Input and Ouput in Python

What we will learn?

- Output Statement

> - print() statement variations
> - end and sep attribute
> - Formatted String

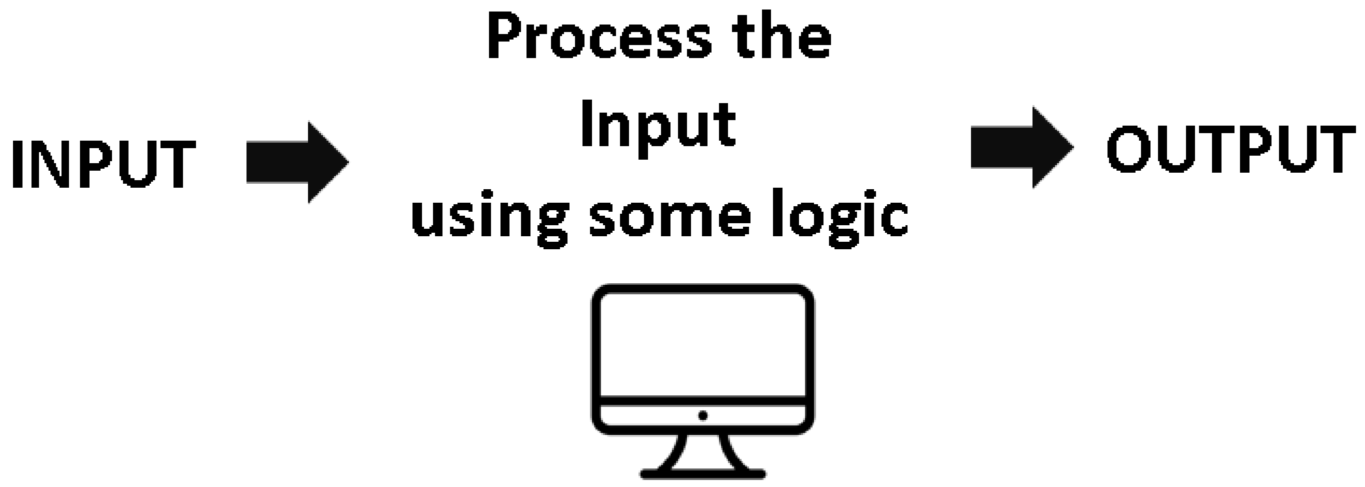
- Input Statement

> - input()
> - int(), float()
> - eval()
> - List comprehension

Introduction

Purpose of computer is to processdata and return resul.

- Data given to computer is INPUT
- Result retured are OUTPUT



Python provides some INPUT and OUTPUT statements

Output Statements

> print() function can be used to display output or results

> ## The print() statement :
>
> > When print() function is called without parameter,it will through cursor to the next line, i.e. Blank line will displayed.

> ## The print("string") Statement :
>
> > When a string is passed, then string is displayed. In strings, double quotes and single quotes can be interchangeably use.

```
print("Python Programming")#Double quotes
print('Python Programming')#single quotes
```

```
     Python Programming
     Python Programming
```

> > Escape Sequence is a charcter that has special meaning. It can be used inside the print() function.

> > To escape the effect of escape sequence, we should use one more \ before the escape sequence.

```
print("It does not do well to dwell on dreams and forget to live.\n \t\t\t\t– Albus Dumbledor
print("It does not do well to dwell on dreams and forget to live.\n \t\\t\t– Albus Dumbledore
```

```
     It does not do well to dwell on dreams and forget to live.
                                     – Albus Dumbledore.
     It does not do well to dwell on dreams and forget to live.
             \t      – Albus Dumbledore.
```

> ## The print(variable list) Statement:
>
> > A list of variables can also be supplied and its values can be printed.

```
x , y = 10 , 'abc'
print(x)
print(y)
print(x,y)
```

```
10
abc
10 abc
```

> **sep Attribute** :
>
> > 1. sep represents **Separator**
> > 2. Default value of sep is **space**
> > 3. We can change the sep value i.e. sep = "characters" which will be used to separate values in the output

```
print(x,y,sep=",")
print(x,y,sep='/')
print(x,y,sep='Z')
print(x,y,sep='--->')
```

```
10,abc
10/abc
10Zabc
10--->abc
```

> **end Attribute** :
>
> > 1. end represents **Ending Character**
> > 2. Default value of end is **new line**
> > 3. We can change the end value i.e. end = "characters" which will be the last thing to be printed after print statement is executed.

```
print('This is')
print('Skill Based')
print('Laboratory')

print('This is ',end='')
print('Skill Based ',end='')
print('Laboratory')

print('This is ',end='\t')
print('Skill Based ',end='\t')
print('Laboratory')
```

```
This is
```

```
Skill Based
Laboratory
This is Skill Based Laboratory
This is         Skill Based     Laboratory
```

The print(object) Statement:

