

What is NumPy?

- NumPy stands for Numerical Python. NumPy is a Python library used for working with arrays.
- It is an open source project and you can use it freely. NumPy can be used to perform a wide variety of mathematical operations on arrays.
- It adds powerful data structures to Python that guarantee efficient calculations with arrays and matrices.

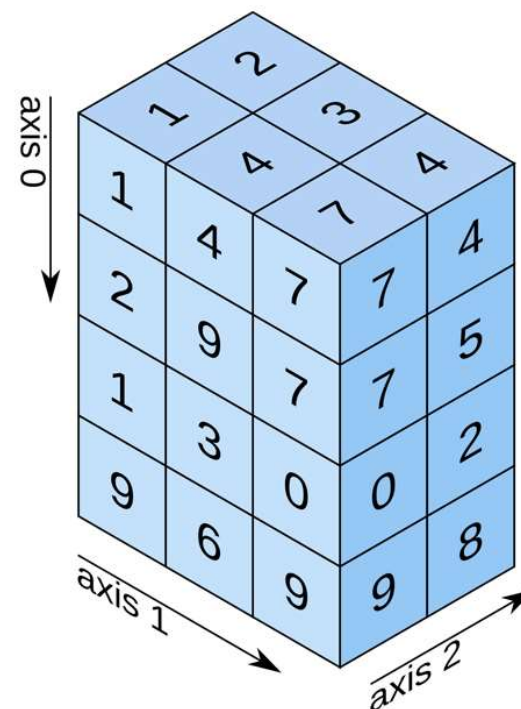
Why we use NumPy?

- In Python we have lists that serve the purpose of arrays, but they are slow to process.
- NumPy aims to provide an array object that is up to 50x faster than traditional Python lists.
- The array object in NumPy is called ndarray, it provides a lot of supporting functions that make working with ndarray very easy.
- Arrays are very frequently used in data science, where speed and resources are very important.

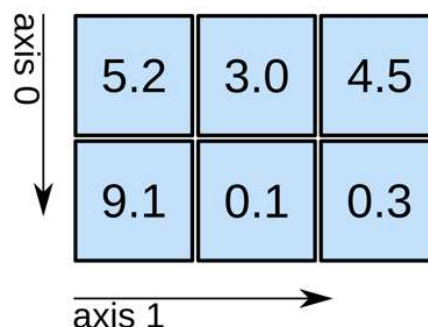
What is NumPy Array?

- An array is a central data structure of the NumPy library. An array is a grid of values and it contains information about the raw data, how to locate an element, and how to interpret an element.
- The most important object defined in NumPy is an N-dimensional array type called ndarray. It describes the collection of items of the same type.

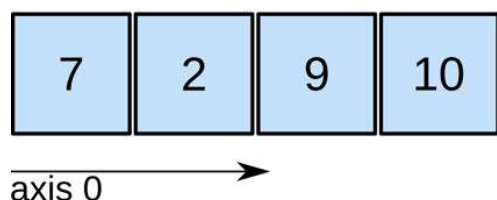
3D tensor



2D tensor



1D tensor



shape: (4,)

shape: (2, 3)

shape: (4, 3, 2)

In []: ▶

In [1]: ▶

Requirement already satisfied: numpy in c:\users\admin\anaconda3\lib\site-packages (1.24.3)

In [2]: ▶

```
In [3]: 1st = [1,2,3,4,5]
1st
```

```
Out[3]: [1, 2, 3, 4, 5]
```

```
In [4]: type(1st)
```

```
Out[4]: list
```

```
In [5]: arr = np.array([1,2,3,4,5])
print(arr)
```

```
[1 2 3 4 5]
```

```
In [8]: arr
```

```
Out[8]: array([1, 2, 3, 4, 5])
```

```
In [6]: type(arr)
```

```
Out[6]: numpy.ndarray
```

```
In [7]: arr.ndim
```

```
Out[7]: 1
```

```
In [ ]:
```

