

## Shape and Reshape

### 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]]
```

2/2

```
Change Theme Language Pypy 3 ↕ ↺ ⚙ ⋮

1  # Enter your code here. Read input from STDIN. Print output to STDOUT
2  import numpy as np
3  arr=list(map(int,input().split()))
4  arr=np.array(arr)
5  print(arr.reshape(3,3))
6
```