

Introduction to Numpy

Table of Content

What will We Learn Today?

1. **What is Numpy**
2. **Numpy usage**
3. **Numpy ability**
4. **Practice Numpy in Python**



C Profile

Professional

- Marketing Data Analyst – B2b Company (2021 - 2022)
- Senior Data Analyst – Kompas (2021)
- Data Scientist – Rukita (2020 – 2021)
- Research Assistant Analyst – Ensterna (2017 – 2019)

Educational Background

- Nuclear Engineering – Universitas Gadjah Mada

Connect with me

- M** <https://dataimpact.medium.com/>
- in** <https://www.linkedin.com/in/ariprabowo/>
- G** <https://github.com/densaiko>



Ari Sulistiyo Prabowo

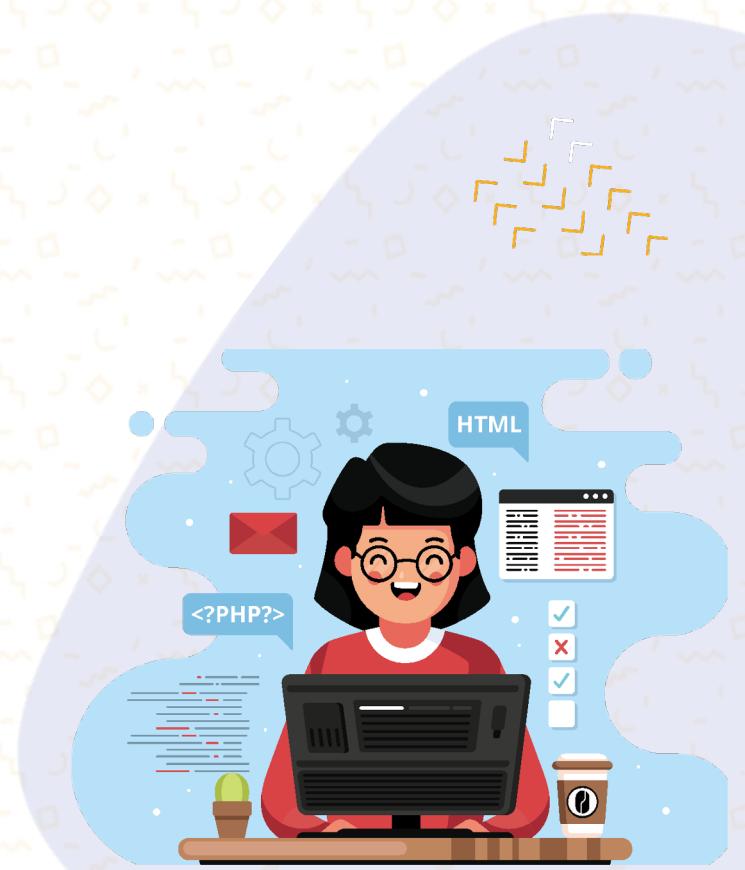




What is Numpy?



- Numpy is a fundamental package **for scientific computing** in Python
- Provides a **high-performance multidimensional array** object, and tools for working with these arrays.
- Used for working with arrays
- has functions for working in domain of linear algebra, fourier transform, and matrices.



1D array

7	2	9	10
---	---	---	----

axis 0 →

shape: (4,)

2D array

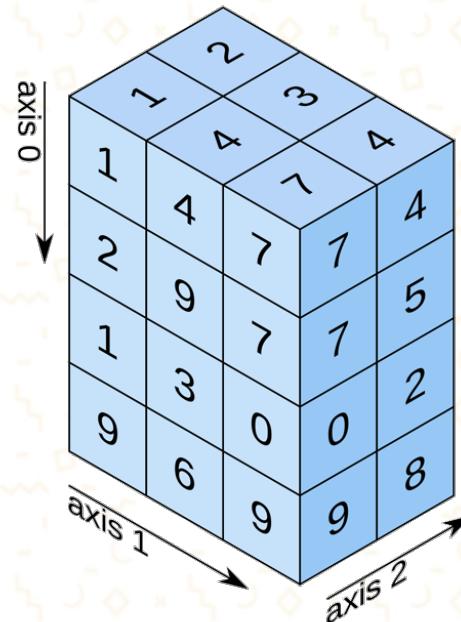
5.2	3.0	4.5
9.1	0.1	0.3

axis 0 ↓

axis 1 →

shape: (2, 3)

3D array



shape: (4, 3, 2)



Why use Numpy?