Difference between Numpy Array and Python List

- **Data Type Storage**
 - Homogeneous (Numpy Array)
 - Heterogeneous (Python List)
- **Numerical Operations**
 - Vectorized (Numpy Array)
 - Iterative (Python List)
- **Performance and Speed**

- High Speed and Less Memory (Numpy Array)
- Low Speed and More Memory (Python List)

In []: ▶

In [9]: ▶ *# diff 1*

```
lst = [2,4,6.15,'a']  
lst
```

Out[9]: [2, 4, 6.15, 'a']

In []: ▶

In [10]: ▶

```
arr = np.array([1,2,3,4,5])  
arr
```

Out[10]: array([1, 2, 3, 4, 5])

In [11]: ▶

```
arr = np.array([1,2,3,4.8,5])  
arr
```

Out[11]: array([1. , 2. , 3. , 4.8, 5.])

In [12]: ▶

```
arr = np.array([1,2,3,4,5,'a'])  
arr
```

Out[12]: array(['1', '2', '3', '4', '5', 'a'], dtype='<U11')

In [14]: ▶

```
arr = np.array([1,2,3,4.8,5,'a'])  
arr
```

Out[14]: array(['1', '2', '3', '4.8', '5', 'a'], dtype='<U32')

In []: ▶

```
In [16]: ❏ # diff 2

arr = np.array([1,2,3,4,5]) * 2
arr
```

Out[16]: array([2, 4, 6, 8, 10])

```
In [17]: ❏ arr = np.array([1,2,3,4,5]) + 2
arr
```

Out[17]: array([3, 4, 5, 6, 7])

```
In [ ]: ❏
```

```
In [22]: ❏ lst = [1,2,3,4,5]
lst
```

Out[22]: [1, 2, 3, 4, 5]

```
In [19]: ❏ lst = [1,2,3,4,5] + 2
lst
```

TypeError

Traceback (most recent call last)

Cell In[19], line 1

```
----> 1 lst = [1,2,3,4,5] + 2
      2 lst
```

TypeError: can only concatenate list (not "int") to list

```
In [20]: ❏ lst = [1,2,3,4,5] * 2
lst
```

Out[20]: [1, 2, 3, 4, 5, 1, 2, 3, 4, 5]

```
In [25]: 1 = []  
         for i in lst:  
             1.append(i*2)  
  
         1
```

Out[25]: [2, 4, 6, 8, 10]

```
In [ ]:
```

```
In [ ]:
```

```
In [26]: # diff 3  
  
         list(range(1, 11))
```

Out[26]: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

```
In [27]: [i**2 for i in list(range(1, 11))]
```

Out[27]: [1, 4, 9, 16, 25, 36, 49, 64, 81, 100]

```
In [34]: %timeit [i**2 for i in list(range(1, 1001))]
```

71.9 μ s \pm 836 ns per loop (mean \pm std. dev. of 7 runs, 10,000 loops each)

```
In [ ]:
```

```
In [29]: np.arange(1, 11)
```

Out[29]: array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10])

```
In [30]: np.arange(1, 11) ** 2
```

Out[30]: array([1, 4, 9, 16, 25, 36, 49, 64, 81, 100])

```
In [35]: %timeit np.arange(1, 1001) ** 2
```

2.42 μ s \pm 95.4 ns per loop (mean \pm std. dev. of 7 runs, 100,000 loops each)

```
In [ ]:
```

Creating 1D, 2D and 3D Array

```
In [42]: # 1D
```

```
arr = np.array([1,2,3,4,5])  
arr
```

Out[42]: array([1, 2, 3, 4, 5])

```
In [43]: arr.ndim
```

Out[43]: 1

```
In [44]: arr.shape
```

Out[44]: (5,)

```
In [ ]:
```

```
In [45]: # 2D
```

```
arr = np.array([[1,2,3,4,5]])  
arr
```

