PYTHON FUNDAMENTALS

IPO-Input, Output, Process

(Computer actions are governed by IPO Cycle)

Python Character Set: is a set of valid characters that a language can recognize. A character represent any letter, digit or any other symbol.

- Letters: A-Z, a-z
- Digits: 0-9
- Special Character etc

Token: The Smallest individual unit in a program is known as token.
Python has following tokens:

1.Keywords 2.Identifiers 3.Literals 4.Operators 5.punctuators

Keywords: Reserve word of the compiler/interpreter which can't be used as identifier.
Example: false, for, if elif, else, break, and, as, while etc.

Identifiers: A Python identifier is a name used to identify a variable, function, module or other object.
Identifier forming rules of python are being specified:

- An identifier is an arbitrarily-long sequence of letters and digits.
- The first character must be a letter; the underscore (_) counts as letter.
- Uppercase and lowercase letters are different. All characters are significant.
- The digits 0 through 9 can be part of the identifier except for the first character.
- identifier are unlimited in length. case is sensitive.
- Identifier must not be a keyword of Python.
- Python does not allow special characters except underscore (_).

Valid Identifiers: Myfile, file123,
Invalid Identifiers: DATA-REC, 2rno

Literals: Literals are data items that have a fixed value. Python allows several kinds of literals:

1 String Literals 2 . Integer Literals 3. Boolean Literals 4. Special Literals None 5. Literals Collections

String Literals: the text enclosed in quotes forms a string literal in python. In string Single quotes or double quotes are treated as same.
Example : 'Johni' , johny"

Nongraphic Characters: are those characters that cannot be typed directly from keyboard. These non geographical characters are represented using escape sequences, it is represented using backslash (\).

Example:
\n : New line character
\t : Horizontal Tab (TAB)

String Types: 1. Single line strings 2. Multiline strings

1. <u>Single line strings:</u> The strings that you create by enclosing text in single (‘ ‘) or double quotes(“ ”). They must terminates in one line.

   Example: Text1=’hello
   there’

2. Multiline strings: They can be created in 2 ways:
   1. By adding backslash at the end of normal single-quote / double-quote strings.
      Example: Text1= ‘hello\
      World’
   2. By typing the text in triple quotation marks. (No backslash needed at the end of line).
      Str=’’’Hello
      World
      ‘’’

<u>Integer Literals</u>: are whole numbers without any fractional part. Example age = 22

There are 3 types of Integer Literals:

1. Decimal Integer Literals- An integer literal consisting of a sequence of digits(0-9) is taken to be decimal integer literal unless it begin with 0. Example 12,67,896
2. Octal Integer Literals:- A sequence of digits starting with Zero followed by digits (0-7) is known as octal Number Example 012,016
3. Hexadecimal Integer Literals:- It is a sequence of digits preceded by OX or ox . It can have digits from 0 to 9 and letter from A to F (10 to 15) Example – 0X1A3B

<u>Floating Point Literals</u>: are numbers having fractional parts.

1. Fractional form: consists of signed or unsigned digits including a decimal point between digits.
   Example 2.0, -13.0, 17.5
2. Exponent form: it has two parts- mantissa and exponent, example: 5.8 can be written as 0.58 X 10^1 =0.58E01, mantissa part is 0.58 and exponent part is 1.

<u>Boolean Literals</u>: is used to represent one of the two Boolean Values i.e True or False. It can either have value as True or as False.

<u>Operators</u>: Operators can be defined as symbols that are used to perform operations on operands.
Types of Operators

1. Arithmetic Operators. 2. Relational Operators. 3. Assignment Operators.
4. Logical Operators. 5. Bitwise Operators 6. Membership Operators
7. Identity Operators

<u>Punctuators</u>: are symbols that are used in programming languages to organize sentence structures, and indicate the rhythm and emphasis of expressions, statements and program structure. Eg: ‘ “ { } ( ) @ #

Expressions: is any legal combination of symbols that represents a value.

Eg: a+5, (3+5)/4

Statements: it is a programming instruction that does something.
e.g
a = 20
print("Calling in proper sequence")

Comments:
which is readable for programmer but ignored by python interpreter.
i.Single line comment: Which begins with # sign.

# program to find sum of three numbers

ii.Multi line comment (docstring): either write multiple line beginning with # sign or use triple quoted multiple line. E.g.
'''this is my
first
python multiline comment
'''

Function:a code that has some name and it can be reused. It is used to avoid repetition of same code again and again. e.g. c=sum(a,b)

Block& indentation: group of statements is block. Indentation at same level create a block.

Variables: Variable is a names location that refers to a value and whose value can be used and processed during program run. A variable can consider as a container which holds value.
Marks=70
Lvalues and Rvalues:

lvalue: Lvalues are the objects to which you can assign value. It is the expressions that can come on the lhs of an assignment.

rvalue: That are the literals and expression that are assigned to lvalue. It is the expressions that can come on the rhs of an assignment.

Eg : a=20

Here a (Memory location) is Lvalue and 20 is rValue

Multiple Assignment: assign a single value to many variables

```
a = b = c = 1   # single value to multiple variable
a,b= 1,2        # multiple value to multiple variable
a,b= b,a       # value of a and b is swaped
```

Dynamic typing: A variable pointing to a value of a certain type, can be made to point to a value/object of different type.

```
X = 25                  # integer type
X = “python”            # x variable data type change to string on just next line
```

Simple Input and Output :- To get input from user you must use built in input() function. You can also read numbers using this method. For this purpose you must use type casting operator int or float as given in examples

```
name=input(‘What is your name?’)

age=int(input(‘What is your age?’))

marks=float(input(‘Enter marks:’))
```

Output: through print statement- print() Function In Python is used to print output on the screen. Syntax of Print Function.

```
print(expression/variable)
```

e.g.

```
print(122)
```

Output :-

```
122
```

```
print('hello India')
```

Output :-

```
hello India
```

```
print(‘Computer',‘Science')
print(‘Computer',‘Science',sep=' &’)
print(‘Computer',‘Science',sep=' & ',end='.')
```

Output :-

```
Computer ScienceComputer & ScienceComputer & Science.
```

Progrm to obtain three numbers and print their sum.

```
num1=int(input(“Enter number 1:”))
num2=int(input(“Enter number 2:”))
num3=int(input(“Enter number 3:”))
Sum=num1+num2+num3
print(“Three numbers are: “, num1,num2,num3)
print(“Sum is:”,Sum)
```