Why use Numpy



Nearly every scientist working in Python draws on the power of NumPy.



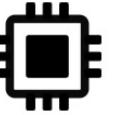
NumPy aims to provide an array object that is up to 50x faster than traditional Python lists



Array can be used in various ecosystem such as scientific domain, data science, machine learning, data visualization



The Almighty Numpy

Quantum Computing	Statistical Computing	Signal Processing	Image Processing	Graphs and Networks	Astronomy Processes	Cognitive Psychology
						
QuTiP PyQuil Qiskit	Pandas statsmodels Xarray Seaborn	SciPy PyWavelets python-control	Scikit-image OpenCV Mahotas	NetworkX graph-tool igraph PyGSP	AstroPy SunPy SpacePy	PsychoPy
Bioinformatics	Bayesian Inference	Mathematical Analysis	Chemistry	Geoscience	Geographic Processing	Architecture & Engineering
						
BioPython Scikit-Bio PyEnsembl ETE	PyStan PyMC3 ArviZ emcee	SciPy SymPy cvxpy FEniCS	Cantera MDAnalysis RDKit	Pangeo Simpeg ObsPy Fatiando a Terra	Shapely GeoPandas Folium	COMPAS City Energy Analyst Sverchok

Source: numpy.org



Numpy Ability



Proof of Numpy is faster than List

```
list1 = [i for i in range(100000)]
list2 = [i for i in range(100000)]

start = time.time()

list_tot = list1 + list2

end = time.time()

print(f"Runtime of the program is {round(end - start,5)}")

Runtime of the program is 0.00234
```

```
np1 = np.array([i for i in range(100000)])
np2 = np.array([i for i in range(100000)])

start = time.time()

num_con = np.concatenate((np1, np2), axis=0)

end = time.time()

print(f"Runtime of the program is {round(end - start,5)}")

Runtime of the program is 0.00068
```



Creating a Numpy Array

1D Array

1	2	3
---	---	---

```
array( [1, 2, 3] )
```

2D Array

1	2	3
1	2	3
1	2	3

```
array( [ [ 1, 2, 3],  
        [ 1, 2, 3],  
        [ 1, 2, 3] ] )
```

www.IndianAIProduction.com

3D Array

1	2	3
1	2	3
1	2	3

```
array( [ [ [ 1, 2, 3],  
          [ 1, 2, 3],  
          [ 1, 2, 3] ],  
        [ [ 1, 2, 3],  
          [ 1, 2, 3],  
          [ 1, 2, 3] ],  
        [ [ 1, 2, 3],  
          [ 1, 2, 3],  
          [ 1, 2, 3] ] ] )
```



Indexing an Array

Expression	Shape
arr[:2, 1:]	(2, 2)
arr[2]	(3,)
arr[2, :] arr[2:, :]	(3,) (1, 3)
arr[:, :2]	(3, 2)
arr[1, :2] arr[1:2, :2]	(2,) (1, 2)