Out[45]: array([[1, 2, 3, 4, 5]])

```
In [46]: arr.ndim
```

Out[46]: 2

```
In [47]: ▶ arr.shape
```

```
Out[47]: (1, 5)
```

```
In [ ]: ▶
```

```
In [48]: ▶ arr = np.array([[1,2,3],[4,5,6]])  
arr
```

```
Out[48]: array([[1, 2, 3],  
               [4, 5, 6]])
```

```
In [49]: ▶ arr.ndim
```

```
Out[49]: 2
```

```
In [50]: ▶ arr.shape
```

```
Out[50]: (2, 3)
```

```
In [ ]: ▶
```

```
In [51]: ▶ arr = np.array([[1,2,3],[4,5,6], [7,8,9]])  
arr
```

```
Out[51]: array([[1, 2, 3],  
               [4, 5, 6],  
               [7, 8, 9]])
```

```
In [52]: ▶ arr.ndim
```

```
Out[52]: 2
```

```
In [53]: ▶ arr.shape
```

```
Out[53]: (3, 3)
```


In []: ▶

In [54]: ▶ *# 3D*

```
arr = np.array([[[1,2,3,4,5,6,7,8,9]]])  
arr
```

Out[54]: array([[[1, 2, 3, 4, 5, 6, 7, 8, 9]]])

In [55]: ▶ arr.ndim

Out[55]: 3

In [56]: ▶ arr.shape

Out[56]: (1, 1, 9)

In [57]: ▶ arr = np.array([[[1,2,3],[4,5,6],[7,8,9]]])
arr

Out[57]: array([[[1, 2, 3],
 [4, 5, 6],
 [7, 8, 9]])])

In [58]: ▶ arr.shape

Out[58]: (1, 3, 3)

In [59]: ▶ arr = np.array([[[1,2],[3,4]],[[5,6],[7,8]]])
arr

Out[59]: array([[[1, 2],
 [3, 4]],
 [[5, 6],
 [7, 8]]])


```
In [71]: ▶ arr = np.zeros(5)
arr
```

```
Out[71]: array([0., 0., 0., 0., 0.])
```

```
In [73]: ▶ arr = np.zeros([3,3], dtype=int)
arr
```

```
Out[73]: array([[0, 0, 0],
               [0, 0, 0],
               [0, 0, 0]])
```

```
In [74]: ▶ arr = np.zeros([3,3,3], dtype=int)
arr
```

```
Out[74]: array([[[0, 0, 0],
                 [0, 0, 0],
                 [0, 0, 0]],

               [[0, 0, 0],
                 [0, 0, 0],
                 [0, 0, 0]],

               [[0, 0, 0],
                 [0, 0, 0],
                 [0, 0, 0]])
```

```
In [ ]: ▶
```

```
In [77]: ▶ arr = np.ones(5, dtype=int)
arr
```

```
Out[77]: array([1, 1, 1, 1, 1])
```

```
In [78]: ▶ arr = np.ones([2,3], dtype=int)
arr
```

```
Out[78]: array([[1, 1, 1],
               [1, 1, 1]])
```

```
In [79]: ▶ arr = np.ones([2,2,3], dtype=int)
arr
```

```
Out[79]: array([[[1, 1, 1],
                 [1, 1, 1]],

                [[1, 1, 1],
                 [1, 1, 1]]])
```

```
In [ ]: ▶
```

```
In [82]: ▶ arr = np.full([3,3], 7)
arr
```

```
Out[82]: array([[7, 7, 7],
                [7, 7, 7],
                [7, 7, 7]])
```

```
In [83]: ▶ arr = np.full([3,3], 'sourav')
arr
```

```
Out[83]: array([['sourav', 'sourav', 'sourav'],
                ['sourav', 'sourav', 'sourav'],
                ['sourav', 'sourav', 'sourav']], dtype='<U6')
```

```
In [ ]: ▶
```

```
In [84]: ▶ arr = np.arange(1, 21)
arr
```

```
Out[84]: array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16, 17,
                18, 19, 20])
```

```
In [85]: ▶ arr = np.arange(1, 21, 3)
arr
```

```
Out[85]: array([ 1,  4,  7, 10, 13, 16, 19])
```

