Transpose and Flatten

Your Transpose and Flatten submission got 20.00 points. Share Tweet

Try the next challenge | Try a Random Challenge

Transpose

We can generate the transposition of an array using the tool `numpy.transpose`.

It will not affect the original array, but it will create a new array.

```
import numpy

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

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

Flatten

The tool *flatten* creates a copy of the input array flattened to one dimension.

```
import numpy

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

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

Task

You are given a NXM integer array matrix with space separated elements (N = rows and M = columns).

Your task is to print the *transpose* and *flatten* results.

Input Format

The first line contains the space separated values of N and M.

The next N lines contains the space separated elements of M columns.

Output Format

Author DOSHI

Difficulty Easy

Max Score 20

Submitted By 13057

NEED HELP?

View discussions

View editorial

View top submissions

RATE THIS CHALLENGE

MORE DETAILS

Download problem statement

Download sample test cases

Suggest Edits

First, print the *transpose* array and then print the *flatten*.

Sample Input

```
2 2
1 2
3 4
```

Sample Output

```
[[1 3]
 [2 4]]
[1 2 3 4]
```

Current Buffer (saved locally, editable) Python 3

```
1  import numpy
2  n,m=[int(i) for i in input().split()]
3  l=[]
4  for i in range(n):
5    l.append([int(j) for j in input().split()])
6  myarray=numpy.array(l)
7  print(numpy.transpose(myarray))
8  print(myarray.flatten())
9
10
11
```

Line: 6 Col: 23

Upload Code as File Test against custom input Run Code Submit Code

You have earned 20.00 points!
87/115 challenges solved. 76%

Congratulations

You solved this challenge. Would you like to challenge your friends?

Next Challenge

Testcase 0 Testcase 1 Testcase 2

Input (stdin) Download

```
2 2
1 2
3 4
```

Expected Output Download

```
[[1 3]
 [2 4]]
[1 2 3 4]
```

Compiler Message

```
Success
```