Taking input in Python

Developers often have a need to interact with users, either to get data or to provide some sort of result. Most programs today use a dialog box as a way of asking the user to provide some type of input. While Python provides us with two inbuilt functions to read the input from the keyboard.

- **input (prompt)**
- **raw_input (prompt)**

input (): This function first takes the input from the user and converts it into a string. The type of the returned object always will be <class 'str'>. It does not evaluate the expression it just returns the complete statement as String. For example, Python provides a built-in function called input which takes the input from the user. When the input function is called it stops the program and waits for the user's input. When the user presses enter, the program resumes and returns what the user typed.

Syntax:

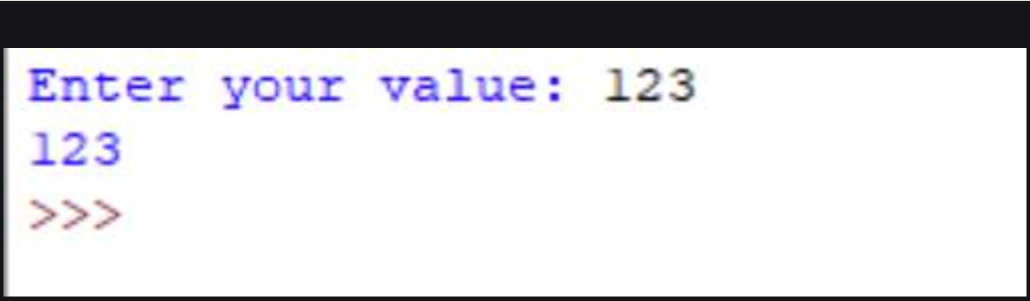
```
inp = input('STATEMENT')
```

```
Example:

1.  >>> name = input('What is your name?\n')     # \n ---> newline  ---> It
causes a line break
            >>> What is your name?
            Ram
            >>> print(name)
            Ram

            # ---> comment in python
```

- Python3

```
# Python program showing
# a use of input()

val = input("Enter your value: ")
print(val)
```

Output:



```
Enter your value: 123
123
>>>
```

Taking String as an input:

- Python3

```
name = input('What is your name?\n')     # \n ---> newline  ---> It causes a line break
print(name)
```

Output:

```
What is your name?

Ram

Ram
```

How the input function works in Python :

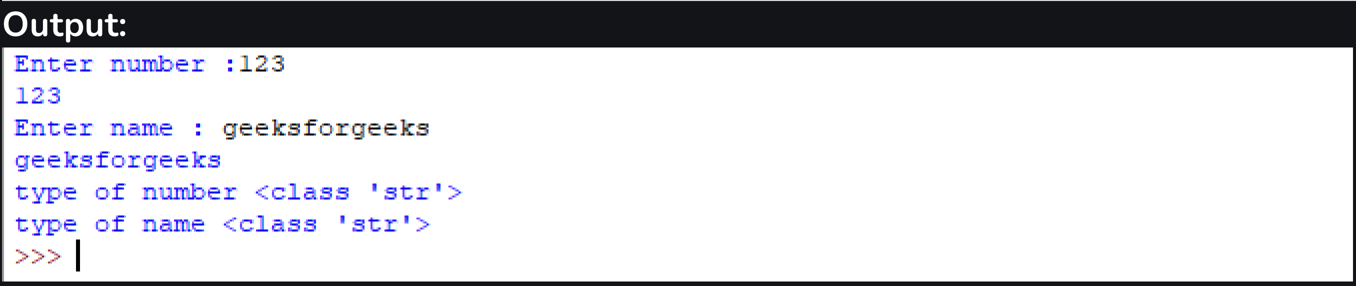
- When input() function executes program flow will be stopped until the user has given input.
- The text or message displayed on the output screen to ask a user to enter an input value is optional i.e. the prompt, which will be printed on the screen is optional.
- Whatever you enter as input, the input function converts it into a string. if you enter an integer value still input() function converts it into a string. You need to explicitly convert it into an integer in your code using typecasting.

Code:

- Python3

```
# Program to check input
# type in Python

num = input ("Enter number :")
print(num)
name1 = input("Enter name : ")
print(name1)

# Printing type of input value
print ("type of number", type(num))
print ("type of name", type(name1))
```

Output:



```
Enter number :123
123
Enter name : geeksforgeeks
geeksforgeeks
type of number <class 'str'>
type of name <class 'str'>
>>> 
```

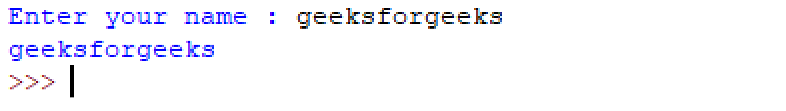
raw_input(): This function works in older version (like Python 2.x). This function takes exactly what is typed from the keyboard, converts it to string, and then returns it to the variable in which we want to store it.