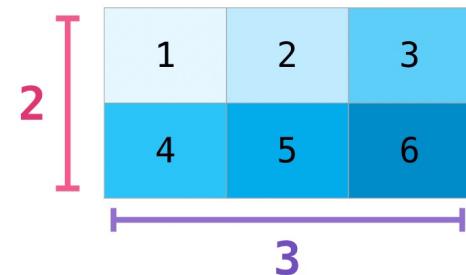


Array Reshaping

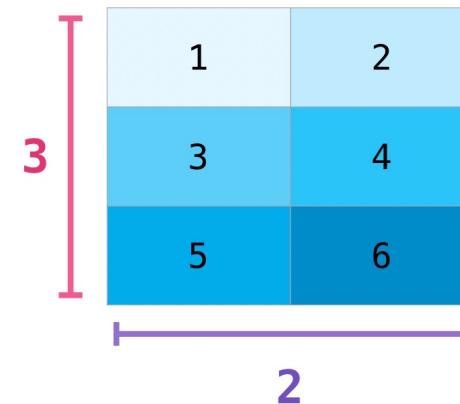
`data`

1
2
3
4
5
6

`data.reshape(2,3)`

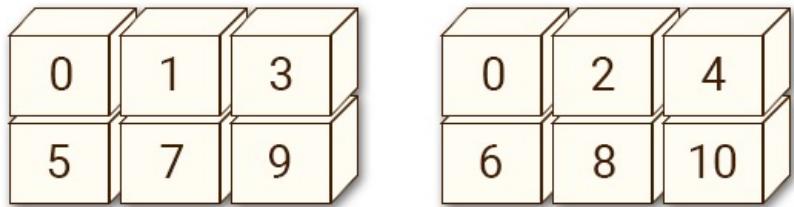


`data.reshape(3,2)`

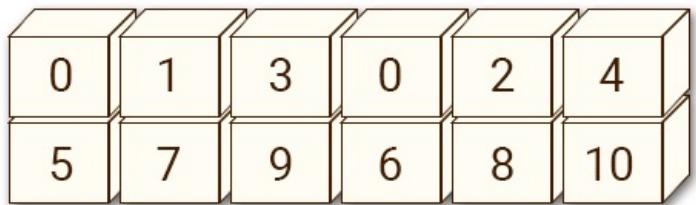




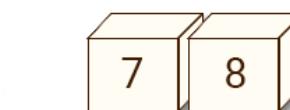
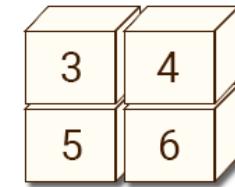
Joining Array



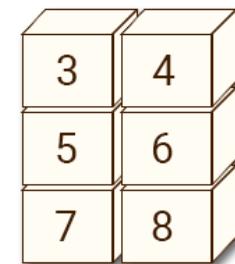
np.concatenate((a, b), 1)



© w3resource.com



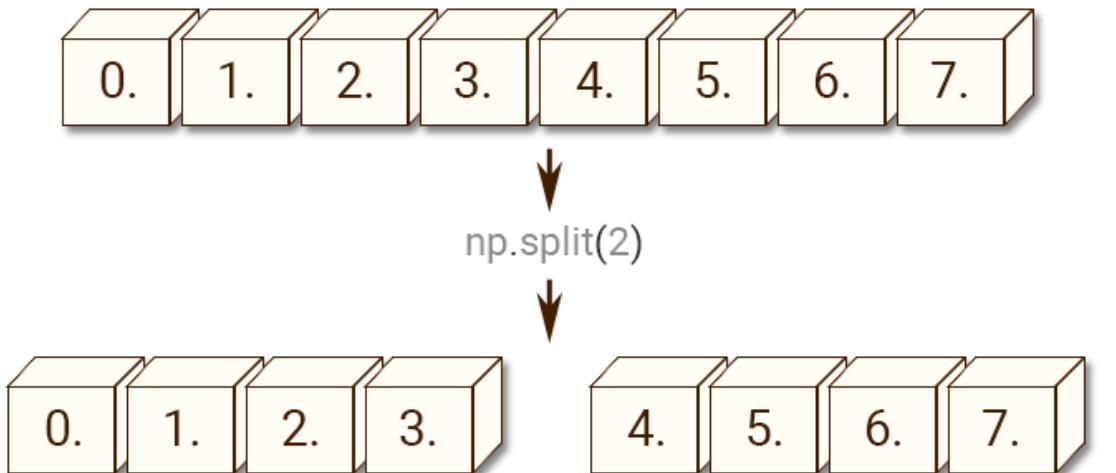
np.concatenate(axis=0)



© w3resource.com



Splitting Array



© w3resource.com

C Sorting Array

Original array:

5	2	8	7	1
---	---	---	---	---

Array after sorting:

1	2	5	7	8
---	---	---	---	---



Filtering Array

Python



NumPy Filtering Arrays

Array →

1	2	3	4
---	---	---	---

completely divisible by 2

New_Array → [2, 4]





Matrix Operation

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \times \begin{bmatrix} 10 & 11 \\ 20 & 21 \\ 30 & 31 \end{bmatrix}$$

$$= \begin{bmatrix} 1 \times 10 + 2 \times 20 + 3 \times 30 & 1 \times 11 + 2 \times 21 + 3 \times 31 \\ 4 \times 10 + 5 \times 20 + 6 \times 30 & 4 \times 11 + 5 \times 21 + 6 \times 31 \end{bmatrix}$$

$$= \begin{bmatrix} 10+40+90 & 11+42+93 \\ 40+100+180 & 44+105+186 \end{bmatrix} = \begin{bmatrix} 140 & 146 \\ 320 & 335 \end{bmatrix}$$

Let's Practice

https://github.com/densaiko/data_science_learning/blob/main/Introduction_to_Numpy.ipynb





Thank YOU