Python Fundamentals

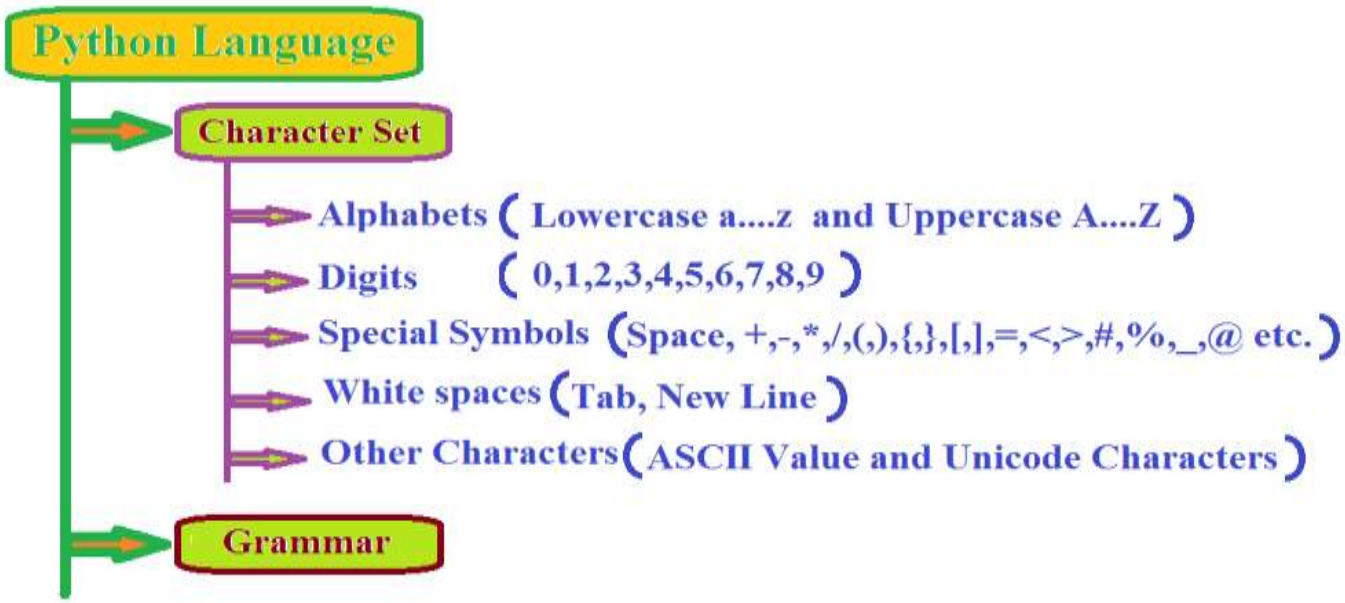
Program:

A Program is set of instructions (statements) that takes some inputs and acts accordingly to carry out specific output. A Python program can contain following basic components.

1. Python Character set
2. Python Tokens
3. Variable
4. Simple Input & Output

Python Character Set:

Character Set is a set of Valid Characters in Python Programming and all these characters can recognized by the Python Interpreter. Following are the Character Sets in Python



Python Tokens:

Everything that is written in python program is a Token or Lexical Unit. In other words, a smallest individual unit in program is known as Token.

Example:

```
# Program to Calculate Sum of Two numbers
A=10
B=20
Sum=A+B
print("Sum= ",Sum)
```

Here
 # is special character that makes a line as comment.
A, B and Sum are variables
=,+,",(,) are Special Characters
Print() is a function of Python

Types of Python Tokens

1. Keywords
2. Identifiers
3. Literals
4. Variables
5. Operators
6. Punctuators
7. Fundamentals of Python Program

1. Keywords (Reserve Words):

The keywords are the reserve words in python language. Keywords have special nature in python like

- Pre-Defined Spelling
- Pre-Defined Special Meaning
- Pre-Defined Purpose (Use)
- Can’t use as general user defined names.

assert	except	True
as	class	for
and	global	from
break	in	import
continue	if	or
del	lambda	raise
def	None	try
else	nonlocal	while
elif	pass	with
finally	return	yield
False	not	is

*All the keywords are written in lower case letters. (Exception: True, False, None)

2. Identifiers

Everything that is user defined in Python Program are termed as identifiers. These are the general user defined names which are used to give a name for variable, function, class, list and dictionary etc.

