Practice > Python > Numpy > Shape and Reshape

Shape and Reshape ★

Your Shape and Reshape submission got 0.00 points. **Try Again!**   ✕

| Problem | Submissions | Leaderboard | Discussions | Editorial 🔒 |

shape

The shape tool gives a tuple of array dimensions and can be used to change the dimensions of an array.

(a). Using shape to get array dimensions

```
import numpy

my__1D_array = numpy.array([1, 2, 3, 4, 5])
print my_1D_array.shape    #(5,) -> 1 row and 5 columns

my__2D_array = numpy.array([[1, 2],[3, 4],[6,5]])
print my_2D_array.shape    #(3, 2) -> 3 rows and 2 columns
```

(b). Using shape to change array dimensions

```
import numpy

change_array = numpy.array([1,2,3,4,5,6])
change_array.shape = (3, 2)
print change_array

#Output
[[1 2]
[3 4]
[5 6]]
```

reshape

The reshape tool gives a new shape to an array without changing its data. It creates a new array and does not modify the original array itself.

```
import numpy

my_array = numpy.array([1,2,3,4,5,6])
print numpy.reshape(my_array,(3,2))

#Output
[[1 2]
[3 4]
[5 6]]
```

Task

You are given a space separated list of nine integers. Your task is to convert this list into a 3X3 NumPy array.

Input Format

A single line of input containing 9 space separated integers.

Output Format

Print the 3X3 NumPy array.

Sample Input

```
1 2 3 4 5 6 7 8 9
```

Sample Output

```
[[1 2 3]
 [4 5 6]
 [7 8 9]]
```

Author	[deleted]
Difficulty	Easy
Max Score	20
Submitted By	59098

NEED HELP?

🖵 View discussions

📖 View editorial

🏆 View top submissions

RATE THIS CHALLENGE

★ ★ ★ ★ ★

MORE DETAILS

⬇ Download problem statement

⬇ Download sample test cases

✎ Suggest Edits

f 𝕏 in

```
1    import numpy
2    a=numpy.array(list(map(int,input().split())))
3    print(numpy.reshape(a,(3,3)))
```

Line: 3 Col: 30

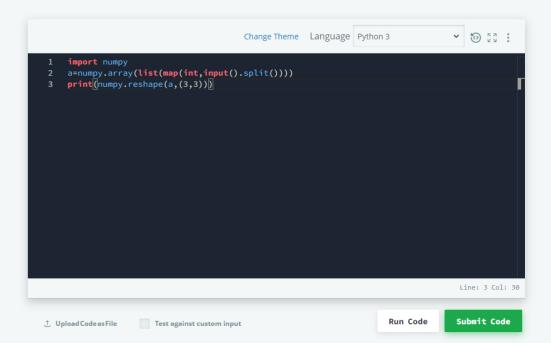⬆ Upload Code as File   ☐ Test against custom input

Run Code   **Submit Code**

Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

✅ **Sample Test case 0**

Input (stdin)   Download

```
1    1 2 3 4 5 6 7 8 9
```

Your Output (stdout)

```
1    [[1 2 3]
2     [4 5 6]
3     [7 8 9]]
```

Expected Output   Download

```
1    [[1 2 3]
2     [4 5 6]
3     [7 8 9]]
```