Example:

- Python

```
# Python program showing
# a use of raw_input()

g = raw_input("Enter your name : ")
print g
```

Output:



```
Enter your name : geeksforgeeks
geeksforgeeks
>>> |
```

Here, ***g*** is a variable that will get the string value, typed by the user during the execution of the program. Typing of data for the raw_input() function is terminated by enter key. We can use raw_input() to enter numeric data also. In that case, we use typecasting. For more details on typecasting refer this.

***Note:** input() function takes all the input as a string only*

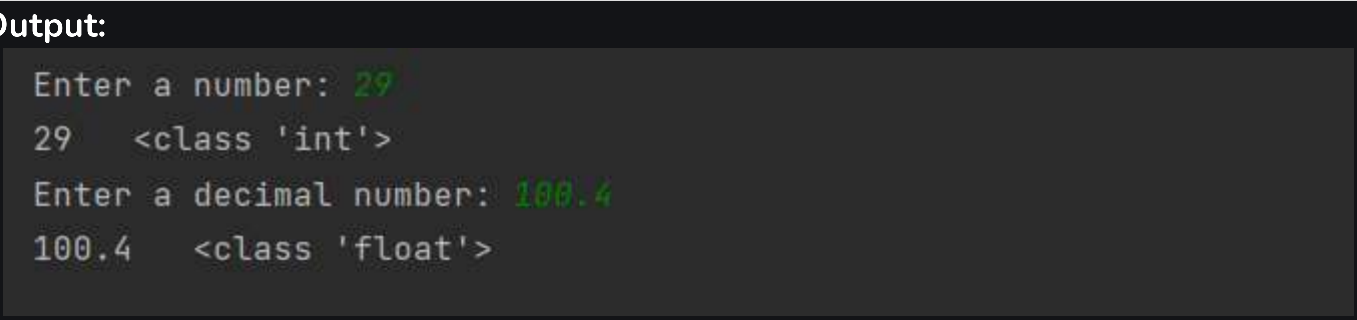
There are various function that are used to take as desired input few of them are : –

- int(input())
- float(input())

- Python3

```
num = int(input("Enter a number: "))
print(num, " ", type(num))


floatNum = float(input("Enter a decimal number: "))
print(floatNum, " ", type(floatNum))
```

Output:

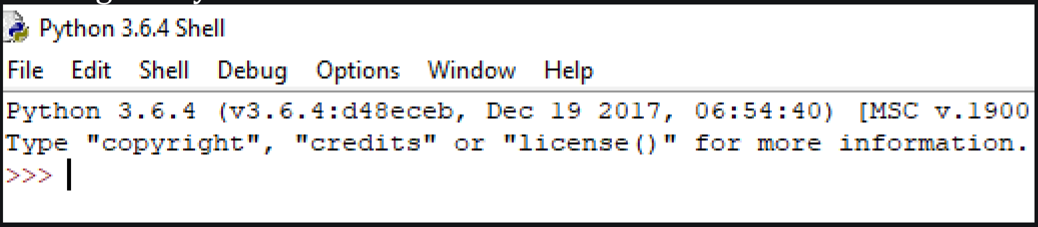


```
Enter a number: 29
29   <class 'int'>
Enter a decimal number: 100.4
100.4   <class 'float'>
```

Output

Taking input from console in Python

What is Console in Python? Console (also called Shell) is basically a command line interpreter that takes input from the user i.e one command at a time and interprets it. If it is error free then it runs the command and gives required output otherwise shows the error message. A Python Console looks like this.

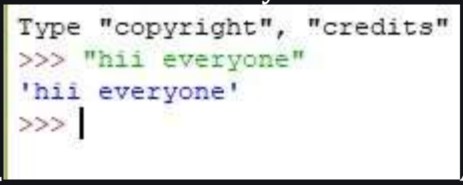


```
Python 3.6.4 (v3.6.4:d48eceb, Dec 19 2017, 06:54:40) [MSC v.1900
Type "copyright", "credits" or "license()" for more information.
>>> |
```

Here we write a command and to execute the command just press enter key and your command will be interpreted. For coding in Python, you must know the basics of the console used in Python. The primary prompt of the python console is the three greater than symbols

>>>

You are free to write the next command on the shell only when these prompts have appeared after executing the first command. The Python Console accepts commands in Python that you write after the prompt.



```
Type "copyright", "credits"
>>> "hii everyone"
'hii everyone'
>>> |
```

Accepting Input from Console User enters the values in the Console and that value is then used in the program as it was required. To take input from the user we make use of a built-in function *input()*.

