

Python Programming



**RGM College of Engineering & Technology
(Autonomous)**

Department of Computer Science & Engineering

Academic Year : 2020-2021

INPUT-OUTPUT STATEMENTS - 2

Agenda:

- 1. Reading multiple values from the keyboard in a single line**
- 2. eval() Function**



Guido Van Rossum

Dept. of CSE, RGM CET(Autonomous), Nandyal

Learning Mantra

**If you really strong in the basics, then
remaining things will become so easy.**

3. Reading multiple values from the keyboard in a single line

Eg:

```
a,b= [int(x) for x in input("Enter 2 numbers :").split()] # List Comprehensions  
print("The Sum is :", a + b)
```

Output:

Enter 2 numbers :10 20

The Sum is : 30

Explanation :

- ❑ Here, we are using only one input function (i.e., **input("Enter 2 numbers :")**). So what ever you provide it is treated as only one string.
- ❑ Suppose, you are providing input as 10 20, this is treated as single string.
- ❑ If you want to split that string into multiple values, then we required to use **split()** function .
- ❑ If we want to split the given string (i.e., 10 20) with respect to space, then the code will be as follows:

input('Enter 2 numbers :').split()

- ❑ Here, we are not passing any argument to split() function, then it takes default parameter (i.e., space) as separator.
- ❑ Now this single string(i.e.,10 20) is splitted into list of two string values.

- ❑ Now the statement : **`input('Enter 2 numbers :').split()`** returns **`['10','20']`**
- ❑ Now, in the list every number is available in string form.
- ❑ So, what we will do here is, each value present in this list is typecast to 'int' value.
`[int(x) for x in input('Enter 2 numbers :').split()]` ==> it retrns `[10,20]`
- ❑ This concept is known as **list comprehension**.
- ❑ **`a,b = [int(x) for x in input('Enter 2 numbers :').split()]`** → it assigns 10 to a and 20 to b. This concept is called as **list unpacking**.
- ❑ **`a,b = [int(x) for x in input('Enter 2 numbers :').split()]`**
- ❑ **`print('Sum is : ',a + b)`** ==> Gives the sum of two values as the result.

❑ **split() function** can take space as separator by default .But we can pass anything as separator.

```
s = input("Enter 2 numbers :")
```

```
print(s, type(s))          # s holds single value '10 20'
```

```
l=s.split() # After split, single string will be divided into list of two values of str type
```

```
print(l,type(l))          → ['10', '20'] <class 'list'>
```

```
l1 = [int(x) for x in l] # This new list contains two int values after type casting of each print(l1)
```

```
a,b = l1 # in this list whatever the values are there, assigns first value to 'a' and second #  
        # value to b. This is called 'list unpacking'.
```

```
print(a)                  → 10
```

```
print(b)                  → 20
```

```
print('Sum is :', a+b)    → 30
```

- ❑ **By substituting the elements of the above code, we will get the below code (same as above code):**

```
a,b = [int(x) for x in input("Enter 2 numbers :").split()]  
print('Sum is :', a+b)
```

Output:

Enter 2 numbers :10 20

Sum is : 30

```
a,b = [int(x) for x in input("Enter 2 numbers :").split(',')]  
print('Sum is :', a+b)
```

Output:

Enter 2 numbers :10,20

Sum is : 30

```
a,b = [int(x) for x in input("Enter 2 numbers :").split(',')]  
print('Sum is :', a+b)
```

Output:

Enter 2 numbers :10 20 **# You didn't gave ','**

ValueError Traceback (most recent call last)

<ipython-input-22-d2adfb9ab442> in <module>

```
----> 1 a,b = [int(x) for x in input("Enter 2 numbers :").split(',')]  
      2 print('Sum is :', a+b)
```

<ipython-input-22-d2adfb9ab442> in <listcomp>(.0)

```
----> 1 a,b = [int(x) for x in input("Enter 2 numbers :").split(',')]  
      2 print('Sum is :', a+b)
```

ValueError: invalid literal for int() with base 10: '10 20'

Demo Program 3: Q. Write a program to read 3 float numbers from the keyboard with , separator and print their sum.

eval() Function:

- ❑ **eval()** Function is a single function which is the replacement of all the typecasting functions in Python.

```
x = (input('Enter Something : '))  
print(type(x))
```

Output:

Enter Something : 'karthi'

<class 'str'>

```
x = (input('Enter Something : '))  
print(type(x))
```

Output:

Enter Something : 10

<class 'str'>

```
x = (input('Enter Something : '))  
print(type(x))
```

Output:

```
Enter Something : 10.5  
<class 'str'>
```

```
x = eval(input('Enter Something : '))  
print(type(x))
```

Output:

```
Enter Something : 'karthi'  
<class 'str'>
```

```
x = eval(input('Enter Something : '))  
print(type(x))
```

Output:

```
Enter Something : 10  
<class 'int'>
```

```
x = eval(input('Enter Something : '))  
print(type(x))
```

Output:

```
Enter Something : 10.5  
<class 'float'>
```

```
x = eval(input('Enter Something : '))  
print(type(x))
```

Output:

```
Enter Something : True  
<class 'bool'>
```

```
x = eval(input('Enter Something : '))  
print(type(x))
```

Output:

```
Enter Something : [10,20,30]  
<class 'list'>
```



```
x = eval(input('Enter Something : '))  
print(type(x))
```

Output:

```
Enter Something : (10,20,30)  
<class 'tuple'>
```

```
x = eval(input('Enter Something : '))  
print(type(x))
```

Output:

```
Enter Something : (10)  
<class 'int'>
```

```
x = eval(input('Enter Something : '))  
print(type(x))
```

Output:

```
Enter Something : (10,  
<class 'tuple'>
```

❑ **If you provide an expression as a string type, eval() function evaluates that expression and provide the result.**

```
x = eval('10+20+30')  
print(x,type(x))      ➔ 60 <class 'int'>
```

```
x = eval(10+20+30)  
print(x,type(x))      ➔ TypeError: eval() arg 1 must be a string, bytes or code object
```

Any question?



If you try to practice programs yourself, then you will learn many things automatically

Spend few minutes and then enjoy the study

The background of the slide features abstract, overlapping green geometric shapes in various shades of green, creating a modern and dynamic look. The shapes are primarily located on the right side and bottom of the slide, with some extending towards the center.

Thank You