Objects like tuples, lists etc.can be passed to print() function to display the elements of those objects.

```
list1 = ['John',101,35000.00]
print(list1)
```

```
['John', 101, 35000.0]
```

The print("string", variable list) Statement:

Most common usage of print() function is to use strings along with variables.

```
a = 2
print(a,' is an Even Number')
print('a =',a, 'is even number')
```

```
2  is an Even Number
a = 2 is even number
```

The print(formatted string) Statement:

output to be displayed can be formatted using

1. **%** (percent) operator
2. **{ }** placeholder operator

1. Formatting string uisng **% operator** has following format

print("formatted string" % (variable list))

In this formatted string, we can use

- **%i or %d** - Decimal Integer Number

- **%f** - Float Number
- **%s** - String
- **%c**- Single Character

```
a = 101
b = 35000.00
c = 'LMN'
print("Employee ID of %s is %i and his salary is %f" % (c,a,b))
print('\nSingle Character %c' % a)
print('Single character %c'% c[1])

d = 'Python'

print("\nSkill Based Laboratory- %20s programming" % (d))#%20s will allot 20 spaces and strir
print("Skill Based Laboratory- %-20s programming" % (d))#%-20s will allot 20 spaces and strir

e = 123.456789
print("\nFloat value is %10.3f" % e) #3 spaces before 123.457
print("Float value is %.3f" % e)
```

```
Employee ID of LMN is 101 and his salary is 35000.000000

Single Character e
Single character M

Skill Based Laboratory-               Python programming
Skill Based Laboratory- Python               programming

Float value is    123.457
Float value is 123.457
```

2. Formatting string uisng **{ } replacement field** has following format

print("formatted string with replacement fields".format(values))

i. Formatters work by putting in one or more replacement fields and placeholders defined by a pair of curly braces { } into a string and calling the **str.format()**.

ii. The value we wish to put into the placeholders and concatenate with the string passed as parameters into the format function.

```
a, b, c = 100, 200, 300
print("a = {}, b = {}, c ={}".format(a,b,c))
print("a = {}, b = {}, c ={}".format(c,b,c))

#with indexes
print("\na = {0}, b = {1}, c ={2}".format(a,b,c))
print("a = {2}, b = {2}, c ={2}".format(a,b,c))

#with names
print("\na = {one}, b = {two}, c ={three}".format(one=a,two=b,three=c))
```

```
a = 100, b = 200, c =300
a = 300, b = 200, c =300

a = 100, b = 200, c =300
a = 300, b = 300, c =300

a = 100, b = 200, c =300
```

Input Statements

> **input()** function is use to take a value from the keyboard and return it as a string.

> Syntax is **input([prompt])**

```
a  = input()
print ('Entered value is ',a)
```

```
abc
Entered value is  abc
```

```
name = input("Enter your name : ")
print("Name entered is : ", name)
print(type(name))
```

```
Enter your name : anc
Name entered is :  anc
<class 'str'>
```

int() method

> > int() method returns an integer object from any number or string. It takes **two arguments**

> > - x - Number or string to be converted to integer object.Default is zero

```
a = int(input('Enter an integer number '))
print (a, type(a))
```

```
Enter an integer number 12
12 <class 'int'>
```

```
#int() usage is conversion from other base to base 10
a = int(input('Enter a hexadecimal number '),16)
print (a)

a = int(input('Enter an octal number '),8)
print (a)

a = int(input('Enter a binary string '),2)
print (a)
```

```
Enter a hexadecimal number A
10
Enter an octal number 10
8
Enter a binary string 1011
11
```

Multiple inputs in same line

```
#Enter two number searated by space
a , b = [ int (x) for x in input('Enter two numbers:').split()]
print('a = {}  b = {}'.format(a,b))
```

```
Enter two numbers:2 3
a = 2  b = 3
```

```
#Enter three number searated by comma
a , b ,c = [ int (x) for x in input('Enter three numbers:').split(',')]
print('a = {}  b = {}  c = {} '.format(a,b,c))
```

```
Enter three numbers:1,2,3
a = 1  b = 2  c = 3
```

```
#list comrehension
list1 = [x for x in input('Enter names:').split()]
```

```
print(list1)

     Enter names:ABC DEF XYZ
     ['ABC', 'DEF', 'XYZ']
```

eval() function

1. It takes the string and evaluates the result of the string as python expression.
2. input() and eval() can be used together
3. This combination can aslo be used to accept objects like list and tuple

```
a,b = 5,15
print(eval('(a+b)*100'))

#using eval() and input() together
print("Result is ",eval(input("Enter exppression to be evaluated : ")))

     2000
     Enter exppression to be evaluated : a+b-a
     Result is  15
```

```
# Acceting list, tuple, set and dictionary object from the keyboard
list1 = eval(input('Enter list: '))
print(type(list1))
print(list1)

     Enter list: [1,2,3]
     <class 'list'>
     [1, 2, 3]
```