- Python

```
# input
input1 = input()


# output
print(input1)
```

We can also typecast this input to integer, float, or string by specifying the input() function inside the type.

1. Typecasting the input to Integer: There might be conditions when you might require integer input from the user/Console, the following code takes two input(integer/float) from the console and typecasts them to an integer then prints the sum.

- Python

```
# input
num1 = int(input())
num2 = int(input())
```

```
# printing the sum in integer
print(num1 + num2)
```

2. Typecasting the input to Float: To convert the input to float the following code will work out.

- Python

```
# input
num1 = float(input())
num2 = float(input())


# printing the sum in float
print(num1 + num2)
```

3. Typecasting the input to String: All kinds of input can be converted to string type whether they are float or integer. We make use of keyword str for typecasting.
we can also take input string by just writing *input()* function by default it makes the input string

- Python

```
# input
string = str(input())


# output
print(string)


# Or by default
string_default = input()


# output
print(string_default)
```

Taking multiple inputs from user in Python

The developer often wants a user to enter multiple values or inputs in one line. In C++/C user can take multiple inputs in one line using scanf but in Python user can take multiple values or inputs in one line by two methods.

- Using split() method
- Using List comprehension

Using split() method :

This function helps in getting multiple inputs from users. It breaks the given input by the specified separator. If a separator is not provided then any white space is a separator. Generally, users use a split() method to split a Python string but one can use it in taking multiple inputs.

Syntax :

```
input().split(separator, maxsplit)
```

Example :

- Python3

```
# Python program showing how to

# multiple input using split


# taking two inputs at a time

x, y = input("Enter two values: ").split()

print("Number of boys: ", x)

print("Number of girls: ", y)


# taking three inputs at a time

x, y, z = input("Enter three values: ").split()

print("Total number of students: ", x)

print("Number of boys is : ", y)

print("Number of girls is : ", z)


# taking two inputs at a time

a, b = input("Enter two values: ").split()

print("First number is {} and second number is {}".format(a, b))


# taking multiple inputs at a time

# and type casting using list() function

x = list(map(int, input("Enter multiple values: ").split()))
```

```
print("List of students: ", x)
```

Output:

```
Enter two values: 5 10

Number of boys:  5

Number of girls:  10

Enter three values: 5 10 15

Total number of students:  5

Number of boys is :  10

Number of girls is :  15

Enter two values: 5 10

First number is 5 and second number is 10

Enter multiple values: 5 10 15 20 25

List of students:  [5, 10, 15, 20, 25]
```

Using List comprehension :

List comprehension is an elegant way to define and create a list in Python. We can create lists just like mathematical statements in one line only. It is also used in getting multiple inputs from a user.

Example:

- Python3

```
# Python program showing

# how to take multiple input

# using List comprehension


# taking two input at a time

x, y = [int(x) for x in input("Enter two values: ").split()]

print("First Number is: ", x)

print("Second Number is: ", y)


# taking three input at a time

x, y, z = [int(x) for x in input("Enter three values: ").split()]

print("First Number is: ", x)

print("Second Number is: ", y)

print("Third Number is: ", z)
```

```
# taking two inputs at a time
x, y = [int(x) for x in input("Enter two values: ").split()]
print("First number is {} and second number is {}".format(x, y))

# taking multiple inputs at a time
x = [int(x) for x in input("Enter multiple values: ").split()]
print("Number of list is: ", x)
```

Output :

```
Enter two values: 5 10
First Number is:  5
Second Number is:  10
Enter three values: 5 10 15
First Number is:  5
Second Number is:  10
Third Number is:  15
Enter two values: 5 10
First number is 5 and second number is 10
Enter multiple values: 5 10 15 20 25
Number of list is:  [5, 10, 15, 20, 25]
```

Note: The above examples take input separated by spaces. In case we wish to take input separated by comma (,), we can use the following:

- Python3

```
# taking multiple inputs at a time separated by comma
x = [int(x) for x in input("Enter multiple value: ").split(",")]
print("Number of list is: ", x)
```