There are following rules for giving names of identifiers

- The first letter of an identifier must be any alphabet (a-z and A-Z) or underscore (_).
- After first letter there may be any alphabet, digit and underscore.
  Ex. City, Total_Marks, _name, num1, num2
- Other than underscore (_), no special character is allowed in identifier name.
  Ex. first-name, last-name (illegal, because of minus (-) sign)
- Blank space is also not allowed in identifier name.
  Ex. first name, last name (illegal, because of blank space)
- Python is case sensitive language so upper case and lower case characters are recognized as differently.
  Ex. city, City, city, CITY, CiTy all are legal.
- An identifier name must not be a keyword of Python.
- The length of an identifier may be as long as any numbers.

Valid identifiers	Invalid identifiers
My_file	False (it is a Keyword)
myFile	break (it is a Keyword)
_id	2BHK (First letter is digit)
Number1	My.style (. is not allowed)
RollNo23	XI Class
Work2home	(Blank space is not allowed)
KVS_2	XI-Class
One23	(- sign is not allowed)
false (All Small letters)	11_class
Break (First letter Capital)	(Fist letter is digit not allowed)

3. Literals

The literals are the type of fixed values which can assign to variable or which can store in memory. Literals can be used in any calculation as constant data items.

Types of literals

(i) String Literal
(ii) Numeric Literal
(iii) Boolean Literal
(iv) Special Literal-None

(i) String Literals:

Everything that is enclosed between ssingle or double quotation marks is known as string literal.

Single Line String

Ex. 'Hello' , "Hello KVS" , "123" , '@gmail.com' , " 3BHK"
" kv jjn, near DIET campus PIN - 333001"

Multi Line String (Use \)

```
Address= "kv jjn, \
near DIET campus\
PIN - 333001"
```

By using single or double quotation to create multi-line string, back slash (\) required.

OR

```
Address= """kv jjn,
near DIET campus
PIN - 333001"""
```

By using triple quotation to create multi-line string, no back slash (\) required.

Scape Sequence in Python(Using \)

Escape Sequence	Meaning of escape sequence
'\n'	New Line
"\t"	Horizontal Tab
'\v'	Vertical Tab
'\\'	\
'\"'	"

(ii) Numeric Literals:

Numeric type of literals are any of following types

- Integer literal (Values Without decimal point)
- Float literal (Values With decimal point)
- Complex literal (Complex Numbers)

Integer literal

Ex. A=1234, Roll_No=23 N=-15

Float literal

Ex. A=-12.34, percent=15.5, B=10/3,

X=25.6 can be written as $0.256x10^2$ = 0.256 E02

Complex literal

Ex. X=A+Bj

Where A and B are any float numbers and j is an imaginary number.

(iii) Boolean Literals:

Python provides two Boolean values as

True: Boolean True (T is Capital in True)

False: Boolean False (F is Capital in False)

(iv) Boolean Literals:

Python provides a special type of literal value that is **None. (N** is Capital in None). None mean absence of value or No any value provided.

4. Variables:

Variables are the named location which used to reserve the memory location and store the value. These variable can be used in various computations in program.

Ex.

city=”Jaipur”

(Here city is a variable and Jaipur is value of city)

Age=15

(Here Age is a variable and 15 is value of age)

(Variable will be discussed in detail in next chapter)

5. Operators:

The operators are symbols or words which can applied on variables/ values in an expression. Operators trigger some computation / action when they applied in expression.