In []: ▶

Random Number Generation

In [86]: ▶ `from numpy import random`In [102]: ▶ `arr = random.randint(0, 100, 5)`
`arr`Out[102]: `array([83, 43, 31, 59, 4])`In [115]: ▶ `arr = random.randint(100, size=[3,3])`
`arr`Out[115]: `array([[29, 37, 40],
 [27, 60, 62],
 [7, 64, 30]])`In [139]: ▶ `arr = random.randint(1000, size=[3,3,3])`
`arr`Out[139]: `array([[[544, 868, 81],
 [3, 899, 266],
 [978, 978, 849]],

 [[577, 479, 912],
 [900, 255, 339],
 [591, 534, 798]],

 [[630, 413, 700],
 [642, 392, 140],
 [642, 574, 697]]])`

In []: ▶

```
In [145]: ▶ arr = random.rand(5)
arr
```

```
Out[145]: array([0.78006878, 0.15830625, 0.91738036, 0.05212452, 0.35688169])
```

```
In [147]: ▶ arr = random.rand(2,5,4)
arr
```

```
Out[147]: array([[[0.22746573, 0.74171749, 0.03685585, 0.63395297],
                  [0.01805078, 0.42744625, 0.00487779, 0.17462455],
                  [0.82733088, 0.46510993, 0.99551384, 0.55956239],
                  [0.85047944, 0.8727314 , 0.10429517, 0.79542156],
                  [0.71396047, 0.4588318 , 0.47138403, 0.55086507]],
                 [[0.23300848, 0.75441855, 0.297782 , 0.13552822],
                  [0.2754164 , 0.93035224, 0.73476509, 0.17342851],
                  [0.02050605, 0.82029464, 0.36401698, 0.08830369],
                  [0.52928357, 0.25903951, 0.37858917, 0.44907175],
                  [0.54785719, 0.78192948, 0.66936096, 0.50989781]]])
```

```
In [ ]: ▶
```

```
In [150]: ▶ arr = random.uniform(30000, 100000, 40)
arr
```

```
Out[150]: array([60076.71672282, 85655.03772477, 30648.77328872, 71290.72219387,
                41526.64550933, 54584.33354167, 94316.85903833, 72810.01374891,
                32901.51428615, 91712.35424573, 63292.49204933, 34751.97381301,
                55241.21484767, 56581.42928152, 98326.09709851, 66018.82535841,
                66178.47341522, 93407.25077321, 58127.39011514, 91448.05326932,
                66390.95831405, 40711.98640532, 93322.21283226, 43725.38482806,
                58617.48930094, 56134.45418608, 97276.7778601 , 45081.60282569,
                83440.76296858, 56942.59745188, 51579.78709208, 86946.33598299,
                85682.69339958, 72747.80494331, 36444.07006515, 54469.50034057,
                98414.17004025, 87573.60199037, 57661.55778519, 95056.17960149])
```

```
In [ ]: ▶
```

