizes the Unicode character set. The ASCII character set is a subset of the Unicode character set. All these character sets are used in statements which are executed by Python interpreter and are identified as characters or alphabets, variable/identifier names, and constants, etc. Python supports the following character sets:

- **Letters:** A-Z, a-z
- **Digits:** 0-9
- **Special symbols:** Special symbols available on the keyboard, like + - \* / \*\* \ ( ) [ ] { } @ \$ # & >= <= != = > . < """"; : % ! \_ (underscore)
- **White spaces:** blank space, tab (->), carriage return , new line, form feed.
- **Other characters:** All other Unicode characters.

### Multiline Comments/Continuation Statements

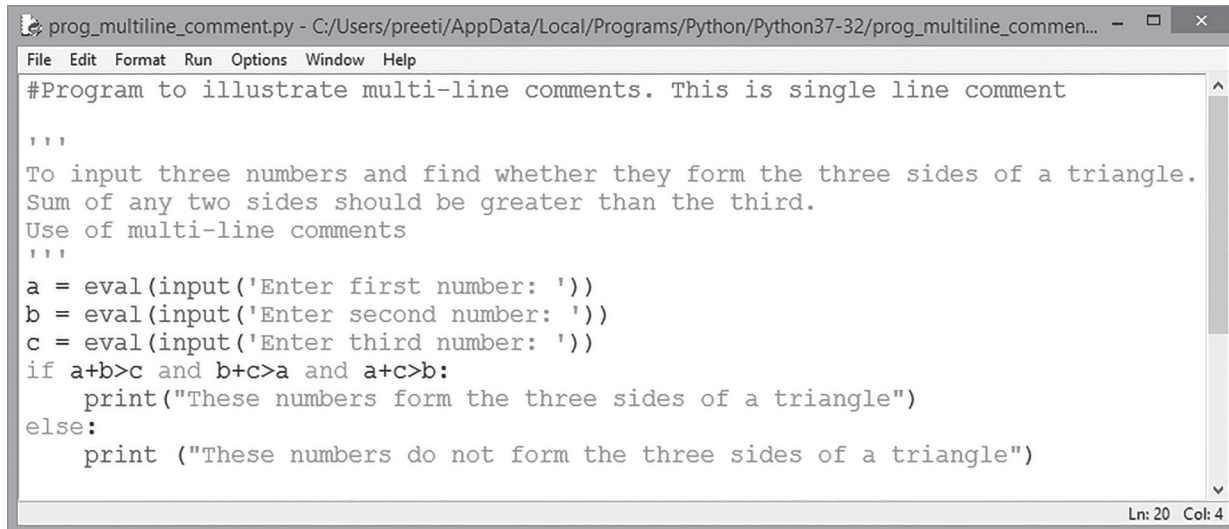
A multiline comment starts and ends with three single quotes (``'). Between start and end of the comment, there can be one or more lines of comment.

The following code contains one multiline comment:

```
'''  
  
This is a multiline comment  
The comment ends here  
'''  
  
x=11  
print(x)
```

## Practical Implementation-1

Write a Python program to input three numbers and find whether they form the three sides of a triangle. Also, give appropriate comments wherever required.

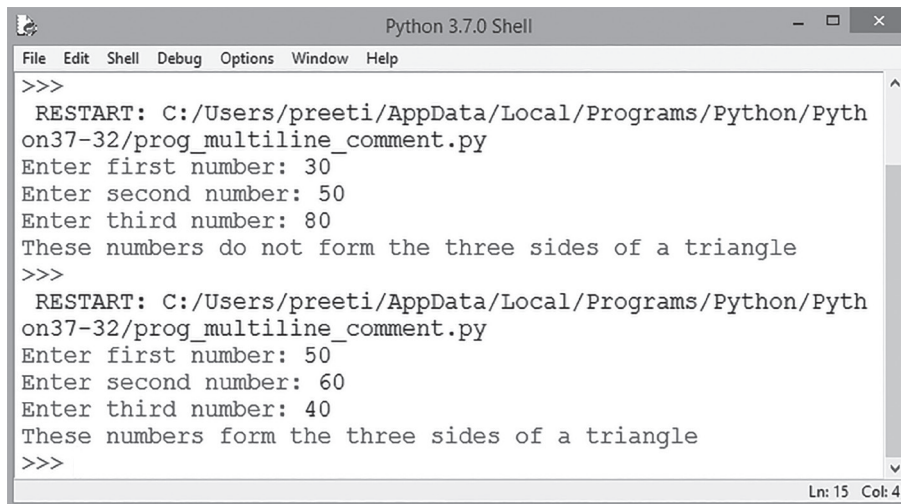


```
prog_multiline_comment.py - C:/Users/preeti/AppData/Local/Programs/Python/Python37-32/prog_multiline_commen...
File Edit Format Run Options Window Help
#Program to illustrate multi-line comments. This is single line comment

'''
To input three numbers and find whether they form the three sides of a triangle.
Sum of any two sides should be greater than the third.
Use of multi-line comments
'''

a = eval(input('Enter first number: '))
b = eval(input('Enter second number: '))
c = eval(input('Enter third number: '))
if a+b>c and b+c>a and a+c>b:
    print("These numbers form the three sides of a triangle")
else:
    print ("These numbers do not form the three sides of a triangle")

Ln: 20 Col: 4
```



