Basic Program Fundamentals in Python

How To Take Input From User In Python

By using the “input() “ Function to take the input from the user.

In python for this we don't need to import any libraries in python.

**Syntax:**

Some point on input() function in python

1. Whatever we take input from the user will be in the string format by default.
2. You can change into your desired data type from the string data type.

Taking input from the user is as follows

===================================

### # asking input from the user

### s=input()

### print(type(s))

### print("The value of is",s)

Output :

90

### <class 'str'>

### The value of is 90

============================================

* By observing the above input and output even though you give the input as an integer value it is stored in the given value as string only.
* The value or string that you give in the user prompt will be stored in the variable, in the above case the string “90” will be stored in the s.
* If you want to take the variable of the desired data type you have to perform typecasting from the string data type to your desired data type.
* At the time of taking input from the user only you can type cast in to the required data type as follows :

**Syntax: variable=required datatype (input())**

Ex1: if you want to take integer value from the user

**Syntax : s=int(input())**

After taking input from the user it will be type casted from string to int

Ex 2: if you want to take a float value from the user.

**Syntax : s=float(input())**

Example:

Input :

##### # Asking for input from the user

##### var\_1=input()

##### print(type(var\_1))

##### print("The Value of var\_1 is ", var\_1)

##### # Taking input as integer from the user

##### var\_2=int(input())

##### print(type(var\_2))

##### print("The Value of var\_2 is ", var\_2)

##### # Taking input as float from the user

##### var\_3=float(input())

##### print(type(var\_3))

##### print("The Value of var\_3 is ", var\_3)

Output

##### Python

##### <class 'str'>

##### The Value of var\_1 is Python

##### 125

##### <class 'int'>

##### The Value of var\_2 is 125

##### 125.5

##### <class 'float'>

##### The Value of var\_3 is 125.5

=======================================================

The above format is used when you want to take a single input from the user without displaying any instructions or message to the user.

When you want to provide any instructions to the user then use

**syntax:input(“Enter Message to display ”)**

Example for displaying any instructions while taking input from the user.

===============================================

Input:

##### length=int(input("Enter the length in meters :"))

##### print("Length you entered is :", length)

Output:

##### Enter the length in meters :20

##### Length you entered is : 20

=====================================================

##

## **Taking multiple Multiple Inputs in a single line**

when you want to take multiple values from the user of the same data type you need to include some functions like map and split

Ex: if you want to take two values of same data type from user then use map function

Syntax : a,b=map(desired data type,input().split())

Example :

Input:

##### # Asking user to enter multiple inputs

##### var\_1,var\_2=map(int,input("Enter the values side by side: ").split())

##### print("The value of first variable is :", var\_1)

##### print("The value of second variable is :", var\_2)

Output:

##### Enter the values side by side: 125 130

##### The value of first variable is : 125

##### The value of second variable is : 130

=================================================

##

##

## **Taking Input for the Sequence Data Types like List, Set.**

In this sequence data types we can take input in two ways

1.using a for loop or while loop and add the elements using append in list or add in set.

2.using map functions.

1. using loops and append method

Before using the append or add method you have to create one empty sequence data type or use the present sequence data type.

=====================================================

Input:

##### # Creating an empty list using in-built function

##### list\_1=list()

##### # Creating an empty set using in-built function

##### set\_1=set()

##### # Taking how many elements you wants to add in the sequence

##### n=int(input("Enter number of elements to add in list: "))

##### for i in range(n):

#####  list\_1.append(int(input("Enter the element for List:")))

##### k=int(input("Enter number of elements to add in set: "))

##### for i in range(k):

#####  set\_1.add(int(input("Enter the element for set:")))

##### print("List elements are:", list\_1)

##### print("Set elements are:", set\_1)

Output

##### Enter number of elements to add in list: 2

##### Enter the element for List:125

##### Enter the element for List:130

##### Enter number of elements to add in set: 3

##### Enter the element for set:135

##### Enter the element for set:140

##### Enter the element for set:145

##### List elements are: [125, 130]

##### Set elements are: {145, 140, 135}

================================================

2. Another way of taking sequential input is to use list and set functions along with the map,by using this you can get many inputs from the user.

Example using the map and split function

=============================================

Input:

##### # Taking list as input from the user

##### list\_1=list(map(int,input("Enter the list elements: ").split()))

##### print("The elements in list are: ",list\_1)

##### # Taking set as input from the user

##### set\_1=set(map(int,input("Enter the set elements: ").split()))

##### print("The elements in set are: ",set\_1)

Output

##### Enter the list elements: 10 20 30 40

##### The elements in list are: [10, 20, 30, 40]

##### Enter the set elements: 90 125 124 130

##### The elements in set are: {90, 124, 125, 130}

==============================================

**More about print() in python**

**Syntax:**  print(object(s), sep= ' ', end = '\n', file=file, flush=flush)

About the syntax:

object(s): one or more values or expressions that are to be printed.

Sep: used to insert in between the values.Default value for the separator is single space.

end: it will specify what to print at the end.by default it has “/n” which means new line.

file: An object with the write method.Stream where the output is to be printed.

Default value = Standard output

flush: it is a boolean value that specifies if the output is flushed or buffered.By default value is False.

flush then the boolean value is True.then the processing of output is slow

Buffered then the boolean value is False.

**Example**

Input:

##### a=77

##### b=90

##### c=89

##### # To print the values side by side

##### print(a,b)

##### # To separate the values with the desired symbol use "sep" in print

##### # Here we used comma to separate the two values

##### print(a,b, sep=",")

##### # To separate the values with hash symbol

##### print(a,b,c, sep="#")

#####

##### Output

##### 77 90

##### 77,90

##### 77#90#89

=======================================================

**Printing of numbers with the help of loop**

When you use loops to print a sequence of numbers then every number gets into a new line.

If we want to print them side by side then you can use end in the print function.

Example

Input:

##### # printing numbers without using end

##### print("Printing the numbers By Default way ")

##### for i in range(4):

#####  print(i)

##### # Printing numbers using the "end"

##### print("To Print the numbers side by side using end ")

##### for k in range(4):

#####  print(k,end=" ")

Output: Printing the numbers By Default way

###### 0

###### 1

###### 2

###### 3

###### To Print the numbers side by side using end

###### 0 1 2 3

**Comments in Python**

* Comments are used to add some explanatory notes.
* These lines are ignored by the python interpreter while executing the program.
* Comments are used to increase the readability of the code and help the other programmers to understand the code.

There are of three types of comments in python

1. Single line comments
2. Inline comments
3. Multi line comments

**Single line comments**

This type of comments start with hash symbol (#).

Example:

Input:

##### # Single line comment

##### print("Hello World\_2 ")

##### # print("Hello World\_3 ")

Output

#### Hello World\_2

In the above mentioned input line 2 and line 3 we used print to get output but line 3 is not printed in the output because that instruction in the line 3 will not be executed and it will be ignored by the python interpreter.As we used (#) before the instruction it will be considered as the comment.

==================================================

**Inline Comments**

Comments that appear at the end of a line of code.

The instruction before that comment will be executed by the interpreter.

Example

a=20 # assigning the value to a

print(a)

Output:

5

===============================================

**Multi line Comments**

Whenever you want to write comments in more than a single line then we use multi line comments.

We use triple Quotes for the multi line comments.

 **Example**

""" multi line comments in python

we use triple quotes for multi line comments """

=========================================================