In []: ▶

```
In [170]: ▶ arr = random.choice([3,5,7,9,11,13], size=10)
arr
```

```
Out[170]: array([ 3,  7,  5,  5, 11,  5,  3,  7,  5,  7])
```

```
In [177]: ▶ arr = random.choice([3,5,7,9], size=(3,4,3))
arr
```

```
Out[177]: array([[[7, 3, 7],
                  [9, 5, 7],
                  [9, 7, 5],
                  [3, 9, 3]],

                 [[9, 3, 7],
                  [3, 9, 9],
                  [3, 9, 7],
                  [9, 7, 3]],

                 [[3, 5, 3],
                  [5, 3, 3],
                  [7, 9, 9],
                  [7, 7, 7]]])
```

```
In [175]: ▶ arr = random.choice([3,5,7,9], size=100)
arr
```

```
Out[175]: array([5, 3, 3, 7, 9, 9, 7, 5, 3, 3, 5, 3, 7, 3, 3, 5, 9, 5, 5, 9, 3, 5,
                  7, 5, 5, 7, 5, 9, 7, 9, 9, 7, 3, 3, 9, 7, 9, 9, 7, 3, 3, 9, 5, 5,
                  3, 3, 9, 5, 3, 3, 3, 9, 3, 7, 5, 9, 5, 5, 7, 9, 7, 7, 7, 5, 5, 7,
                  5, 7, 7, 9, 3, 7, 7, 9, 7, 3, 7, 5, 9, 5, 7, 9, 9, 3, 3, 7, 9, 7,
                  5, 7, 5, 3, 9, 7, 9, 7, 3, 5, 5, 9])
```

```
In [181]: ▶ arr = random.choice([3,5,7,9], p=[0.1, 0.2, 0.0, 0.7], size=100)
arr
```

```
Out[181]: array([9, 9, 9, 9, 9, 9, 9, 5, 9, 9, 9, 9, 9, 9, 9, 5, 9, 9, 5, 9, 9, 9,
          9, 9, 9, 9, 3, 5, 5, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 5, 9,
          5, 9, 9, 9, 9, 9, 5, 9, 3, 9, 9, 3, 9, 9, 3, 9, 3, 9, 5, 9, 3, 9,
          3, 9, 9, 3, 9, 5, 9, 9, 9, 3, 9, 9, 3, 9, 9, 3, 5, 3, 9, 9, 9, 9,
          9, 5, 9, 9, 9, 9, 9, 9, 3, 9, 5, 5])
```

```
In [182]: ▶ arr = random.choice([3,5,7,9], p=[0.3, 0.0, 0.7, 0.0], size=100)
arr
```

```
Out[182]: array([7, 3, 7, 3, 7, 3, 3, 7, 7, 7, 7, 3, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7,
          3, 7, 7, 3, 7, 7, 7, 7, 7, 7, 3, 7, 7, 7, 3, 3, 3, 7, 3, 3, 7, 3,
          7, 7, 7, 7, 3, 3, 7, 7, 3, 7, 7, 3, 7, 7, 3, 3, 7, 7, 7, 3, 7,
          7, 3, 7, 7, 3, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 3, 7, 3, 7,
          3, 7, 3, 7, 7, 7, 3, 7, 7, 3, 3, 3])
```

```
In [ ]: ▶
```

Arithmetic / Vectorised Operation

```
In [183]: ▶ arr = np.array([1,2,3,4,5])
arr
```

```
Out[183]: array([1, 2, 3, 4, 5])
```

```
In [184]: ▶ arr = np.array([1,2,3,4,5]) + 5
arr
```

```
Out[184]: array([ 6,  7,  8,  9, 10])
```

```
In [185]: ▶ arr = np.array([1,2,3,4,5]) - 5
arr
```

```
Out[185]: array([-4, -3, -2, -1,  0])
```



```
In [186]: ▶ arr = np.array([1,2,3,4,5])  
arr = arr - 5  
arr
```

```
Out[186]: array([-4, -3, -2, -1,  0])
```

```
In [187]: ▶ arr = np.array([1,2,3,4,5]) * 5  
arr
```

```
Out[187]: array([ 5, 10, 15, 20, 25])
```

```
In [188]: ▶ arr = np.array([1,2,3,4,5]) / 5  
arr
```

```
Out[188]: array([0.2, 0.4, 0.6, 0.8, 1.  ])
```

```
In [189]: ▶ arr = np.array([1,2,3,4,5]) // 5  
arr
```

```
Out[189]: array([0, 0, 0, 0, 1], dtype=int32)
```

```
In [190]: ▶ arr = np.array([1,2,3,4,5]) % 5  
arr
```

```
Out[190]: array([1, 2, 3, 4, 0], dtype=int32)
```

```
In [191]: ▶ arr = np.array([1,2,3,4,5]) ** 5  
arr
```

```
Out[191]: array([ 1, 32, 243, 1024, 3125], dtype=int32)
```

```
In [ ]: ▶
```

