

Stack Overflow is a community of 4.7 million programmers, just like you, helping each other.

Join them; it only takes a minute:

[Sign up](#)

Join the Stack Overflow community to:



Ask programming questions



Answer and help your peers



Get recognized for your expertise

## python ValueError : too many values to unpack



I am a python beginner . I was trying to run this code :

```
def main():  
    print ( " This program computes the average of two exam scores . ")  
    score1,score2 = input ("Enter two scores separated by a comma:")  
    average = (score1 + score2)/2.0  
    print ("The average of the score is : " , average )
```

when I summon the `main()` I got this `ValueError` :

```
ValueError: too many values to unpack (expected 2)
```

what is wrong with this code ?

`python` `python-3.x`

edited Sep 30 '13 at 16:05

[Leon Young](#)  
457 ● 3 ● 10

asked Sep 30 '13 at 15:48

[DAVID](#)  
13 ● 1 ● 1 ● 3

1 @musical\_coder: Don't try to remove `()` from print functions unless you're absolutely sure the OP isn't using Python 3. – [Wooble](#) Sep 30 '13 at 15:51

### 3 Answers

1. You need to split the input you receive, because it arrives all in one string
2. Then, you'll need to convert them to numbers, because the term `score1 + score2` will do string addition otherwise and you will get an error.

answered Sep 30 '13 at 15:50

[wim](#)  
62.2k ● 20 ● 120 ● 206



Did you find this question interesting? Try our newsletter

Sign up for our newsletter and get our top new questions delivered to your inbox ([see an example](#)).

You need to split on the comma:

```
score1,score2 = input ("Enter two scores separated by a comma:").split(",")
```

Note however that `score1` and `score2` will still be strings. You will need to convert them into numbers using either `float` or `int` (depending on what number type you want).

See an example:

```
>>> score1,score2 = input("Enter two scores separated by a comma:").split(",")  
Enter two scores separated by a comma:1,2  
>>> score1  
'1'  
>>> score1 = int(score1)  
>>> score1  
1  
>>> score1 = float(score1)  
>>> score1  
1.0
```

```
1.0
>>>
```

edited Sep 30 '13 at 15:55

answered Sep 30 '13 at 15:50



iCodez

72.2k ● 16 ● 88 ● 114

The input arrives as a single string. But Python has a split personality when it comes to strings: it can treat them as a single string value, or as a list of characters. When you tried to assign it to `score1, score2` it decided you wanted a list of characters. Evidently you typed more than two characters, so it said you had too many.

The other answers have perfectly good suggestions for doing what you really wanted so I won't repeat them here.

answered Sep 30 '13 at 15:55



Mark Ransom

170k ● 18 ● 168 ● 373