Ex.

A=100

Sum=A+200

Here Sum and A are variables. 200 is integer literal.

+ is operator that will trigger action of addition A+200.

= is an operator that will assign the result of A+200 into Sum variable.

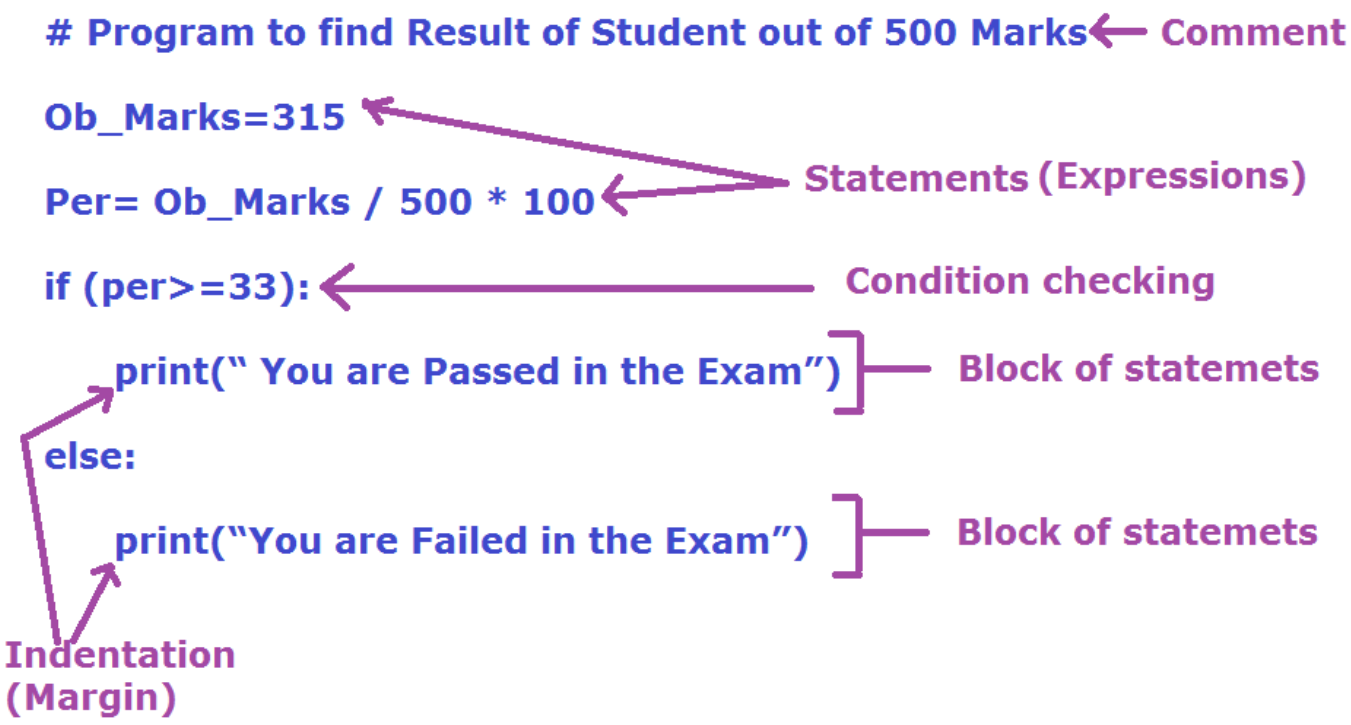
(Operators will be discussed in detail in next chapter)

6. Punctuators:

Punctuators are special type of symbols which are used in Python programming to organize the structure of program. Punctuator put emphasis on expressions, statements, and blocks of program.
Ex. #, ( ), [ ], { }, : , @ etc.

7. Basics of Python Program:

A basic structure of python program is given as below.