Mathematical Operation using function

```
In [192]: ▶ arr1 = np.array([1,2,3,4,5])  
arr2 = np.array([1,2,3,4,5])
```

```
In [193]: ▶ arr1 + arr2
```

```
Out[193]: array([ 2,  4,  6,  8, 10])
```

```
In [194]: ▶ arr1 = np.array([1,2,3,4,5])  
arr2 = np.array([1,2,3,4,5])  
  
np.add(arr1, arr2)
```

```
Out[194]: array([ 2,  4,  6,  8, 10])
```

```
In [195]: ▶ arr1 = np.array([1,2,3,4,5])  
arr2 = np.array([1,2,3,4,5])  
  
a = np.subtract(arr1, arr2)  
a
```

```
Out[195]: array([0, 0, 0, 0, 0])
```

```
In [196]: ▶ arr1 = np.array([1,2,3,4,5])  
arr2 = np.array([1,2,3,4,5])  
  
a = np.multiply(arr1, arr2)  
a
```

```
Out[196]: array([ 1,  4,  9, 16, 25])
```

```
In [197]: ▶ arr1 = np.array([1,2,3,4,5])  
arr2 = np.array([1,2,3,4,5])  
  
a = np.divide(arr1, arr2)  
a
```

```
Out[197]: array([1., 1., 1., 1., 1.])
```

```
In [198]: ▶ arr1 = np.array([1,2,3,4,5])
arr2 = np.array([1,2,3,4,5])

a = np.floor_divide(arr1, arr2)
a
```

Out[198]: array([1, 1, 1, 1, 1])

```
In [199]: ▶ arr1 = np.array([1,2,3,4,5])
arr2 = np.array([1,2,3,4,5])

a = np.mod(arr1, arr2)
a
```

Out[199]: array([0, 0, 0, 0, 0])

```
In [200]: ▶ arr1 = np.array([1,2,3,4,5])
arr2 = np.array([1,2,3,4,5])

a = np.power(arr1, arr2)
a
```

Out[200]: array([1, 4, 27, 256, 3125])

```
In [ ]: ▶
```

Statistical Operations

```
In [204]: ▶ arr = np.array([4,5,5,7,6,2,1,3,6,87,89,4,63,3,2,216,498,1,65,63,1324,6])
arr
```

Out[204]: array([4, 5, 5, 7, 6, 2, 1, 3, 6, 87, 89, 4, 63, 3, 2, 216, 498, 1, 65, 63, 1324, 6])

```
In [205]: ▶ len(arr)
```

Out[205]: 22

```
In [206]: ▶ arr = np.sum([4,5,5,7,6,2,1,3,6,87,89,4,63,3,2,216,498,1,65,63,1324,6])  
arr
```

Out[206]: 2460

```
In [207]: ▶ arr = np.min([4,5,5,7,6,2,1,3,6,87,89,4,63,3,2,216,498,1,65,63,1324,6])  
arr
```

Out[207]: 1

```
In [208]: ▶ arr = np.max([4,5,5,7,6,2,1,3,6,87,89,4,63,3,2,216,498,1,65,63,1324,6])  
arr
```

Out[208]: 1324

```
In [209]: ▶ arr = np.mean([4,5,5,7,6,2,1,3,6,87,89,4,63,3,2,216,498,1,65,63,1324,6])  
arr
```