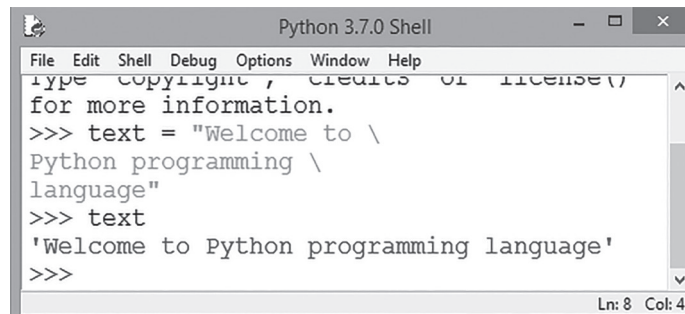
```
Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
>>>
RESTART: C:/Users/preeti/AppData/Local/Programs/Python/Python37-32/prog_multiline_comment.py
Enter first number: 30
Enter second number: 50
Enter third number: 80
These numbers do not form the three sides of a triangle
>>>
RESTART: C:/Users/preeti/AppData/Local/Programs/Python/Python37-32/prog_multiline_comment.py
Enter first number: 50
Enter second number: 60
Enter third number: 40
These numbers form the three sides of a triangle
>>>

Ln: 15 Col: 4
```

## Multiline/Continuation Statements

We can also extend the text across multiple lines using backslash (\). To spread the text across multiple lines, just add a backslash at the end of the statement before pressing **Enter** key to continue typing text on the next line.

*For example,*



```
Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
type copyright, credits or license()
for more information.
>>> text = "Welcome to \
Python programming \
language"
>>> text
'Welcome to Python programming language'
>>>

Ln: 8 Col: 4
```

In the above example, backslash (\) character has been added at the end of each statement, which allows the text to split into multiple lines and hence the output is obtained as shown.

## Clarity and Simplicity of Expression

Expression in a program is a sequence of operators and operands to do an arithmetic or logical computation such as comparing two values, defining a variable or an object and performing arithmetic calculations using one or more variables.

If any expression becomes very big or very complex, we must write it in two steps rather than in a single step. This will make the expression simple and more accurate. Simple expressions are also easily readable and can be understood by others without making any extra effort.



For example,

$$X = (A + B) / (A - B) - (U + VY) / (X + Y)$$

We can simplify it and write as:

$$X1 = (A + B) / (A - B)$$

$$X2 = (U + VY) / (X + Y)$$

$$X = X1 - X2$$

- Use library functions to make programs more powerful.

For example,

To find the value of  $x^4$

Instead of evaluating it using the expression as  $x*x*x*x$

We can use formula available in Python as `power(x, 4)`

## SOLVED QUESTIONS

1. What are comments in Python? Is there any difference between multiline strings and multiline comments?

**Ans.** A comment starts with a # symbol and ends with the end of line. Comments are non-executable statements which are ignored by the Python Interpreter. Multiline strings can be used as multiline comments, but these are not actually comments.

2. What is the output of—  
`33 == 33.0`

**Ans.** True shall be returned as the output as comparison operator evaluates true and false. And in Python we need not specify whether the number is int or float.

3. What will be the output of the following code?

```
x = 2
y = 10
x *= y * x + 1
```

**Ans.** 42; evaluated as `x *= y * x + 1` means `x = x * (y * x + 1)`

4. Which of the following mathematical symbols are part of the Python character set?

`+, -, ^, /, :, !, ≠, →, μ, ≤, =, <=`

**Ans.** The symbols `+, -, ^, =, <=` and `/` form part of mathematical symbols in the Python character set.

## UNSOLVED QUESTIONS

1. What is a character set in Python? What are different categories of character set in Python? Give two examples of each category.
2. Differentiate between multiline strings and multiline comments.
3. What does the continuation character (`\`) signify?
4. How does simplification of statements in Python increase the overall program efficiency?