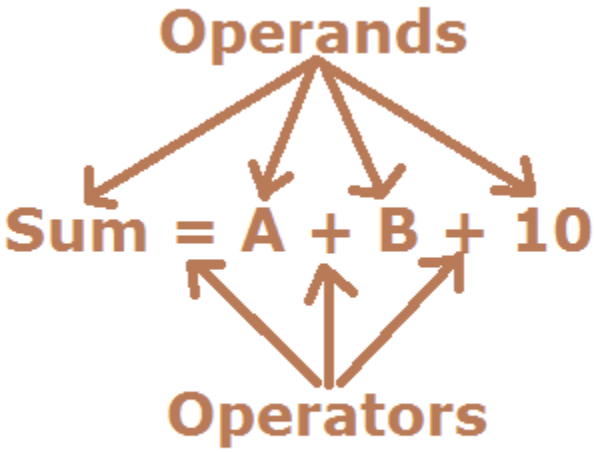


A Python program consist of following components-

(I) Expression
(II) Comment
(III) Block
(IV) Indentation

(I) Expression

It is any valid combination of operators and operands that evaluates and produce a result.



Expreion = Operatorrs + Operands

.

(II) Comments

Comments are non-executable statements in Python language. The following are the characteristics of comments in programming.

- Comments are useful for user to easily understand the code.
- It increase the readability of program.
- Comments are non- executable statements
- Python Interpreter does not read the comments.
- No error capture in comment.
- Comments never increase the total execution time of program.

Two Types of comments:

(A) Single Line Comment

Single Line Comment start with # (hash) sign and it should be ended in one line.

Example:

```
AC=50   # Here AC is Marks of Art & Craft Subject.
FD=80 #Here FD is Marks of Fashion Design subject.
T=AC + FD
# T is total marks of Art & Craft and Fashion
#Design subjects.
```

(B) Multi Line Comment

Multi Line Comment can be written in more than one lines. It can start with either “”” (triple time double quotation) or ‘’’ (triple time Single quotation).

Multi Line comments should be ended with the same type of quotation by which it started.

Example:

```
AC=90
FD=80
T=AC+FD
‘’’ Here AC is Marks of Art & Craft Subject.
FD is Marks of Fashion Design subject.
T is total marks of Art & Craft and Fashion
Design subjects. ‘’’

“”” Here AC is Marks of Art & Craft Subject.
FD is Marks of Fashion Design subject.
T is total marks of Art & Craft and Fashion
Design subjects. “””
```

(III) Block

Block is a group of one or more statements which always executes together. In other words, when we want to execute multiple statements as one statement then we put all such statements under one group and this group is known as block of statements.

Example:

```
Marks=450        # Out of 500          Statement-1
if (Marks > = 0):                      Statement-2
    Per=Marks / 500 * 100              Statement-3
    print("Your Scored: ", Per)        Statement-4
    print("Keep Study Continue")       Statement-5
    print("Good Luck!")                Statement-6
```

(IV) Indentation

Indentation mean the margin of statements from left side of screen. In Python programming Language the indentation plays very important role. It decides the position of statement while executing program.

Example:

```
Num=10
if ( Num >0):
    print("Number is Positive")
elif ( Num <0):
    print("Number is Negative")
elif ( Num ==0):
    print("Number is ZERO")
print("Thanks!")
```

Python Input-Output

Python provides some inbuilt functions for getting input from keyboard and putting output at Screen.

Input- Get / Read

Output- Put / Write

1. input() function
   The input() used to read the string value from keyboard.
2. print() function
   The print() used to write/ display the value on screen.
3. int() function
   int() used to convert the string value in integer value.
4. float() function
   float() used to convert the string value in float value

Example:

```
Name=input("Enter Name of Student :")
Age=intput("Enter Age of Student in Year: ")
Age=int(Age)
Weight= intput("Enter Weight of Student in Kg: ")
Weight=float(Weight)
print(Name)
print(Age)
print(Weight)
```

****Finish****