Out[209]: 111.81818181818181

```
In [210]: ▶ arr = np.median([4,5,5,7,6,2,1,3,6,87,89,4,63,3,2,216,498,1,65,63,1324,6])  
arr
```

Out[210]: 6.0

```
In [211]: ▶ arr = np.std([4,5,5,7,6,2,1,3,6,87,89,4,63,3,2,216,498,1,65,63,1324,6])  
arr
```

Out[211]: 286.0771727354886

```
In [212]: ▶ arr = np.var([4,5,5,7,6,2,1,3,6,87,89,4,63,3,2,216,498,1,65,63,1324,6])  
arr
```

Out[212]: 81840.14876033059

```
In [213]: ▶ 286.0771727354886**2
```

Out[213]: 81840.14876033059

```
In [214]: ▶ arr = np.argmin([4,5,5,7,6,2,1,3,6,87,89,4,63,3,2,216,498,1,65,63,1324,6])  
arr
```

Out[214]: 6

```
In [215]: ▶ arr = np.argmax([4,5,5,7,6,2,1,3,6,87,89,4,63,3,2,216,498,1,65,63,1324,6])  
arr
```

Out[215]: 20

```
In [217]: ▶ np.sqrt(45)
```

Out[217]: 6.708203932499369

```
In [218]: ▶ np.sqrt(25)
```

Out[218]: 5.0

```
In [219]: ▶ np.sqrt(144)
```

Out[219]: 12.0

```
In [ ]: ▶
```

```
In [220]: ▶ arr = np.array([[1,2,3],[4,5,6], [7,8,9]])  
arr
```

Out[220]: array([[1, 2, 3],
[4, 5, 6],
[7, 8, 9]])

```
In [221]: ▶ arr.transpose()
```

Out[221]: array([[1, 4, 7],
[2, 5, 8],
[3, 6, 9]])

In []: ▶

In [222]: ▶ `arr = np.array([[[1,2],[3,4]],[[5,6],[7,8]]])`
`arr`

Out[222]: `array([[[1, 2],
[3, 4]],

[[5, 6],
[7, 8]]])`

In [223]: ▶ `arr.transpose()`

Out[223]: `array([[[1, 5],
[3, 7]],

[[2, 6],
[4, 8]]])`

In []: ▶

In [224]: ▶ `arr`

Out[224]: `array([[[1, 2],
[3, 4]],

[[5, 6],
[7, 8]]])`

In [225]: ▶ `arr.reshape(1, -1)`

Out[225]: `array([[1, 2, 3, 4, 5, 6, 7, 8]])`

```
In [226]: ▶ arr.reshape(-1, 1)
```

```
Out[226]: array([[1],
                 [2],
                 [3],
                 [4],
                 [5],
                 [6],
                 [7],
                 [8]])
```

```
In [ ]: ▶
```

```
In [227]: ▶ arr = np.array([[1,2,3],[4,5,6], [7,8,9]])
arr
```

```
Out[227]: array([[1, 2, 3],
                 [4, 5, 6],
                 [7, 8, 9]])
```

```
In [230]: ▶ arr.reshape(1, -1)
```


```
Out[230]: array([[1, 2, 3, 4, 5, 6, 7, 8, 9]])
```

```
In [231]: ▶ arr.reshape(-1, 1)
```


```
Out[231]: array([[1],
                 [2],
                 [3],
                 [4],
                 [5],
                 [6],
                 [7],
                 [8],
                 [9]])
```

```
In [233]: ▶ arr.reshape(-1, 1).shape
```

```
Out[233]: (9, 1)
```

In [234]:  `arr.reshape(1, -1)`

Out[234]: `array([[1, 2, 3, 4, 5, 6, 7, 8, 9]])`

In [235]:  `arr.reshape(1, -1).shape`

Out[235]: `(1, 9)`

In []: 

In []: 

In []: 

In []: 

In []: 